National Institute for Applied Statistics Research Australia

University of Wollongong, Australia

Working Paper

10-21

Robust Particle Density Tempering for State Space Models

David Gunawan, Robert Kohn, and Minh Ngoc Tran

National Institute for Applied Statistics Research Australia, University of Wollongong, Wollongong NSW 2522, Australia Phone +61 2 4221 5435, Fax +61 2 4221 4998.
Email: karink@uow.edu.au

Robust Particle Density Tempering for State Space Models

David Gunawan[2,4], Robert Kohn[1,4] and Minh Ngoc Tran[3,4]

[1]School of Economics, UNSW Business School, University of New South Wales

[2]School of Mathematics and Applied Statistics, University of Wollongong

[3]Discipline of Business Analytics, University of Sydney

[4]Australian Center of Excellence for Mathematical and Statistical Frontiers

April 6, 2021

Abstract

Density tempering (also called density annealing) for state space models is a sequential Monte Carlo (SMC) approach to Bayesian inference for general state models, that provides an alternative to MCMC. It moves a collection of parameters and latent states (which we call particles) through a number of stages, with each stage having its own target density. Initially, the particles are generated from a distribution that is easy to sample from, e.g. the prior; the target density at the final stage is the posterior density of interest. Tempering is usually carried out either in batch mode, involving all of the data at each stage, or in sequential mode, where the tempering involves adding observations at each stage; we call this data tempering. Our article proposes two innovations for particle based density tempering. First, data tempering is made more robust to outliers and structural changes by adding batch tempering at each stage. Second, we propose generating the parameters and states at each stage using two Gibbs type Markov moves, where the parameters are generated conditional on the states and conversely. We explain how this allows

the tempering to scale up in terms of the number parameters and states it can handle. Most of the current literature uses a pseudo-marginal Markov move step with the states integrated out and the parameters generated by a random walk proposal; this strategy is inefficient when the states or parameters are high dimensional. The article demonstrates the performance of the proposed methods using univariate stochastic volatility models with outliers and structural breaks and high dimensional factor stochastic volatility models having both a large number of parameters and a large number of latent state.

Keywords: Factor stochastic volatility model; Hamiltonian Monte Carlo; Particle Gibbs; Outliers; Structural change; Particle Markov chain Monte Carlo

1 Introduction

Joint inference over both the parameters and latent states in nonlinear and non Gaussian models is challenging because the likelihood is usually an intractable integral over the latent states. In a seminal contribution, Andrieu et al. (2010) propose two batch particle Markov chain Monte Carlo (PMCMC) methods for state space models. The first method is the particle marginal Metropolis-Hastings (PMMH), where the parameters are generated with the unobserved latent states integrated out. The second is the particle Gibbs (PG) method, which generates the parameters conditional on the states and conversely. Both methods run a particle filter algorithm within an MCMC sampling scheme at each iteration. Andrieu et al. show that for any finite number of particles, the augmented target density used by these two algorithms has the joint posterior density of the parameters and states as a marginal density. A number of papers, including Lindsten and Schon (2013), Lindsten et al. (2014), Olsson and Ryden (2011), Mendes et al. (2020), and Gunawan et al. (2020), extend their work.

Density tempering (also called annealed importance sampling (AIS)) is an alternative to MCMC as a computational approach for Bayesian inference. It is a sequential importance sampling method (Neal, 2001), where samples are first drawn from an easily-generated distribution, e.g., the prior, and then moved towards the target distribution through a sequence of intermediate stages, with each stage having its own target density. Del Moral et al. (2006) outline the sequential Monte Carlo (SMC) methodology, which consists of resampling, reweighting and Markov move steps at each stage of the SMC.

It is generally recognized (Del Moral et al., 2006) that density tempering has the following advantages compared to MCMC, and in particular, PMCMC approaches: (a) the Markov chain generated by an MCMC sampler can often be trapped in local

modes; it is also difficult to assess whether the chain has mixed adequately and converged to its invariant density. AIS explores the parameter space more efficiently when the target distribution is multimodal; assessing convergence is also easier. (b) It is usually difficult to use MCMC to estimate the marginal likelihood, which is often used to choose between models (Kass and Raftery, 1995; Chib and Jeliazkov, 2001). Using AIS to estimate the marginal likelihood is more straightforward. Section 4.2 also shows the usefulness of the sequential approach to estimate the sequence of independent standard normal random variables used to form the goodness of fit statistics to test for the model adequacy of a general time series state space model as in Smith (1985), Shephard (1994), and Gerlach et al. (1998). Using MCMC to compute the goodness of fit statistics can be very time-consuming as it is necessary to repeatedly construct the full Markov chain for each time period t; see, for example, Gerlach et al. (1998). (c) MCMC algorithms are not parallelizable in general, whereas it is straightforward to parallelize the AIS algorithm; see, for example, Neal (2001), Del Moral et al. (2006), and Duan and Fulop (2015).

Duan and Fulop (2015) extend the annealing/density tempering method of Neal (2001), and Del Moral et al. (2006) to time series state space models having intractable likelihoods and call their approach the Density Tempered Sequential Monte Carlo (SMC) algorithm. Their approach processes the time series as a batch. Fulop and Li (2013) and Chopin et al. (2013) propose full sequential Bayesian inference on the parameters and states and call their algorithm Marginalised Resample-Move (MRM) and SMC^2, respectively.

Our paper builds on this literature and extends density tempering methods in two ways. The first contribution adds a batch tempering step to make sequential density tempering more robust. Sections 4.3 and 4.4 show that the proposed sequential estimation is more stable than Fulop and Li (2013) and Chopin et al. (2013) in datasets with outliers and structural breaks because it incorporates information from the data more gradually.

The Markov move step in SMC is used to help diversify the particles so that they better approximate the tempered target density at that temperature level. Duan and Fulop (2015), Fulop and Li (2013) and Chopin et al. (2013) use a pseudo marginal Metropolis-Hastings (PMMH) approach with the likelihood estimated unbiasedly by the particle filter in the Markov move component. There are two drawbacks to a Markov move based on PMMH. First, it is computationally costly to follow the guidelines of Pitt et al. (2012) for high dimensional state space models and set the optimal number of particles in the particle filter such that the variance of the log of the estimated likelihood is in the range of 1 to 3; our empirical example in Section 6 shows this. It is possible to implement the correlated PMMH of Deligiannidis et al. (2018),

which is more efficient than standard PMMH for low dimensional state space models. However, the correlated PMMH is ineffective for a model with a high dimensional state, e.g. the factor SV model in Section 6. Second, it is hard to implement the PMMH Markov move efficiently for a high dimensional parameter as it is difficult to obtain a good proposal for the parameter; this is because the first and second derivatives with respect to the parameters can only be estimated, while a random walk proposal is easy to implement but is very inefficient in high dimensions.

Our second contributions is to propose two flexible annealing approaches that use Markov move steps that are more efficient than the density tempered SMC approach of Duan and Fulop (2015), MRM of Fulop and Li (2013), and SMC^2 of Chopin et al. (2013). The first is based on the PG algorithm of Andrieu et al. (2010); we call it Annealed Importance Sampling for Intractable Likelihood with particle Gibbs (AISIL-PG). The second is based on the Hamiltonian Monte Carlo (HMC) algorithm (Duane et al., 1987; Neal, 2011; Girolami and Calderhead, 2011; Hoffman and Gelman, 2014; Betancourt, 2017); we call it Annealed Importance Sampling for Intractable Likelihood with Hamiltonian Monte Carlo (AISIL-HMC).

The AISIL-PG and AISIL-HMC methods allow for particle Metropolis within Gibbs moves and Hamiltonian Monte Carlo moves, which is important for two reasons. First, it permits applications to much higher dimensional parameter spaces and much better proposals than random walk proposals; in particular, in some cases we are able to sample from the full conditional distribution using Gibbs steps. Second, using particle Metropolis within Gibbs (PMwG) Markov moves can substantially reduce the number of particles required because it is then unnecessary for the variance of the log of the likelihood estimate to be in the range of 1 to 3 as shown in Section 6.

We illustrate the proposed AISIL methods empirically using a sample of daily US stock returns to estimate the univariate stochastic volatility (SV) models with outliers and structural breaks and the multivariate high dimensional factor SV model. The factor SV model is popular because it parsimoniously models a vector of returns. Most of the applications of the multivariate factor stochastic volatility models are in financial econometrics (Nardari and Scruggs, 2007; Aguilar and West, 2000; Zhou et al., 2014). Current popular MCMC estimation approaches for the factor SV model by Chib et al. (2006) and Kastner et al. (2017) are neither exact nor flexible; neither paper corrects for the approximation errors. They follow the approach proposed by Kim et al. (1998) to approximate the distribution of outcome innovations by a carefully constructed seven component mixture of normal distributions. Gunawan et al. (2020) recently proposed an exact and flexible particle MCMC sampler called the particle hybrid sampler (PHS) and applied it to estimate multivariate factor stochastic volatility model. In the empirical applications, we compare the accuracy

of the proposed AISIL methods to the PHS. We note that with our proposed AISIL methods, it is unnecessary to deal with the autocorrelation issues of PMCMC and MCMC samplers in general. Mendes et al. (2020) and Gunawan et al. (2020) find that it is necessary to use a combination of PMMH and PG sampler (PHS) to reduce the autocorrelation in the Markov chain in the factor SV model. Our article shows that the Gibbs-type Markov move steps in AISIL give accurate and very similar results to the PHS. Section 6 discusses this further.

The rest of the article is organised as follows. Section 2 outlines the basic state space model and its estimation by AISIL. Section 3 discusses the application of the AISIL-PG and AISIL-HMC methods to the univariate SV model. Section 4 presents empirical results for the univariate example. Section 5 introduces the factor SV model, and discusses its estimation by AISIL-PG and AISIL-HMC. Section 6 presents empirical results for the factor SV models. The online supplement to the paper contains some further technical and empirical results. We use the following notation in both the main paper and the online supplement. Eq. (1), Section 1, Algorithm 1 and Sampling Scheme 1, etc. refer to the main article, while Eq. (S1), Section S1, Algorithm S1, and Sampling Scheme S1, etc. refer to the supplement.

2 Flexible Density Tempering

This section first introduces the basic state space model and then discusses the two novel flexible AISIL approaches.

2.1 The Basic State Space Model

We consider a state space model where the latent states $\boldsymbol{X}_t$ determine the evolution of the system[1]. The density of $\boldsymbol{X}_1$ is $f_1^{\boldsymbol{\theta}}(\boldsymbol{x}_1)$ and the transition density of $\boldsymbol{X}_t$ given $\boldsymbol{X}_{1:t-1} = \boldsymbol{x}_{1:t-1}$ is $f_t^{\boldsymbol{\theta}}(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ for $t \geq 2$ [2]. The observations $\boldsymbol{Y}_t = \boldsymbol{y}_t$ for $t = 1, ..., T$ are linked to the latent states through the observation equation $g_t^{\boldsymbol{\theta}}(\boldsymbol{y}_t|\boldsymbol{x}_t)$. Denote the d dimensional Euclidean space by $\mathbb{R}^d$. We define that: (i) The state vector $\boldsymbol{x}_t \in \mathcal{X}$ and $\boldsymbol{y}_t \in \mathcal{Y}$ where $\mathcal{X} \subset \mathbb{R}^{d_x}$ and $\mathcal{Y} \subset \mathbb{R}^{d_y}$; (ii) the parameter vector $\boldsymbol{\theta} \in \boldsymbol{\Theta}$, where $\boldsymbol{\Theta}$ is a subset of $\mathbb{R}^{d_\theta}$.

Our aim is to perform Bayesian inference over the latent states and the parameters conditional on the observations $\boldsymbol{y}_{1:T}$. By using Bayes rule, the joint posterior of the

[1] A vector is written in boldface.

[2] We use the colon notation for collections of variables, i.e. $\boldsymbol{a}_t^{1:N} = \left(a_t^1, ..., a_t^N\right)$, $\boldsymbol{a}_{1:t}^{1:N} = \left(\boldsymbol{a}_1^{1:N}, ..., \boldsymbol{a}_t^{1:N}\right)$, and $\boldsymbol{a}_{1:M,1:t} = (a_{1,1}, ..., a_{M,1}, ..., a_{1,t}, ..., a_{M,t})$.

latent states and the parameters is

$$p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}) = \frac{p(\boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})}{p(\boldsymbol{y}_{1:T})},$$
$$\text{where} \quad p(\boldsymbol{x}_{1:T}, \boldsymbol{y}_{1:T}|\boldsymbol{\theta}) = p(\boldsymbol{x}_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}) \prod_{t=1}^{T} p(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta});$$

$p(\boldsymbol{\theta})$ is the prior density of $\boldsymbol{\theta}$ and $p(\boldsymbol{y}_{1:T})$ is the marginal likelihood of $\boldsymbol{y}_{1:T}$. For notational simplicity we will often write $\boldsymbol{x} = \boldsymbol{x}_{1:T}$ and $\boldsymbol{y} = \boldsymbol{y}_{1:T}$.

2.2 The Annealing Approach for State Space Models

2.2.1 Batch Estimation Method

This section discusses the proposed AISIL method for batch estimation problems. The main idea is to begin with an easy-to-sample distribution and propagate a particle cloud $\left\{\boldsymbol{\theta}_{1:M}^{(p)}, \boldsymbol{x}_{1:M}^{(p)}, \boldsymbol{W}_{1:M}^{(p)}\right\}$ through a sequence of tempered target densities $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$, for $p = 0, ..., P$, to the posterior density of interest which is much harder to sample from directly. The tempered densities are defined as

$$\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) := \eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) / Z_{a_p}, \quad \text{where} \quad Z_{a_p} := \int \eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})\, d\boldsymbol{\theta} d\boldsymbol{x}$$
$$\text{and} \quad \eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) := (\pi_0(\boldsymbol{\theta}, \boldsymbol{x}))^{1-a_p} \left(p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x})\, p(\boldsymbol{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta})\right)^{a_p}.$$

The tempering sequence $\boldsymbol{a}_{0:P}$ is such that $a_0 = 0 < a_1 < ... < a_P = 1$. If it is both easy to generate from the densities $p(\boldsymbol{\theta})$ and $p(\boldsymbol{x}|\boldsymbol{\theta})$, and evaluate them, then we take $\pi_0(\boldsymbol{x}, \boldsymbol{\theta}) = p(\boldsymbol{\theta})\, p(\boldsymbol{x}|\boldsymbol{\theta})$, and hence

$$\eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x})^{a_p}\, p(\boldsymbol{x}|\boldsymbol{\theta})\, p(\boldsymbol{\theta}). \tag{1}$$

Algorithm 1 summarizes the general AISIL approach, which is now discussed. The initial particle cloud $\left\{\boldsymbol{\theta}_{1:M}^{(0)}, \boldsymbol{x}_{1:M}^{(0)}, \boldsymbol{W}_{1:M}^{(0)}\right\}$ is obtained by generating the $\left\{\boldsymbol{\theta}_{1:M}^{(0)}, \boldsymbol{x}_{1:M}^{(0)}\right\}$ from $\pi_0(\boldsymbol{x}, \boldsymbol{\theta})$, and giving the particles equal weight, i.e., $\boldsymbol{W}_{1:M}^{(0)} = 1/M$. The weighted particles (particle cloud) $\left\{\boldsymbol{\theta}_{1:M}^{(p-1)}, \boldsymbol{x}_{1:M}^{(p-1)}, \boldsymbol{W}_{1:M}^{(p-1)}\right\}$ at the $(p-1)$st level (or stage) of the annealing is an estimate of $\xi_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x})$. Based on this estimate of $\xi_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x})$, the AISIL algorithm goes through the following steps to obtain an estimate of $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$. The move from $\xi_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x})$ to $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$ is implemented by reweighting the particles by the ratio of the two unnormalized densities $\eta_{a_p}/\eta_{a_{p-1}}$,

yielding the following weights, $\boldsymbol{W}_{1:M}^{(p)} = \boldsymbol{w}_{1:M}^{(p)} / \sum_{j=1}^{M} w_j^{(p)}$, where

$$w_i^{(p)} = W_i^{(p-1)} \frac{\eta_{a_p}\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)}{\eta_{a_{p-1}}\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)} = W_i^{(p-1)} p\left(\boldsymbol{y}|\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)^{a_p - a_{p-1}}, \quad (2)$$

where η_{a_p} is defined in Eq. (1). We follow Del Moral et al. (2012) and choose the tempering sequence adaptively to ensure sufficient particle diversity by selecting the next value of a_p automatically such that the effective sample size (ESS) stays close to some target value ESS_T chosen by the user[3]. An ESS close to the target value is achieved by evaluating the ESS over a grid $a_{1:G,p}$ of potential a_p values and select as a_p the value of $a_{j,p}$ whose ESS is the closest to ESS_T. The particles $\left\{\boldsymbol{\theta}_{1:M}^{(p)}, \boldsymbol{x}_{1:M}^{(p)}\right\}$ are then resampled proportionately to their weights $\boldsymbol{W}_{1:M}^{(p)}$, which means that the resampled particles then have equal weight $\boldsymbol{W}_{1:M}^{(p)} = 1/M$. This has the effect of eliminating particles with negligible weights and replicating the particles with large weights, so the ESS is now M.

Repeatedly reweighting and resampling can seriously reduce the diversity of the particles, and lead to particle depletion. The particle cloud $\left\{\boldsymbol{\theta}_{1:M}^{(p)}, \boldsymbol{x}_{1:M}^{(p)}, \boldsymbol{W}_{1:M}^{(p)}\right\}$ may then not be a good representation of $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$. To improve the approximation of the particle cloud to $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$, we carry out R Markov moves steps for each particle using a Markov kernel K_{a_p} which retains $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$ as its invariant density. Thus, at each annealing schedule, we run a short MCMC scheme for each of the M particles.

Step 1 and Steps 2a-2d of Algorithm 1 are standard and apply to any model, with only slight modifications; hence, it is only necessary to discuss the Markov move step in detail for each model. We propose two Markov move algorithms that leave the target density $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$ invariant. The first algorithm is based on particle Gibbs (Andrieu et al., 2010), and is denoted as the AISIL-PG. The second algorithm is based on Hamiltonian Monte Carlo (Neal, 2011), and is denoted as AISIL-HMC. By Del Moral et al. (2006), the AISIL algorithm provides consistent inference for the target density $p(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T})$ as M goes to infinity. See also Beskos et al. (2016) for recent consistency results for the adaptive algorithm employed here and the associated central limit theorem.

[3]The effective sample size ESS is used to measure the variability in the $W_i^{(p)}$, and is defined as $\text{ESS} = \left(\sum_{i=1}^{M}\left(W_i^{(p)}\right)^2\right)^{-1}$. The ESS varies between 1 and M, with a low value of ESS indicating that the weights are concentrated only on a few particles and a value of M means that the particles are equally weighted.

Algorithm 1 Generic AISIL Algorithm

1. Set $p = 0$ and initialise the particle cloud $\left\{\boldsymbol{\theta}_{1:M}^{(0)}, \boldsymbol{x}_{1:M}^{(0)}, \boldsymbol{W}_{1:M}^{(0)}\right\}$, by generating the $\left\{\boldsymbol{\theta}_{1:M}^{(0)}, \boldsymbol{x}_{1:M}^{(0)}\right\}$ from $\pi_0(\boldsymbol{x}, \boldsymbol{\theta})$, and giving them equal weight, i.e., $W_i^{(0)} = 1/M$, for $i = 1, ..., M$
2. While the tempering sequence $a_p < 1$ do
   (a) Set $p \leftarrow p + 1$
   (b) Find a_p adaptively by searching across a grid of a_p to maintain effective sample size (ESS) around some constant ESS_T.
   (c) Compute new weights,

$$\boldsymbol{W}_{1:M}^{(p)} = \frac{\boldsymbol{w}_{1:M}^{(p)}}{\sum_{j=1}^{M} w_j^{(p)}} \text{ where } w_i^{(p)} = W_i^{(p-1)} \frac{\eta_{a_p}\left(\boldsymbol{\theta}_{\boldsymbol{i}}^{(p-1)}, \boldsymbol{x}_{\boldsymbol{i}}^{(p-1)}\right)}{\eta_{a_{p-1}}\left(\boldsymbol{\theta}_{\boldsymbol{i}}^{(p-1)}, \boldsymbol{x}_{\boldsymbol{i}}^{(p-1)}\right)}. \quad (3)$$

   (d) Resample $\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)$ using the weights $\boldsymbol{W}_{1:M}^{(p)}$ to obtain $\left\{\boldsymbol{\theta}_{1:M}^{(p)}, \boldsymbol{x}_{1:M}^{(p)}, \boldsymbol{W}_{1:M}^{(p)} = 1/M\right\}$.
   (e) Markov moves
      i. Let $K_{a_p}((\boldsymbol{\theta}, \boldsymbol{x}), \cdot)$ be a Markov kernel having invariant density $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$. For $i = 1, ..., M$, move each $\left(\boldsymbol{\theta}_i^{(p)}, \boldsymbol{x}_i^{(p)}\right)$ R times using the Markov kernel K_{a_p} to obtain $\left\{\widetilde{\boldsymbol{\theta}}_i, \widetilde{\boldsymbol{x}}_i\right\}$.
      ii. Set $\left(\boldsymbol{\theta}_{1:M}^{(p)}, \boldsymbol{x}_{1:M}^{(p)}\right) \leftarrow \left(\widetilde{\boldsymbol{\theta}}_{1:M}, \widetilde{\boldsymbol{x}}_{1:M}\right)$.

2.2.2 Sequential Estimation Method

This section discusses the extension of the density tempered/annealing approach in Section 2.2.1 to the situation where we are interested in sequential estimation of both the states and parameters as new observations arrive. This approach has very useful applications, especially in finance and economics. For example, in finance, investors need to update their volatility and return forecasts whenever new data become available. As another example, economists need to revise their forecasts as news on the state of the economy changes. Here, PMCMC techniques are expensive and time-consuming to use as they need to repeatedly construct the full Markov chain as new observations arrive.

The main issue is how to jointly sample and propagate the states and parameters as new observations arrive. Our framework deals with this by extending the AISIL

sequence of target distributions as the number of observations increases. The main idea is to propagate a particle cloud $\left\{\boldsymbol{\theta}_{1:M}^{(p)}, \boldsymbol{x}_{1:t-1,1:M}^{(p)}, \boldsymbol{W}_{1:M}^{(p)}\right\}$ that reflects the joint posterior of states $\boldsymbol{x}_{1:t-1}$ and parameters $\boldsymbol{\theta}$ at time $t-1$ through a sequence of tempered target densities $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$, for $p = 0, ..., P$, to the joint posterior density of states $\boldsymbol{x}_{1:t}$ and parameters $\boldsymbol{\theta}$ at time t. The tempered densities are defined as

$$\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) := \eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) / Z_{a_p}, \quad \text{where } Z_{a_p} := \int \eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) \, d\boldsymbol{\theta} d\boldsymbol{x},$$
$$\text{and } \eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) := p(\boldsymbol{y}_t | \boldsymbol{x}_t, \boldsymbol{\theta})^{a_p} p(\boldsymbol{x}_t | \boldsymbol{x}_{t-1}, \boldsymbol{\theta}) p(\boldsymbol{\theta}, \boldsymbol{x}_{1:t-1} | \boldsymbol{y}_{1:t-1}).$$

The move from $\xi_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x})$ to $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$ is implemented by reweighting the particles by the ratio of the two unnormalised densities $\eta_{a_p}/\eta_{a_{p-1}}$, yielding the weights

$$w_i^{(p)} := W_i^{(p-1)} \frac{\eta_{a_p}\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)}{\eta_{a_{p-1}}\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)} = W_i^{(p-1)} p\left(\boldsymbol{y}_t | \boldsymbol{x}_{i,t}^{(p-1)}, \boldsymbol{\theta}_i^{(p-1)}\right)^{a_p - a_{p-1}}.$$

Sections 4.3, 4.4, and 6 respectively show that our sequential approach is robust to the outliers, structural breaks and applies to high dimensional state space models; this happens because tempering incorporates the information in new observations gradually. This also means that it is easy to maintain the ESS close to its target ESS_T. The resampling and Markov move steps are similar to Section 2.2.1 and are discussed in Sections 3 and 5 for the univariate SV and the multivariate factor SV models, respectively.

3 Applying the AISIL method to the Univariate Stochastic Volatility Model

This section applies the AISIL methods to the simple univariate SV model in a batch context. It is straightforward to extend this to sequential estimation. The univariate model is introduced first as it forms the basis of the factor SV model studied later.

$$y_t = \exp(x_t/2) \epsilon_t, \quad x_{t+1} = \mu + \phi(x_t - \mu) + \tau\eta_t, \quad x_1 \sim N\left(\mu, \frac{\tau^2}{1-\phi^2}\right), \tag{4}$$

with $\epsilon_t \sim N(0,1)$ and $\eta_t \sim N(0,1)$; they are independent. We call $\boldsymbol{x}_{1:T}$ the latent volatility process. The vector of unknown parameters is $\boldsymbol{\theta} = (\mu, \phi, \tau^2)$. The parameters have the following priors, and are assumed to be independent apriori: (a) $p(\mu) \propto I(-10 < \mu < 10)$. (b) Following Jensen and Maheu (2010), the prior for τ^2 is inverse Gamma $\text{IG}(v_0/2, s_0/2)$ with $v_0 = 10$ and $s_0 = 0.5$. (c) To ensure

stationarity, the persistence parameter is restricted $-1 < \phi < 1$; we follow Kim et al. (1998) and choose the prior for ϕ as $(\phi + 1)/2 \sim \text{Beta}(a_0, b_0)$, with $a_0 = 100$ and $b_0 = 1.5$.

3.1 AISIL-HMC Markov Moves

This section applies the AISIL-HMC Markov moves for the univariate stochastic volatility model in Eq. (4). The appropriate sequence of intermediate target densities for $p = 1, ..., P$, is

$$\eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x})^{a_p} p(\boldsymbol{x}|\boldsymbol{\theta}) p(\boldsymbol{\theta}),$$
$$\text{where} \quad p(\boldsymbol{y}|\boldsymbol{\theta}, \boldsymbol{x})^{a_p} = \prod_{t=1}^{T} p(y_t|x_t, \boldsymbol{\theta})^{a_p}, \;\; p(\boldsymbol{x}|\boldsymbol{\theta}) = p(x_1|\boldsymbol{\theta}) \prod_{t=2}^{T} p(x_t|x_{t-1}, \boldsymbol{\theta});$$

$p(\boldsymbol{\theta})$ is the prior of $\boldsymbol{\theta}$. The Markov move has two steps. The first step updates the latent log volatilities $\boldsymbol{x}_{1:T}$ conditional on the parameters $\boldsymbol{\theta}$ using Hamiltonian Monte Carlo (HMC). The second step updates the parameters $\boldsymbol{\theta}$ conditional on the latent log-volatilities $\boldsymbol{x}_{1:T}$. Section S1 gives the details.

Section S1.1 discusses sampling the log-volatilities using Hamiltonian Monte Carlo method.

3.2 AISIL-PG Markov Moves

This section discusses the application of AISIL-PG Markov moves to a univariate stochastic volatility model. The main idea of Markov moves based on particle Gibbs is to define a sequence of tempered target densities on an augmented space that includes all of the parameters and all the particles generated by the particle filter. The augmented target density has $\xi_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})$ as a marginal density for $p = 0, ..., P$. Section 3.2.1 discusses the particle filter. Section 3.2.2 discusses the augmented target density.

3.2.1 Particle Filter

We first describe the particle filter methods used to obtain sequential approximations to the densities $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$ for $t = 1, ..., T$. The particle filter algorithm consists of recursively producing a set of weighted particles $\left\{\boldsymbol{x}_{1:t}^{(j)}, \widetilde{W}_t^{(j)}\right\}_{j=1}^{N}$ such that the intermediate densities $p(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t}, \boldsymbol{\theta})$, defined on the sequence of spaces $\{\boldsymbol{\chi}^t; t = 1, ..., T\}$,

are approximated by

$$\widehat{p}\left(\boldsymbol{x}_{1:t}|\boldsymbol{y}_{1:t},\boldsymbol{\theta}\right)=\sum_{j=1}^{N}\widetilde{W}_{t}^{(j)}\delta_{\boldsymbol{x}_{1:t}^{(j)}}\left(d\boldsymbol{x}_{1:t}\right),$$

where $\delta_a\left(dx\right)$ is the Dirac delta distribution located at a. In more detail, given N samples $\left\{\boldsymbol{x}_{t-1}^{(j)},\widetilde{W}_{t-1}^{(j)}\right\}_{j=1}^{N}$ representing the filtering density $p\left(\boldsymbol{x}_{t-1}|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta}\right)$ at time $t-1$, we use

$$p\left(x_t|\boldsymbol{y}_{1:t},\boldsymbol{\theta}\right)\propto\int p\left(y_t|x_t,\boldsymbol{\theta}\right)p\left(x_t|x_{t-1},\boldsymbol{\theta}\right)p\left(x_{t-1}|\boldsymbol{y}_{1:t-1},\boldsymbol{\theta}\right)dx_{t-1},$$

to obtain the particles $\left\{x_t^{(j)},\widetilde{W}_t^{(j)}\right\}_{j=1}^{N}$ representing $p(x_t|\boldsymbol{y}_{1:t},\boldsymbol{\theta})$, by first drawing from a known and easily sampled proposal density function $m\left(x_t|x_{t-1},y_t,\boldsymbol{\theta}\right)$ and then computing the unnormalised weights $\widetilde{w}_t^{(j)}$ to account for the difference between the target posterior density and the proposal, where

$$\widetilde{w}_t^{(j)}:=\widetilde{W}_{t-1}^{(j)}\frac{p\left(y_t|x_t^{(j)},\boldsymbol{\theta}\right)p\left(x_t^{(j)}|x_{t-1}^{(j)},\boldsymbol{\theta}\right)}{m_t\left(x_t^{(j)}|x_{t-1}^{(j)},y_t,\boldsymbol{\theta}\right)},\ j=1,...,N;$$

these are then normalized as $\widetilde{W}_t^{(j)}:=\widetilde{w}_t^{(j)}/\sum_{r=1}^{N}\widetilde{w}_t^{(r)}$.

Chopin (2004) shows that as the number of time periods t increases, the normalised weights of the particle system become concentrated on only a few particles and eventually the normalised weight of a single particle converges to one. This is known as the 'weight degeneracy' problem. One way to reduce the impact of weight degeneracy is to include resampling steps in a particle filter algorithm. A resampling scheme is defined as $\mathcal{M}\left(\widetilde{\boldsymbol{a}}_{t-1}^{1:N}|\widetilde{\boldsymbol{W}}_{t-1}^{1:N}\right)$, where $\widetilde{a}_{t-1}^{j}$ indexes a particle in $\left\{x_{t-1}^{(j)},\widetilde{W}_{t-1}^{(j)}\right\}_{j=1}^{N}$ that is chosen with probability $\widetilde{W}_{t-1}^{(j)}$. Section S5 states some assumptions on the proposal density $m\left(x_t|x_{t-1},y_t,\boldsymbol{\theta}\right)$ and resampling scheme. In our empirical application, we use the bootstrap filter with $p\left(x_t|x_{t-1},\boldsymbol{\theta}\right)$ as a proposal density and multinomial resampling.

The particle filter produces the unbiased estimate of the likelihood

$$\widehat{p}\left(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}\right)=\widehat{p}\left(y_1|\boldsymbol{\theta}\right)\prod_{t=2}^{T}\widehat{p}\left(y_t|y_{t-1},\boldsymbol{\theta}\right)=\prod_{t=1}^{T}\left\{\frac{1}{N}\sum_{j=1}^{N}\widetilde{w}_t^{(j)}\right\},$$

i.e. $\mathrm{E}\left(\widehat{p}\left(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}\right)\right)=p\left(\boldsymbol{y}_{1:T}|\boldsymbol{\theta}\right)$ (Proposition 7.4.1 of Del Moral, 2004)). The likelihood estimate is an essential component of the PMMH Markov move of the density

tempered SMC algorithm of Duan and Fulop (2015).

3.2.2 AISIL-PG Augmented Target Density

This section discusses the appropriate sequence of intermediate augmented target densities for the state space model described in Section 2.1. It includes all the random variables which are produced by the particle filter. Let $\boldsymbol{U}_{1:T}^{1:N} = \left(\boldsymbol{x}_{1:T}^{1:N}, \widetilde{\boldsymbol{a}}_{1:T-1}^{1:N}\right)$ denote the collection of these particles. The joint distribution of the particles given the parameters $\boldsymbol{\theta}$ is

$$\psi\left(\boldsymbol{U}_{1:T}|\boldsymbol{\theta}\right) \coloneqq \prod_{j=1}^{N} m_1\left(x_1^j|\boldsymbol{\theta}, y_1\right) \prod_{t=2}^{T} \left\{ \mathcal{M}\left(\widetilde{\boldsymbol{a}}_{t-1}^{1:N}|\widetilde{\boldsymbol{W}}_{t-1}^{1:N}\right) \prod_{j=1}^{N} m_t\left(x_t^j|x_{t-1}^{\widetilde{a}_{t-1}^j}, y_t, \boldsymbol{\theta}\right) \right\}.$$

The backward simulation algorithm introduced by Godsill et al. (2004) is used to obtain a sample from the particle approximation of $p\left(\boldsymbol{x}_{1:T}|\boldsymbol{y}_{1:T}, \boldsymbol{\theta}\right)$ by sampling the indices $J_T, J_{T-1}, ..., J_1$ sequentially. The selected particle trajectory is denoted by $\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}} = \left(x_1^{j_1}, ..., x_T^{j_T}\right)$; the collection of all random variables except the chosen particle trajectory is denoted by $\boldsymbol{U}_{1:T}^{-\boldsymbol{j}_{1:T}}$. The AISIL-PG constructs a sequence of tempered densities $\xi_{a_p}\left(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}\right)$, $p = 1, ..., P$, based on the following augmented target density

$$\begin{aligned} \widetilde{\xi}_{a_p}\left(\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, \boldsymbol{U}_{1:T}^{-\boldsymbol{j}_{1:T}}, \boldsymbol{\theta}\right) &= \frac{\xi_{a_p}\left(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}\right)}{N^T} \\ &\times \frac{\psi\left(\boldsymbol{U}_{1:T}|\boldsymbol{\theta}\right)}{m_1\left(x_1^{j_1}|\boldsymbol{\theta}, y_1\right) \prod_{t=2}^{T} \widetilde{W}_{t-1}^{\widetilde{a}_{t-1}^{j_t}} m_t\left(x_t^{j_t}|x_{t-1}^{\widetilde{a}_{t-1}^{j_t}}, \boldsymbol{\theta}, y_t\right)} \end{aligned} \tag{5}$$

The next lemma gives the important properties of the target density. Its proof is in Section S6.

Lemma 1. *The target distribution in Eq. (5) has the marginal distribution*

$$\widetilde{\xi}_{a_p}\left(\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, \boldsymbol{\theta}\right) = N^{-T} \xi_{a_p}\left(\boldsymbol{\theta}, \boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}\right).$$

Let $\boldsymbol{\theta} \coloneqq \left(\boldsymbol{\theta}_1, ..., \boldsymbol{\theta}_H\right)$ be a partition of the parameter vector into H components, where each component may be a vector. Algorithm 2 gives the Markov move based on the particle Gibbs algorithm.

Algorithm 2 Markov move based on the Particle Gibbs algorithm

For $i = 1, ..., M$, we propose the following Markov steps

1. For $h = 1, ..., H$, sample $\boldsymbol{\theta}_{ih}^*$ from the proposal density $q_h\left(\cdot|\boldsymbol{j}_{i,1:T}, \boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{\theta}_{-ih}, \boldsymbol{\theta}_{ih}\right)$.

2. Accept the proposed values $\boldsymbol{\theta}_{ih}^*$ with probability

$$\min\left(1, \frac{\xi_{a_p}\left(\boldsymbol{\theta}_{ih}^*|\boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{j}_{i,1:T}, \boldsymbol{\theta}_{-ih}\right)}{\xi_{a_p}\left(\boldsymbol{\theta}_{ih}|\boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{j}_{i,1:T}, \boldsymbol{\theta}_{-ih}\right)} \frac{q_h\left(\boldsymbol{\theta}_{ih}|\boldsymbol{\theta}_{-ih}, \boldsymbol{\theta}_{ih}^*, \boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{j}_{i,1:T}\right)}{q_h\left(\boldsymbol{\theta}_{ih}^*|\boldsymbol{\theta}_{-ih}, \boldsymbol{\theta}_{ih}, \boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{j}_{i,1:T}\right)}\right).$$

3. Sample $\boldsymbol{U}_{i,1:T}^{\left(-\boldsymbol{j}_{i,1:T}\right)} \sim \widetilde{\xi}_{a_p}\left(\cdot|\boldsymbol{j}_{i,1:T}, \boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{\theta}_i\right)$ using conditional particle filter given in Algorithm S4 in Section S4.

4. Sample $J_{i,t} = j_{i,t} \sim \widetilde{\xi}_{a_p}\left(\boldsymbol{j}_{i,1:T}|\boldsymbol{x}_{i,1:t}^{1:N}, \boldsymbol{a}_{i,1:t-1}^{1:N}, \boldsymbol{\theta}_i, \boldsymbol{x}_{i,t+1:T}^{\boldsymbol{j}_{i,t+1:T}}, \boldsymbol{j}_{i,t+1:T}\right)$ for $t = T-1, ..., 1$ and $J_{i,T} = j_{i,T} \sim \widetilde{\xi}_{a_p}\left(j_{i,T}|\boldsymbol{\theta}_i, \boldsymbol{x}_{i,1:T}^{1:N}, \boldsymbol{a}_{i,1:T-1}^{1:N}\right)$ using the backward simulation algorithm given in Algorithm S5 of Section S4.

Parts 1 and 2 of the Algorithm 2 follow from Lemma 1. Section S1.2 gives the details for sampling the univariate SV parameters $\boldsymbol{\theta}$ conditional on the selected trajectory $\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}$. The conditional density in Part 3 is given by construction

$$\widetilde{\xi}_{a_p}\left(\cdot|\boldsymbol{j}_{i,1:T}, \boldsymbol{x}_{i,1:T}^{\boldsymbol{j}_{i,1:T}}, \boldsymbol{\theta}_i\right) = \frac{\psi\left(\boldsymbol{U}_{1:T}|\boldsymbol{\theta}\right)}{m_1\left(dx_1^{j_1}|\boldsymbol{\theta}, y_1\right)\prod_{t=2}^T \widetilde{W}_{t-1}^{\widetilde{a}_{t-1}^{j_t}} m_t\left(x_t^{j_t}|x_{t-1}^{\widetilde{a}_{t-1}^{j_t}}, \boldsymbol{\theta}\right)},$$

and is the density under $\widetilde{\xi}_{a_p}$ of all the variables that are generated by the particle filter algorithm conditional on a pre-specified path. This is the conditional particle filter algorithm of Andrieu et al. (2010). It is a particle filter algorithm in which a particle $\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}} = \left(\boldsymbol{x}_1^{j_1}, ..., \boldsymbol{x}_T^{j_T}\right)$, and the associated sequence of ancestral indices are kept unchanged with all the other particles and indices resampled and updated. Algorithm S4 in Section S4 describes the conditional sequential Monte Carlo algorithm and is the key part of the Markov move steps. Following Lindsten and Schon (2013), the conditional density in Part 4 is

$$\widetilde{\xi}_{a_p}\left(j_t|\boldsymbol{\theta}, x_{1:t}^{1:N}, \widetilde{\boldsymbol{a}}_{1:t-1}^{1:N}, \boldsymbol{x}_{t+1:T}^{\boldsymbol{j}_{t+1:T}}, \boldsymbol{j}_{t+1:T}\right) \propto \widetilde{w}_t^{j_t} f_{t+1}^{\theta}\left(\boldsymbol{x}_{t+1}^{j_{t+1}}|\boldsymbol{x}_t^{j_t}\right),$$

for $t = T-1, ..., 1$ and $\widetilde{\xi}_{a_p}\left(j_T|\boldsymbol{\theta}, \boldsymbol{x}_{1:T}^{1:N}, \widetilde{\boldsymbol{a}}_{1:T-1}^{1:N}\right) \propto \widetilde{w}_T^{j_T}$.

3.3 Estimating the Marginal Likelihood

The marginal likelihood $p(\boldsymbol{y}_{1:T})$ is often used in the Bayesian literature to compare models (Chib and Jeliazkov, 2001). An advantage of the AISIL method is that it offers a natural way to estimate the marginal likelihood. We note that $p(\boldsymbol{y}_{1:T}) = Z_{a_P}$, $Z_{a_0} = 1$, so that

$$p(\boldsymbol{y}_{1:T}) = \prod_{p=1}^{P} \frac{Z_{a_p}}{Z_{a_{p-1}}} \text{ with } \frac{Z_{a_p}}{Z_{a_{p-1}}} = \int \left(\frac{\eta_{a_p}(\boldsymbol{\theta}, \boldsymbol{x})}{\eta_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x})} \right) \xi_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x}) \, d\boldsymbol{\theta} d\boldsymbol{x}.$$

Because the particle cloud $\left\{ \boldsymbol{\theta}_{1:M}^{(p-1)}, \boldsymbol{x}_{1:M}^{(p-1)}, \boldsymbol{W}_{1:M}^{(p-1)} \right\}$ obtained after iteration $p-1$ approximates $\xi_{a_{p-1}}(\boldsymbol{\theta}, \boldsymbol{x})$, the ratio above is estimated by

$$\widehat{\frac{Z_{a_p}}{Z_{a_{p-1}}}} = \sum_{i=1}^{M} W_i^{(p-1)} \frac{\eta_{a_p}\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)}{\eta_{a_{p-1}}\left(\boldsymbol{\theta}_i^{(p-1)}, \boldsymbol{x}_i^{(p-1)}\right)},$$

giving the marginal likelihood estimate

$$\widehat{p}(\boldsymbol{y}_{1:T}) = \prod_{p=1}^{P} \widehat{\frac{Z_{a_p}}{Z_{a_{p-1}}}}.$$

4 Univariate Examples

4.1 Univariate Stochastic Volatility Model

This section illustrates the AISIL-PG and AISIL-HMC methods for both batch and sequential estimation problems by applying them to the univariate stochastic volatility (SV) model discussed in Section 3 and compares their performance to the particle hybrid sampler (PHS) of Gunawan et al. (2020). We apply the methods to a sample of daily US food industry stock returns obtained from the Kenneth French website, using a sample from December 11th, 2001 to the 11th November 2013, a total of 3001 observations.

The performance of HMC depends heavily on choosing suitable values for the three tuning parameters i) the mass matrix $\widetilde{M}$, ii) the step size ϵ, and iii) the number of leapfrog steps L. The step size ϵ determines how well the leapfrog integration approximates the Hamiltonian dynamics. If it is too large, then a low acceptance rate may result, but if it is too small, then it becomes computationally expensive to obtain distant proposals. Similarly, if L is too small, then the proposal will be close to the current value of the latent state vectors, resulting in undesirable random walk behaviour. If L is too large, then HMC will generate proposals that retrace their

steps. The precision matrix Σ^{-1} of the AR(1) process of the latent states is a sparse tridiagonal matrix whose diagonal elements are equal to $0.5a_p + (1+\phi^2)/\tau^2$ with the exception of the first and last diagonal elements which are equal to $0.5a_p + 1/\tau^2$; the super and sub diagonal elements are equal to $-\phi/\tau^2$. We set $\widetilde{M} = \Sigma^{-1}$. Our article uses two adaptive approaches. The first follows an adaptive method based on Garthwaite et al. (2015) to select an ϵ that yields a specified average acceptance probability across all M particles and L is set to some fixed value. We denote this method as HMC-I. The second adaptive approach is to select both ϵ and L based on the adaptive method by Buchholz et al. (2020) and denote it as HMC-II.

Table 1 summarizes the estimation results for the univariate SV model estimated using the PHS, AISIL-HMC, and AISIL-PG methods. The terms PG-batch, PG-seq, and PG-batch-seq for AISIL-PG denote batch estimation, sequential estimation, and a combination of batch and sequential estimation, respectively. Section 2.2.2 discuses sequential estimation using tempering. The estimation methods based on HMC Markov moves are defined similarly. For the PG-batch-seq method, batch estimation is used for the first 80% of the data and sequential estimation is used for the rest. The PHS chain consists of 50000 iterates for burnin and another 5000 iterates used as for inference. All the AISIL estimates are obtained using 10 independent runs, each with $M = 560$ samples, to generate a total of 5600 samples for each algorithm. We set the constant $\text{ESS}_T = 0.8M$, and use the PHS estimates as the "gold standard" to assess the accuracy of the two annealing approaches. The computation was done using a Matlab implementation of the algorithms and 28 CPU-cores of a high performance computer cluster.

Table 1 shows that all the AISIL-PG estimates, both batch and sequential, are very close to the PHS estimates for all parameters even with as few as 150 particles and $R = 10$ Markov move steps. Fig. 2 shows the kernel density estimates of the marginal posteriors of the univariate stochastic volatility parameters as well as the estimates of the log-volatility estimated using the PHS and the AISIL-PG batch and sequential methods with different numbers of particles, and Markov move steps. The density estimates from the AISIL-PG methods are very close to the density estimates of the PHS sampler for all the parameters for all configurations. The posterior mean estimates of the log-volatilities $\boldsymbol{x}_{1:T}$ are also indistinguishable.

Fig. 3 presents the sequential parameter learning in the univariate SV model estimated using PG-seq method. It is clear that all parameter estimates vary a lot in the beginning when the number of observations is small, then stabilize as the number of observations gets larger. This sequential approach can take into account impacts of parameter and model uncertainties on decision making over time, but it is computationally more expensive than the PG-batch approach because the it uses

more resample and Markov move steps.

The estimates of τ^2 and ϕ using the AISIL-HMC-I-batch with $R = 20$ is the closest to the estimates of the PHS compared to AISIL-HMC-I-batch with $R = 10$, AISIL-HMC-II-batch, and AISIL-HMC-II-batch-seq. But in general, they are very close to each other. Fig. 1 shows the kernel density estimates of the marginal posteriors of the univariate SV parameters and the estimates of the log-volatility estimated using the PHS and the AISIL-HMC methods. The figure confirms that the densities estimates of τ^2 and ϕ estimated from AISIL-HMC-I-batch with $R = 20$ are very close to the estimates from the PHS. The posterior mean estimates of the log-volatilities $\boldsymbol{h}_{1:T}$ are very close to the estimates from the PHS. All of the AISIL-HMC methods are always faster than the AISIL-PG methods.

In general, AISIL-PG is more accurate than the AISIL-HMC, but its CPU time is slightly larger. With both AISIL methods, it is also unnecessary to deal with autocorrelation issues of particle MCMC and MCMC in general. However, if we suspect slow convergence for some parameters that are highly correlated with the latent states in the model, it is easy to adapt the hybrid Markov move steps of Gunawan et al. (2020) in our framework. Our strategy is to generate parameters that converge slowly using a PMMH step, with the rest of parameters generated, conditional on the states, using PG step.

The density tempered sequential Monte Carlo samplers of Duan and Fulop (2015) use pseudo marginal Metropolis-Hastings (PMMH) Markov move steps and follow the Pitt et al. (2012) guidelines to set the optimal number of particles in the particle filter such that the variance of log of estimated likelihood is around 1; this requires around 5000 particles. We do not compare our approach with theirs because it is clear that the AISIL-PG approach is more efficient as it only needs 250 particles or less.

Table 1 also presents the estimates and standard error estimates of the log of the marginal likelihood using AISIL-HMC and AISIL-PG methods; the table shows that the estimated standard errors using AISIL-HMC are bigger than for the AISIL-PG.

Another important advantage of the PG-batch-seq and HMC-batch-seq approaches is that it can provide sequential one-step ahead predictive densities of the log returns, and hence prices; this is particularly useful in financial applications. Fig. 4 shows the sequential one-step ahead predictive density for log-return of the US food industry from 22/06/2011 to 11/11/2013 estimated using the PG-batch-seq method. Financial risk measures such as value at risk (VaR) can be computed from the one step ahead predictive densities. The VaR is the most widely used measure of market risk (Holton, 2003). The t-period π%-VaR of a return series, denoted as

$\text{VaR}_{\pi,t}$, is defined by

$$\pi = \Pr\left(y_t \leq \text{VaR}_{\pi,t} | \mathcal{F}_{t-1}\right),$$

where $\mathcal{F}_{t-1}$ is the information set up to time $t-1$. The one-step ahead π%-VaR, $\text{VaR}_{\pi,t+1}$, can be estimated using the πth-quantile of the posterior predictive return distribution at time $t+1$ given the information up to time t. Fig. 5 shows the sequential one-step ahead VaR estimates at risk levels 0.01, 0.05, and 0.10 for the log return of the US food industry from 22/06/2011 to 11/11/2013. The lowest VaR estimates are between 08/08/2011 to 18/08/2011 obtained using the PG-batch-seq method.

Table 1: Results for the univariate SV model estimated using the PHS, AISIL-HMC, and AISIL-PG samplers for US food stock returns data with $T = 3001$. The AISIL results are obtained using 10 independent runs of each algorithm. Time is the time in minutes for one run of the algorithm. The posterior standard deviations of the SV parameters and standard errors of the log of the marginal likelihood estimates are in brackets.

Method	N	L	R	μ	ϕ	τ^2	$\log \widehat{p}(y)$	Time
PHS	100	-	-	−0.4886 (0.2054)	0.9853 (0.0040)	0.0226 (0.0040)	-	513
HMC-I-batch	-	100	10	−0.4934 (0.1992)	0.9839 (0.0043)	0.0252 (0.0041)	−3665.29 (1.11)	20
HMC-I-batch	-	100	20	−0.4888 (0.2007)	0.9849 (0.0039)	0.0234 (0.0039)	−3665.88 (1.03)	38
HMC-II-batch	-	-	20	−0.4959 (0.2006)	0.9839 (0.0045)	0.0253 (0.0051)	−3664.31 (1.92)	27
HMC-II-batch-seq	-	-	20	−0.4966 (0.2058)	0.9850 (0.0041)	0.0235 (0.0043)	−3663.29 (1.40)	55
PG-batch	150	-	10	−0.4996 (0.2096)	0.9850 (0.0040)	0.0231 (0.0040)	−3669.39 (0.21)	42
PG-batch	250	-	10	−0.4986 (0.2106)	0.9851 (0.0041)	0.0229 (0.0040)	−3666.14 (0.34)	57
PG-batch	250	-	20	−0.4797 (0.2059)	0.9850 (0.0041)	0.0229 (0.0041)	−3666.22 (0.32)	112
PG-seq	250	-	10	−0.4866 (0.2009)	0.9849 (0.0042)	0.0230 (0.0041)	−3661.79 (1.00)	270
PG-batch-seq	250	-	10	−0.4867 (0.2017)	0.9849 (0.0040)	0.0232 (0.0041)	−3665.03 (0.48)	135

Figure 1: The kernel density estimates of the marginal posterior densities of the univariate SV parameters and log volatilities for the PHS (PMCMC) and AISIL-HMC samplers. The bottom right panel plots the estimated posterior means of the log volatilities.

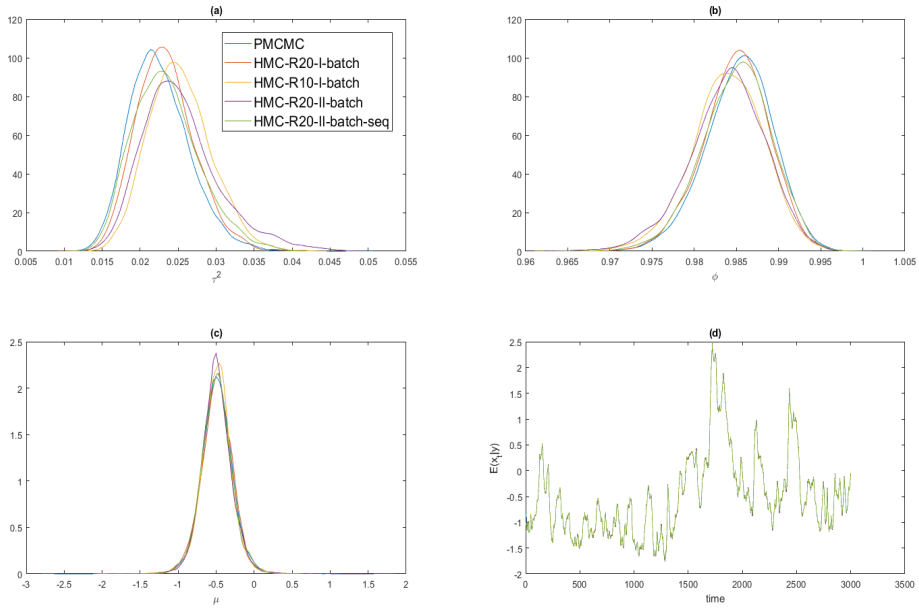


Figure 2: The kernel density estimates of the marginal posterior densities of the univariate SV parameters and log volatilities for the PHS (PMCMC) and AISIL-PG samplers. The bottom right panel plots the estimated posterior means of the log volatilities.

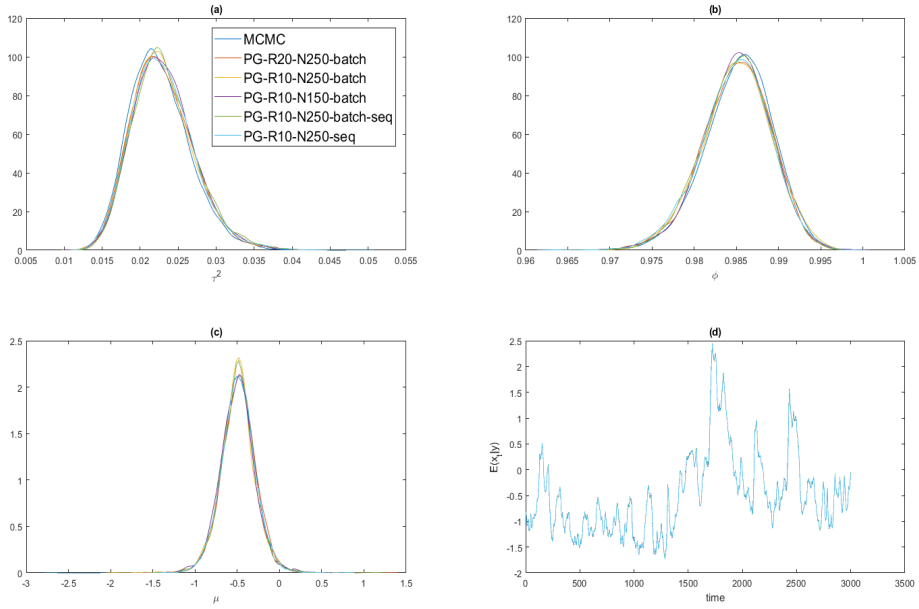


Figure 3: Parameter learning in the Univariate SV Model estimated using the PG-seq method

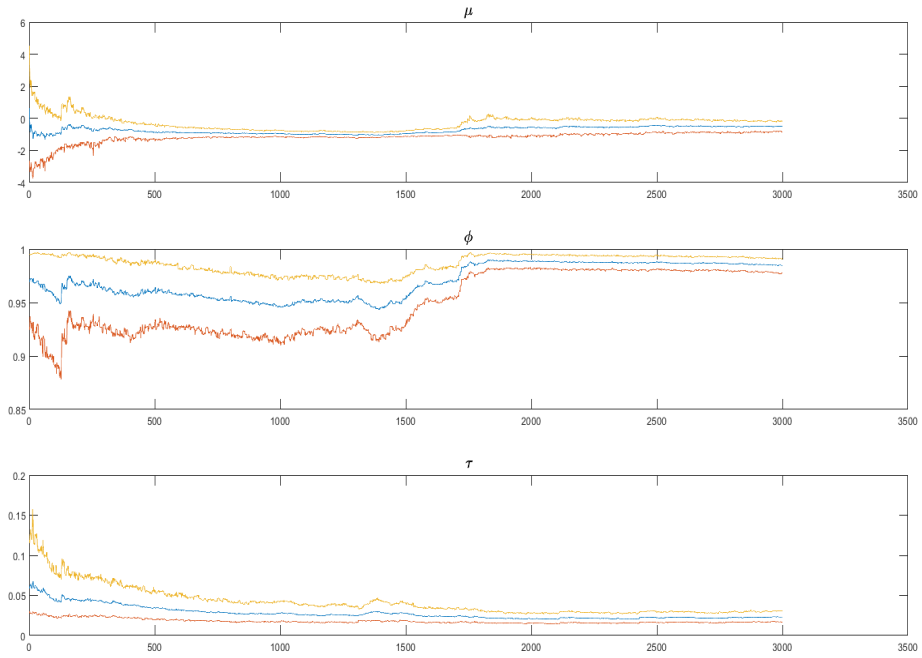


Figure 4: Sequential one-step ahead predictive density estimates for log return of the US food industry from 22/06/2011 to 11/11/2013 estimated using the PG-batch-seq method

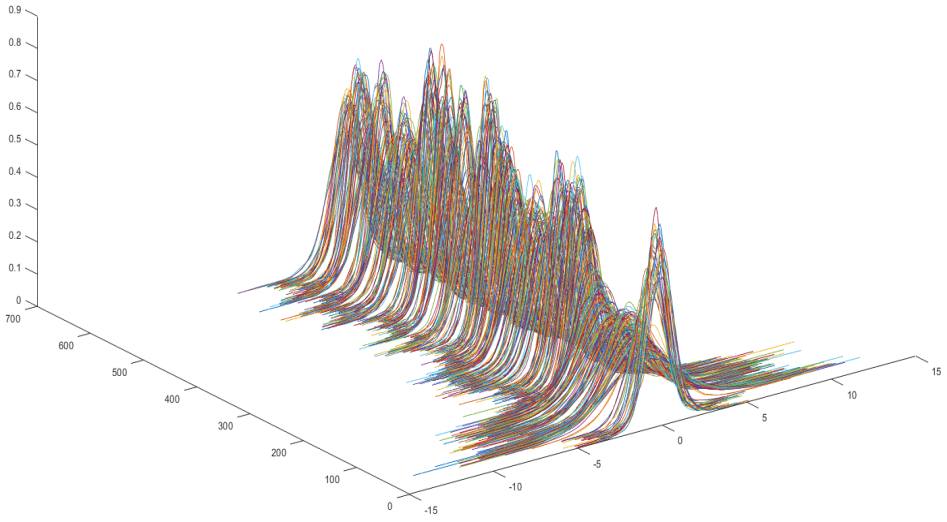
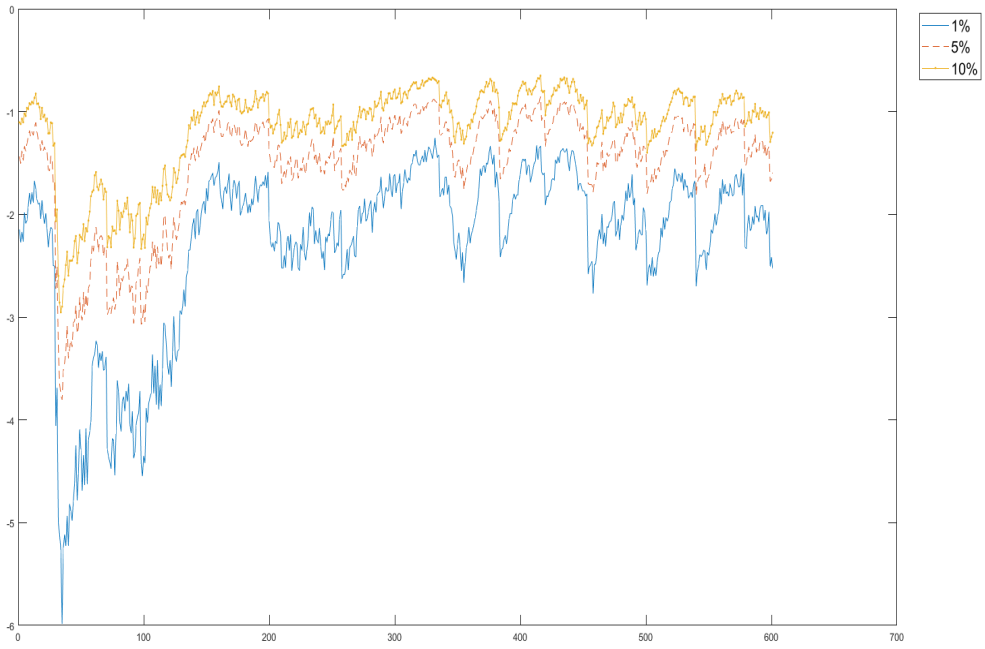


Figure 5: Sequential one-step ahead Value at Risk (VaR) estimates for log return of the US food industry from 22/06/2011 to 11/11/2013 estimated using the PG-batch-seq method



4.2 Time Series Diagnostics

The sequential approach also allows us to efficiently estimate the sequence of standard normal random variables used to form the goodness of fit statistics to test for the adequacy of a general time series state space model (Gerlach et al., 1998). Let $y_1, ..., y_T$ be a sequence of time series observations generated by the continuous random variables $Y_1, ..., Y_T$. If the correct model is fitted to the data, then the sequence

$$u_t = p\left(Y_t \leq y_t | \boldsymbol{y}_{1:t-1}\right), \ t = 1, ..., T, \tag{6}$$

is a realization of independent uniform $U(0,1)$ random variables. A sequence of independent standard normal random variables $N(0,1)$ is obtained by setting $v_t = \Phi^{-1}(u_t)$, where Φ is the standard normal cumulative distribution function. To obtain u_t, it is necessary to integrate out the parameters and the latent states by evaluating

$$u_t = \int p\left(Y_t \leq y_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}, x_t\right) p\left(x_t | x_{t-1}, \boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}, \boldsymbol{x}_{1:t-1} | \boldsymbol{y}_{1:t-1}\right) d\boldsymbol{\theta} dx_t. \tag{7}$$

For many statistical models, such as the stochastic volatility model, the integral in Eq. (7) cannot be evaluated analytically. One way to estimate u_t is to use MCMC

or particle MCMC methods. Given a sample of MCMC draws, $\boldsymbol{\theta}_m$ and $x_{m,t}$, for $m = 1, ..., M$, an estimate of u_t is

$$\widehat{u}_t = \frac{1}{M} \sum_{m=1}^{M} p\left(Y_t \leq y_t | \boldsymbol{y}_{1:t-1}, \boldsymbol{\theta}_m, x_{m,t}\right).$$

The sequence $\widehat{v}_t$ is obtained by setting $\widehat{v}_t = \Phi^{-1}\left(\widehat{u}_t\right)$ for $t = 1, ..., T$. However, MCMC can be very time-consuming as it is necessary to repeatedly construct the full Markov chain for each time period t. The sequential approach, PG-seq, can be used to estimate u_t efficiently. If the model is correct, the sequence of u_t is uniform and independent and the sequence of v_t is standard normal and independent. We apply the approach to test for model adequacy using simulated and real data. The simulated data uses the parameter values $\phi = 0.98$, $\mu = -0.48$, $\tau^2 = 0.02$, and $T = 1000$ observations. A single run of the sequential approach, PG-seq, with $M = 560$ samples is used to obtain the estimate of $\widehat{u}_t$ and $\widehat{v}_t$, for $t = 1, ..., T$. Fig. 6 shows the diagnostic plots for the series $\widehat{v}_t$ for $t = 1, ..., T$. The autocorrelation plot in the figure suggests that there is no autocorrelation in the v_t series. The QQ-plot shows evidence of normality in the $\widehat{v}_t$ series. The Anderson-Darling test (Stephens, 1974) can also be used to test the null hypothesis that the series v_t is normally distributed; the null hypothesis is not rejected at the 5% level of significance (p-value=0.65). This is expected because the dataset is simulated from the SV model. We now apply the methods to a sample of daily US food industry stock returns as in previous section. Fig. 7 shows the diagnostic plots for the series $\widehat{v}_t$ for this real data. The autocorrelation plot in the figure suggests that the v_t series is uncorrelated; (2) the quantiles of $\widehat{v}_t$ are similar to the quantiles of standard normal distribution, except in both the tails. The Anderson-Darling test for normality of the v_t series is rejected at the 5% level of significance (p-value=0.00).

Figure 6: Simulated data, left panel – the autocorrelation plot of the $\widehat{v}_t$ series; right panel – the QQ-plot of the $\widehat{v}_t$ series versus the standard normal

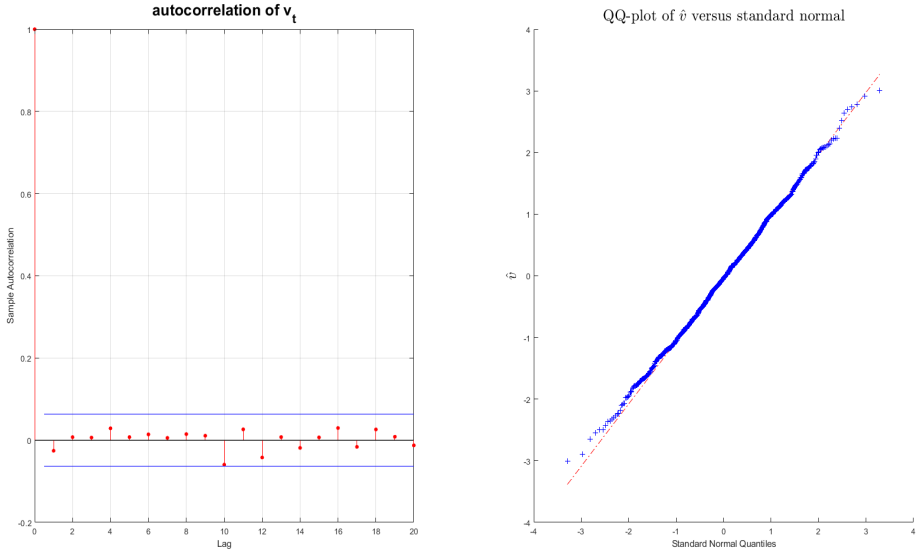
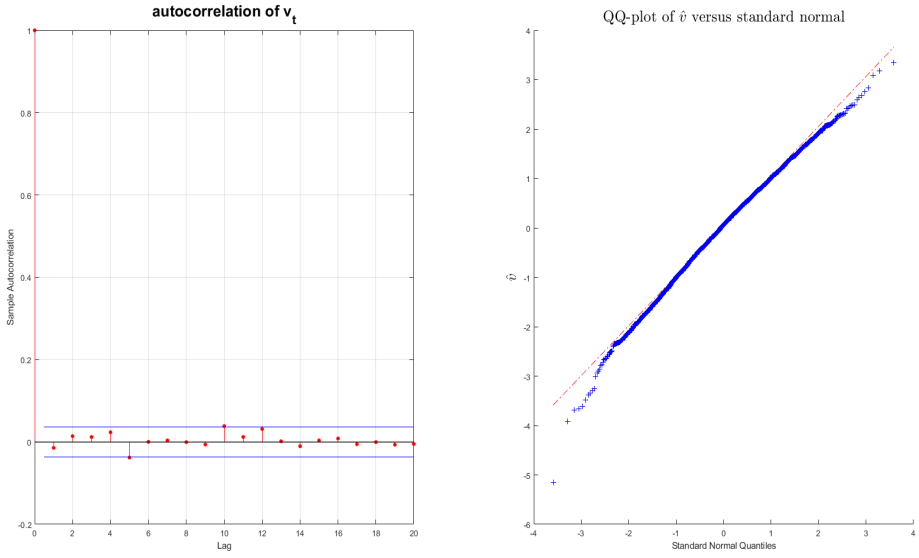


Figure 7: US Food stock returns data, left panel – the autocorrelation plot of the $\widehat{v}_t$ series; right panel – the QQ-plot of the $\widehat{v}_t$ series versus the standard normal



4.3 SV Model with Outliers

This section uses data simulated from a standard SV model contaminated by outliers to study the performance of the proposed sequential approach PG-seq with tempering (denoted by PG-seqT) and compares it to the standard sequential approach without

tempering (denoted by PG-seqNT) as in Chopin et al. (2013)[4]. Several scenarios are considered. In each scenario, a dataset is generated with $T = 1000$ observations using parameter values $\phi = 0.98$, $\mu = -0.48$, and $\tau^2 = 0.02$. Then, the generated observations are contaminated with noise $B_t \times \beta_t$, where B_t is a Bernoulli indicator with parameter 0.01 and β_t is normally distributed with mean zero and standard deviation κ. The parameter κ controls the degree of contaminations and is set to $(5, 10, 15, 25)$. A single dataset is simulated in each case. The estimates of the PG-batch method are used as the "gold standard" to assess the accuracy of the PG-seqT and PG-seqNT approaches. All of the AISIL estimates are obtained using 10 independent runs, each with $M = 560$ samples. For the PG-batch and PG-seqT, ESS_T is set to $0.8M$.

Table 2 summarizes the simulation results for the SV models with different degrees of data contamination $\kappa = (5, 10, 15, 25)$. The first three rows report the standard error of the posterior mean estimates of the SV model parameters over the 10 runs of the algorithms; the fourth and fifth rows report the mean and standard error of the log of the marginal likelihood estimates. The PG-seqT and PG-batch provide stable results for all values of κ, but the performance of PG-seqNT deteriorates as κ increases. When $\kappa = 25$, the standard error of τ^2 from PG-seqNT is 2.84 and 2.09 times larger than from PG-seqT and PG-batch, respectively. The Monte Carlo error of the PG-seqNT increases as outliers become larger in magnitude. The outliers lead to highly variable weights in the reweighting step of the PG-seqNT algorithm and very low effective sample size (ESS) of the particles. More importantly, the Monte Carlo error of the log of the marginal likelihood estimates seem to deteriorate much faster than the parameter estimates as κ increases. When $\kappa = 25$, the standard error of the log of the marginal likelihood estimate from PG-seqNT is 28.05 and 21.49 times larger than from PG-seqT and PG-batch, respectively. The log of the marginal likelihood estimates from the PG-seqNT are substantially different to the estimates from PG-seqT and PG-batch for $\kappa = 10, 15, 25$.

Fig. 8 shows the normalised ESS over time obtained from a run of PG-seqNT for the SV model with $\kappa = 25$. The normalised ESS is close to zero when the data point is the outlier. This confirms that the proposed PG-seqT is more stable than the standard PG-seqNT as it ensures the ESS stays close to ESS_T. Fig. 9 (top panel) shows the number of Markov steps over time obtained from a run of PG-seqT algorithm. The figure shows that a larger number of Markov move steps are taken when the data is an outlier. The middle panel of Fig. 9 shows the absolute value of

[4]Note that the SMC^2 in Chopin et al. (2013) integrates out the latent states, $\boldsymbol{x}_{1:t}$, up to time t and only targets the posterior density of parameters $\boldsymbol{\theta}$. The standard sequential approach without tempering considered here targets the joint posterior density of parameters $\boldsymbol{\theta}$ and latent states, $\boldsymbol{x}_{1:t}$, up to time t

the log of $\widehat{p}(y_t|\boldsymbol{y}_{1:t-1})$ for $t = 1, .., T$. This figure can be used to identify outliers in the data by investigating the unusually large value of the absolute value of the log of $\widehat{p}(y_t|\boldsymbol{y}_{1:t-1})$ for $t = 1, ..., T$.

Figure 8: Top panel: The normalised ESS (ESS/M) over time obtained from a run of PG-seqNT algorithm for SV model with $\kappa = 25$; Bottom panel: The series B_t for $t = 1, ..., T$ and $B_t = 1$ if the observation is an outlier and 0, otherwise.

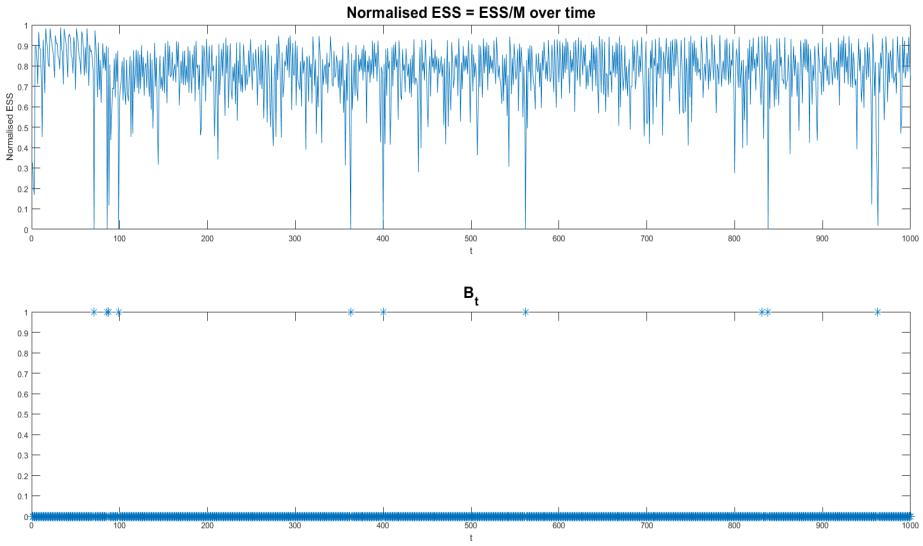


Figure 9: Top panel: The number of Markov steps over time obtained from a run of PG-seqT algorithm for the SV model with $\kappa = 25$; Middle panel: the absolute value of the log of $\widehat{p}(y_t|\boldsymbol{y}_{1:t-1})$ for $t = 1, .., T$; Bottom panel: The B_t, $t = 1, ..., T$ series; $B_t = 1$ if the observation is an outlier and 0 otherwise.

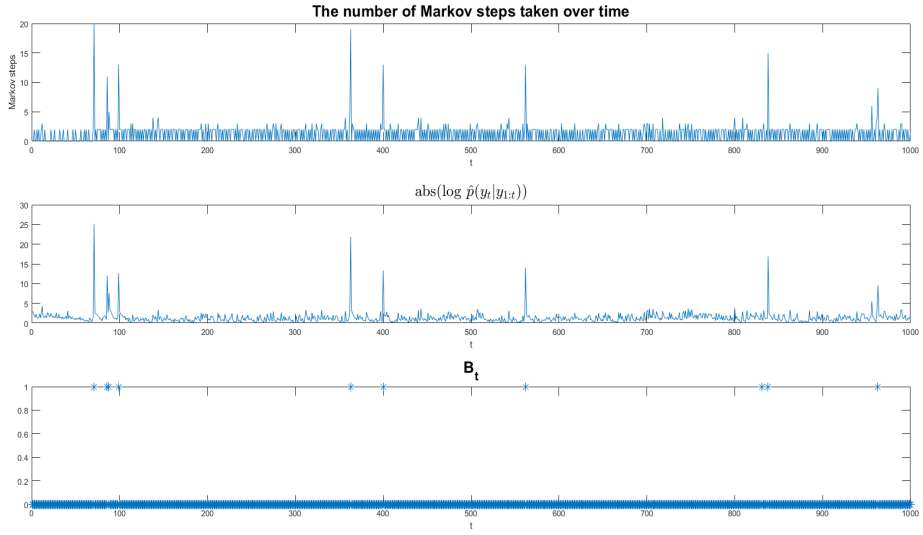


Table 2: Simulation results for the SV models with different degrees of data contamination $\kappa = (5, 10, 15, 25)$. The results are computed from 10 independent runs of each algorithm. The first three rows report the standard error of the posterior mean estimates of the SV parameters (μ, ϕ, τ^2) over the 10 independent runs. The fourth and fifth rows report the mean and standard error of the log of the marginal likelihood estimates over 10 runs.

	PG-seqT				PG-seqNT				PG-batch			
κ	5	10	15	25	5	10	15	25	5	10	15	25
μ std err.	0.0100	0.0062	0.0058	0.0048	0.0094	0.0051	0.0081	0.0047	0.0040	0.0026	0.0025	0.0020
ϕ std err.	0.0006	0.0023	0.0014	0.0021	0.0007	0.0022	0.0018	0.0039	0.0003	0.0040	0.0010	0.0025
τ^2 std err.	0.0011	0.0058	0.0027	0.0069	0.0009	0.0063	0.0034	0.0196	0.0004	0.0087	0.0028	0.0094
$\log \widehat{p}(\boldsymbol{y})$ mean	-1381.15	-1431.94	-1276.53	-1348.71	-1381.83	-1457.13	-1289.08	-1459.16	-1382.53	-1432.79	-1278.11	-1349.54
$\log \widehat{p}(\boldsymbol{y})$ std err.	0.3812	0.5907	0.3311	0.5871	0.5808	13.1455	7.2123	16.4654	0.1548	0.3069	0.3321	0.7661

4.4 Markov Switching Stochastic Volatility Models

This section demonstrates the flexibility of PG-seqT by applying it to the two state Markov switching SV model

$$y_t = \mu_{s_t}^p + \exp\left(x_t/2\right)\epsilon_t,\ x_t = \mu + \phi\left(x_{t-1} - \mu\right) + \tau\eta_t,\ x_1 \sim N\left(\mu, \tau^2/\left(1-\phi^2\right)\right);$$

$\epsilon_t \sim N\left(0,1\right)$, $\eta_t \sim N\left(0,1\right)$, $\boldsymbol{s}_{1:T} = \left(s_1, ..., s_T\right)^\intercal$, $s_t = 1, 2$ and $s_t = k$ indicates that y_t is from regime k. This model is similar to that in Chib (1998).

The one-step ahead transition probability matrix for s_t is

$$P = \begin{bmatrix} p_{11} & p_{12} \\ p_{21} & p_{22} \end{bmatrix},$$

where $p_{lk} = \Pr\left(s_t = k | s_{t-1} = l\right)$ is the probability of moving from regime l at time $t-1$ to regime k at time t. The prior for μ_1^p and μ_2^p is $N\left(0, 100\right)$; the prior p_{11} and p_{22} is $Beta\left(4,1\right)$. The same priors as in Section 3 are used for the other SV parameters.

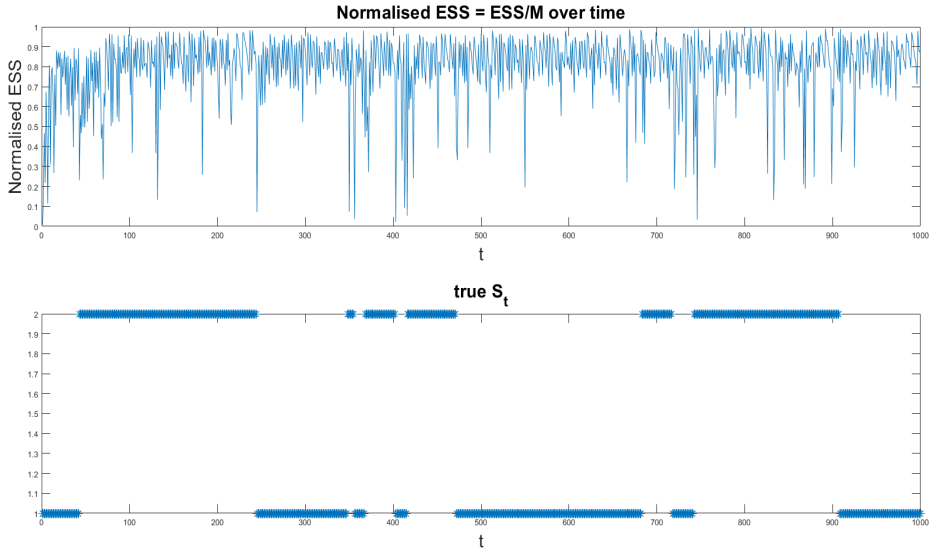
The performance of PG-seqT is compared to PG-seqNT and PG-batch for this model. A dataset is generated with $T = 1000$ observations using parameter values $\phi = 0.98$, $\mu = -0.48$, $\tau^2 = 0.02$, $\mu_1 = -1$, $\mu_2 = 1$, and $p_{kk} = 0.99$ for $k = 1, 2$. The PG-batch method is used as the "gold standard" to assess the accuracy of the PG-seqT and PG-seqNT. All the AISIL estimates are obtained using 10 independent runs, each with $M = 560$ samples. For the PG-batch and PG-seqT, the ESS_T is set to $0.8M$.

Table 3 summarizes the simulation results for the Markov switching SV model. The first seven rows report the standard errors of the posterior mean estimates of the model parameters over the 10 runs; the eighth and ninth rows report the mean and standard error of the log of the marginal likelihood estimates. In general, the standard error of the posterior mean estimates obtained by PG-seqT and PG-batch is smaller than PG-seqNT. The standard error of the log of the marginal likelihood estimates from PG-seqNT is about 3 times larger than from PG-seqT and PG-batch. Fig. 8 shows the normalised ESS over time obtained from a run of PG-seqNT algorithm. It is clear that the normalised ESS is quite small when the state is changed from 1 to 2 or from 2 to 1.

Table 3: Simulation results for the Markov switching SV models. The results are computed from 10 independent runs of each algorithm. The first seven rows report the standard error of the posterior mean estimates of the SV parameters $(\mu_1^p, \mu_2^p, \mu, \phi, \tau^2)$ over the 10 independent runs. The eighth and ninth rows report the mean and standard error of the log of the marginal likelihood estimates over 10 runs.

	PG-seqT	PG-seqNT	PG-batch
μ_1^p std err.	0.0013	0.0013	0.0008
μ_2^p std err.	0.0015	0.0025	0.0013
p_{11} std err.	0.0003	0.0003	0.0001
p_{22} std err.	0.0004	0.0002	0.0003
μ std err.	0.0084	0.0066	0.0033
ϕ std err.	0.0007	0.0008	0.0004
τ^2 std err.	0.0005	0.0005	0.0005
$\log \widehat{p}(\boldsymbol{y})$ mean	-1434.83	-1435.20	-1437.83
$\log \widehat{p}(\boldsymbol{y})$ std err.	0.5100	1.5235	0.5071

Figure 10: Top panel: The normalised $ESS := (ESS/M)$ over time obtained from the PG-seqNT algorithm for the Markov switching SV model; Bottom panel: the true series s_t for $t = 1, ..., T$



5 The Multivariate Factor Stochastic Volatility Model

5.1 Model

The factor SV model is a parsimonious multivariate stochastic volatility model that is often used to model a vector of stock returns; see, for example, Chib et al. (2006)

and Kastner et al. (2017). It is a high dimensional state space model having a large number of parameters and a large number of latent states.

Suppose that $\boldsymbol{P}_t$ is a $S \times 1$ vector of daily stock prices and define $\boldsymbol{y}_t \coloneqq \log \boldsymbol{P}_t - \log \boldsymbol{P}_{t-1}$ as the vector of stock returns. We model $\boldsymbol{y}_t$ as a factor SV model

$$\boldsymbol{y}_t = \beta \boldsymbol{f}_t + V_t^{\frac{1}{2}} \boldsymbol{\epsilon}_t, \ (t = 1, ..., T), \tag{8}$$

where $\boldsymbol{f}_t$ is a $K \times 1$ vector of latent factors (with $K \lll S$), and β is a $S \times K$ factor loading matrix of unknown parameters. The model the latent factor is $\boldsymbol{f}_t \sim N(0, D_t)$ with $\boldsymbol{\epsilon}_t \sim N(0, I)$. The time varying variance matrices V_t and D_t depend on the unobserved random variables $\boldsymbol{h}_t = (h_{1t}, ..., h_{St})$ and $\boldsymbol{\lambda}_t = (\lambda_{1t}, ..., \lambda_{Kt})$ such that

$$V_t \coloneqq \text{diag}\left\{\exp(h_{1t}), ..., \exp(h_{St})\right\}, \ D_t \coloneqq \text{diag}\left\{\exp(\lambda_{1t}), ..., \exp(\lambda_{Kt})\right\}.$$

Each of the log volatilites λ_{kt} and h_{st} is assumed to follow an independent first order autoregressive process, with

$$h_{st} - \mu_{\epsilon s} = \phi_{\epsilon s}(h_{st-1} - \mu_{\epsilon s}) + \eta_{\epsilon st}, \eta_{\epsilon st} \sim N\left(0, \tau_{\epsilon s}^2\right), s = 1, ..., S \tag{9}$$

and

$$\lambda_{kt} = \phi_{fk}\lambda_{kt-1} + \eta_{fkt}, \eta_{fkt} \sim N\left(0, \tau_{fk}^2\right), k = 1, ..., K. \tag{10}$$

For $s = 1, ..., S$ and $k = 1, ..., K$, we choose the priors for the persistence parameters $\phi_{\epsilon s}$ and ϕ_{fk}, the priors $\tau_{\epsilon s}^2$, τ_{fk}^2, and $\mu_{\epsilon s}$ as in Section 3. For every unrestricted element of the factor loadings matrix β, we follow Kastner et al. (2017) and choose independent Gaussian distributions $N(0, 1)$. These prior densities cover most possible values in practice.

Section S3 of the supplement discusses parameterisation and identification issues regarding the factor loading matrix β and the latent factors $\boldsymbol{f}_t$.

Conditional Independence in the factor SV model

The key to making the estimation of the factor SV model tractable is that given the values of $(\boldsymbol{y}_{1:T}, \boldsymbol{f}_{1:T}, \beta)$, the factor model in Eq. (8) separates into $S + K$ independent components consisting of K univariate SV models for the latent factors with f_{kt} the tth 'observation' of the kth factor univariate SV model and S univariate SV models for the idiosyncratic errors with ϵ_{st} the tth 'observation' on the sth idiosyncratic error SV model. Section 5.2 discusses the AISIL-HMC method for the factor SV model and Section 5.3 discusses the AISIL-PG method.

5.2 Application of the AISIL-HMC method to the Multivariate Factor Stochastic Volatility Model

This section discusses the application of the AISIL-HMC method to the multivariate factor SV model described in Section 5.1 in a batch context. We first define the appropriate sequence of intermediate target densities,

$$\begin{aligned}
\xi_{a_p}\left(\boldsymbol{\theta}, \boldsymbol{h}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{f}_{1:T}\right) &\propto p\left(\boldsymbol{y}_{1:T}|\boldsymbol{f}_{1:T}, \boldsymbol{h}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\theta}\right)^{a_p} p\left(\boldsymbol{f}_{1:T}|\boldsymbol{h}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\theta}\right) \\
&\quad \times p\left(\boldsymbol{\lambda}_{1:T}, \boldsymbol{h}_{1:T}|\boldsymbol{\theta}\right) p\left(\boldsymbol{\theta}\right), \\
\text{where} \quad \boldsymbol{\theta} &= \left\{\boldsymbol{\theta}_{\epsilon s} = \left\{\mu_{\epsilon s}, \phi_{\epsilon s}, \tau_{\epsilon s}^2\right\}_{s=1}^{S}, \boldsymbol{\theta}_{fk} = \left\{\phi_{fk}, \tau_{fk}^2\right\}_{k=1}^{K}, \beta\right\}, \\
p\left(\boldsymbol{y}_{1:T}|\boldsymbol{f}_{1:T}, \boldsymbol{h}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{\theta}\right)^{a_p} &= \prod_{t=1}^{T}\prod_{s=1}^{S} p\left(y_{st}|\boldsymbol{\beta}_s, \boldsymbol{f}_t, h_{st}\right)^{a_p}, \\
p\left(\boldsymbol{f}_{1:T}|\boldsymbol{\lambda}_{1:T}\right) &= \prod_{t=1}^{T}\prod_{k=1}^{K} p\left(f_{kt}|\lambda_{kt}\right), \\
p\left(\boldsymbol{\lambda}_{1:T}|\boldsymbol{\theta}_f\right) &= \prod_{k=1}^{K}\prod_{t=2}^{T} p\left(\lambda_{kt}|\lambda_{kt-1}, \boldsymbol{\theta}_{fk}\right) p\left(\lambda_{k1}|\boldsymbol{\theta}_{fk}\right), \\
p\left(\boldsymbol{h}_{1:T}|\boldsymbol{\theta}_\epsilon\right) &= \prod_{s=1}^{S}\prod_{t=2}^{T} p\left(h_{st}|h_{st-1}, \boldsymbol{\theta}_{\epsilon s}\right) p\left(h_{s1}|\boldsymbol{\theta}_{\epsilon s}\right).
\end{aligned} \tag{11}$$

Algorithm 3 in Appendix B gives the Markov move step based on Hamiltonian Monte Carlo for the multivariate factor SV model with the target densities $\xi_{a_p}\left(\boldsymbol{\theta}, \boldsymbol{h}_{1:T}, \boldsymbol{\lambda}_{1:T}, \boldsymbol{f}_{1:T}\right)$. Steps 1 to 4 update the idiosyncratic log-volatilities parameters $(\mu_{\epsilon s}, \phi_{\epsilon s}, \tau_{\epsilon s}^2)$ for $s = 1, ..., S$, factor log-volatilities parameters (ϕ_{fk}, τ_{fk}^2) for $k = 1, ..., K$, the factor loading matrix β and the latent factors $\boldsymbol{f}_t$, respectively. Section S2.2 discusses the updating the idiosyncratic and factor log-volatilities parameters. Section S3 discusses sampling of the factor loading matrix β and the latent factors $\boldsymbol{f}_t$. Steps 5 and 6 update S idiosyncratic log-volatilities, $\boldsymbol{h}_{s,1:T}$ and K factor log-volatilities $\boldsymbol{\lambda}_{k,1:T}$ using Hamiltonian Monte Carlo, respectively. Section S2.1 presents the details on updating the idiosyncratic and factor log-volatilities.

5.3 Application of the AISIL-PG method to the Multivariate Factor Stochastic Volatility Model

This section discusses the application of the AISIL-PG method to a multivariate factor SV model in a batch context. Similarly to Section 5.2, Eq. (11) gives the appropriate sequence of intermediate target densities. Appendix A discusses the augmented intermediate target density for the multivariate factor SV model.

Algorithm 4 in Appendix C describes the AISIL-PG Markov moves for the multivariate factor stochastic volatility models. Steps 1 to 4 update the idiosyncratic log-volatilities parameters $(\mu_{\epsilon s}, \phi_{\epsilon s}, \tau^2_{\epsilon s})$ for $s = 1, ..., S$, factor log-volatilities parameters (ϕ_{fk}, τ^2_{fk}) for $k = 1, ..., K$, the factor loading matrix β and the latent factors $\boldsymbol{f}_t$, respectively. Section S2.2 discusses the details to update the idiosyncratic and factor log-volatilities parameters. Section S3 shows how to sample the factor loading matrix β and the latent factors $\boldsymbol{f}_t$. Steps 5 and 7 update the $\boldsymbol{U}^{-\boldsymbol{j}_{\epsilon s,1:T}}_{\epsilon s,1:T}$ for $s = 1, ..., S$ and $\boldsymbol{U}^{-\boldsymbol{j}_{\epsilon fk,1:T}}_{\epsilon fk,1:T}$ for $k = 1, ..., K$, respectively, by running $(S + K)$ conditional sequential Monte Carlo algorithm. Steps 6 and 8 update the indices $\boldsymbol{j}_{\epsilon s,1:T}$ and $\boldsymbol{j}_{fk,1:T}$, by running $(S + K)$ backward simulation algorithms.

6 Examples: Multivariate Factor Stochastic Volatility Model

This section compares the performance of AISIL-PG, AISIL-HMC and PHS by applying them to the multivariate factor SV model discussed in Section 5.1 using one factor. We then estimate models with 1 to 6 factors and show how to use the marginal likelihood to select the number of factors. A sample of daily returns for $S = 26$ value weighted industry portfolios is used, from December 11th, 2001 to 29th November 2005, a total of 1000 observations. The data was obtained from the Kenneth French website with the industry portfolios used listed in Section S7. We consider both batch and sequential estimation for AISIL-PG method. As in Section 4.1, for AISIL-PG, batch estimation is denoted by PG-batch and PG-batch-seq for a combination of batch and sequential estimation. For the PG-batch-seq method, batch estimation is used for the first 975 observations and sequential estimation is used for the last 25 observations. The PHS is regarded as the "gold standard"; it was run for 50000 iterates with another 5000 iterates used as burn in.

The density tempered SMC of Duan and Fulop (2015) uses the PMMH Markov steps and follows the Pitt et al. (2012) guidelines to set the optimal number of particles in the particle filter to ensure that the variance of the log of the estimated likelihood is around 1. The PMMH Markov step generates the parameters with latent factors, factor and idiosyncratic log-volatilities integrated out resulting in an $(S + K)$ dimensional state space model. The tempered measurement density at the pth stage is

$$\left\{ N\left(\boldsymbol{y}_t; 0, \beta D_t \beta^{'} + V_t\right) \right\}^{a_p},$$

Eq. (9) gives the state transition densities for the idiosyncratic log-volatilities ($s =$

$1, \ldots, S$) and Eq. (10) gives the state transition equations for the factor log-volatilities ($k = 1, \ldots, K$).

Table 4 shows the variance of log of the estimated likelihood for different numbers of particles evaluated at posterior means of the parameters obtained using the PHS of Gunawan et al. (2020) with $a_p = 1$. It shows that even with 5000 particles, the PMMH Markov step would get stuck. Deligiannidis et al. (2018) proposed the correlated PMMH method and it is possible to implement it in the Markov move step instead of the standard PMMH method of Andrieu et al. (2010). The correlated PMMH correlates the random numbers used in constructing the estimators of the likelihood at current and proposed values of the parameters and sets the correlation very close to 1 to reduce the variance of the difference in the logs of estimated likelihoods at the current and proposed values of the parameters appearing in the Metropolis-Hastings (MH) acceptance ratio. Deligiannidis et al. (2018) show that the correlated PMMH can be much more efficient and can significantly reduce the number of particles required by the standard PMMH approach when the dimension of the latent states is small. However, the current example considers a high dimensional latent state vector in the one factor-Factor SV model. Mendes et al. (2020) found that it is very challenging to preserve the correlation between the logs of the estimated likelihoods for such a high dimensional state space model. The Markov move based on the correlated PMMH approach will also get stuck unless enough particles are used to ensure the variance of the log of the estimated likelihood is around 1. A second drawback of the PMMH Markov move step as in Duan and Fulop (2015) is that the dimension of the parameter space in the factor SV model is large making it very hard to implement the PMMH Markov step efficiently; this makes it very difficult to obtain good proposals for the parameters, because the first and second derivatives of log of the estimated likelihood with respect to the parameters are not available analytically and can only be estimated. Sherlock et al. (2015) note that in general it is even more difficult to obtain accurate estimate of gradient of the log of the estimated likelihood than it is to obtain accurate estimates of the log of the estimated likelihood. The random walk proposal is easy to implement, but is very inefficient in high dimensions.

Table 4: The variance of the log of estimated likelihood for the PMMH step for different numbers of particles for the US stock returns dataset; $T = 1000$, $S = 26$, and $K = 1$ with the tempered sequence a_p set to 1, evaluated at the posterior means of the parameters obtained from the PHS of Gunawan et al. (2020). Time is the time in seconds to compute the log of the estimated likelihood once.

Number of Particles	Variance of log of estimated likelihood	Time
250	2198.72	4.86
500	1164.51	9.88
1000	813.53	20.13
2500	439.05	50.43
5000	345.85	99.24

The AISIL estimates are obtained using 10 independent runs with $M = 280$ samples to generate a total of 2800 samples for each algorithm. For the AISIL methods, $\text{ESS}_T = 0.8M$. For the AISIL-HMC method, the number of leapfrog L and the step size ϵ are set as in the adaptive approach proposed by Buchholz et al. (2020), and set $R = 20$; for the AISIL-PG method we set $R = 10$, $N = 250$. The computation is done using a Matlab implementation of the algorithms and 28 CPU-cores of a high performance computer cluster.

Table 5 shows the estimates of some of the parameters of the factor SV model estimated using the PHS, AISIL-HMC-batch, and the AISIL-PG methods with both batch and sequential methods. We found that all of the AISIL-PG and AISIL-HMC estimates are very close to the PHS estimates for all the parameters. Figures S2 and S3 show the kernel density estimates of the marginal posteriors of four of the $\phi_{\epsilon s}$ and $\tau^2_{\epsilon s}$ parameters estimated using the PHS, AISIL-PG-batch, AISIL-PG-batch-seq and AISIL-HMC-batch methods. The two figures confirm that the density estimates from the AISIL-PG and AISIL-HMC are close to the density estimates from the PHS. Figure S1 shows the posterior mean estimates of the log-volatilities for $t = 1, ..., T$ for some of the stock returns. The posterior mean estimates of the log-volatilities estimated using both AISIL methods are close to the PHS estimates.

We note that with the proposed AISIL methods, it is unnecessary to deal with the autocorrelation issues of PMCMC and MCMC samplers in general. Mendes et al. (2020) and Gunawan et al. (2020) found that it is necessary to use a combination of PMMH and PG sampler (PHS) to reduce the autocorrelation in the Markov chain in the factor SV model. We show that the Gibbs-type Markov move steps employed by proposed AISIL methods give accurate and very similar results to the PHS. The AISIL-HMC-batch is about two times faster than AISIL-PG-batch. However, a Markov move based on HMC is only feasible in models where the first order derivatives can be computed in closed form. Further, if we suspect that some

parameters converge slowly because they are highly correlated with the latent states in the model, then it is straightforward to adapt the hybrid PMMH and PG Markov move steps of Gunawan et al. (2020) in our framework. Our strategy is to generate parameters that are subject to slow convergence using a PMMH step and the rest of the parameters are generated conditional on the states using PG steps. The Markov move based on the PHS is computationally more expensive than the Markov move based on PG only because it is necessary to run the particle filter twice at each iteration.

Table 5 also shows the estimates of the log of the marginal likelihood for the one factor model. We can clearly see that the estimated standard errors of the estimates of the log of the marginal likelihood estimated using AISIL-HMC are bigger than for the AISIL-PG.

As in the univariate case, it is also possible to obtain sequential one-step ahead predictive densities of the portfolio return at time $t+1$ given the information up to time t, $\boldsymbol{y}_{p,t+1|t} = \boldsymbol{\gamma}^{\intercal}\boldsymbol{y}_{t+1|t}$, where $\boldsymbol{\gamma}$ is the $S \times 1$ vector of portfolio weights. Suppose, for example, that we have an equally weighted portfolio. Figure 11 shows the sequential one-step ahead predictive densities of the equally weighted portfolio of US industry stock returns from 25/10/2005 to 29/11/2005 estimated using AISIL-PG. Using the multivariate factor SV model, we can also estimate the sequential one-step ahead predictive correlation between two asset returns over time. The full predictive correlation matrix at time $t+1$ given the information up to time t is

$$\Gamma_{t+1|t} = \text{diag}\left(\Sigma_{t+1|t}\right)^{-\frac{1}{2}} \Sigma_{t+1|t} \text{diag}\left(\Sigma_{t+1|t}\right)^{-\frac{1}{2}},$$

where
$\Sigma_{t+1|t} = \beta_{t+1|t}^{T} D_{t+1|t} \beta_{t+1|t} + V_{t+1|t}$, $V_{t+1|t} = \text{diag}\left\{\exp\left(h_{1,t+1|t}\right), ..., \exp\left(h_{S,t+1|t}\right)\right\}$,
$D_{t+1|t} = \text{diag}\left\{\exp\left(\lambda_{1,t+1|t}\right), ..., \exp\left(\lambda_{K,t+1|t}\right)\right\}$.

Figure 12 shows the sequential one-step ahead predictive correlation estimates between the banking and automobiles industries and the banking and food industries from 25/10/2005 to 29/11/2005, estimated using AISIL-PG. The figure shows that the food and automobiles industries are positively correlated with the banking industry from 25/10/2005 to 29/11/2005, with the time varying correlation ranging from 0.61 to 0.76. Another important task in portfolio management is determining the portfolio VaR at time $t+1$ given information up to time t, or $VaR_{\pi,t+1|t}$. The one-step ahead π%-VaR is defined as

$$VaR_{p,\pi,t+1} := \sqrt{\boldsymbol{\gamma}^{T}\Sigma_{t+1|t}\boldsymbol{\gamma}} F_{p,t+1|t}^{-1}(\pi),$$

where $F^{-1}_{p,t+1|t}(\pi)$ is the π-th percentile of the distribution function of the one-step ahead predicted portfolio return at time $t+1$ given the information up to time t, with $\boldsymbol{y}_{p,t+1|t} = \boldsymbol{\gamma}^T \boldsymbol{y}_{t+1|t}$. Figure 12 shows the sequential one-step ahead VaR estimates at risk level 0.05 for the equally weighted portfolio of US industry stock return from 25/10/2005 to 29/11/2005 estimated using AISIL-PG.

Finally, the log of the marginal likelihood estimates can be used to select the number of factors. We estimated factor models with 1-6 factors using the AISIL-PG-batch method with an unrestricted factor loading matrix β. Table 6 gives the logs of the marginal likelihoods for the six models, with standard errors in brackets, obtained with 10 independent runs for each factor model. The standard errors are reasonably small with respect to the absolute values of the log of the marginal likelihood estimates, indicating that the log of the marginal likelihoods are estimated quite accurately. The log of marginal likelihood increases significantly from $k = 1$ to $k = 3$ factor, then the improvement of having additional factor quickly flattens out. It is therefore reasonable to say that four-factor model is the best.

Figure 11: Sequential one-step ahead predictive density estimates for the equally weighted portfolio of US industry stock return from 25/10/2005 to 29/11/2005 estimated using AISIL-PG

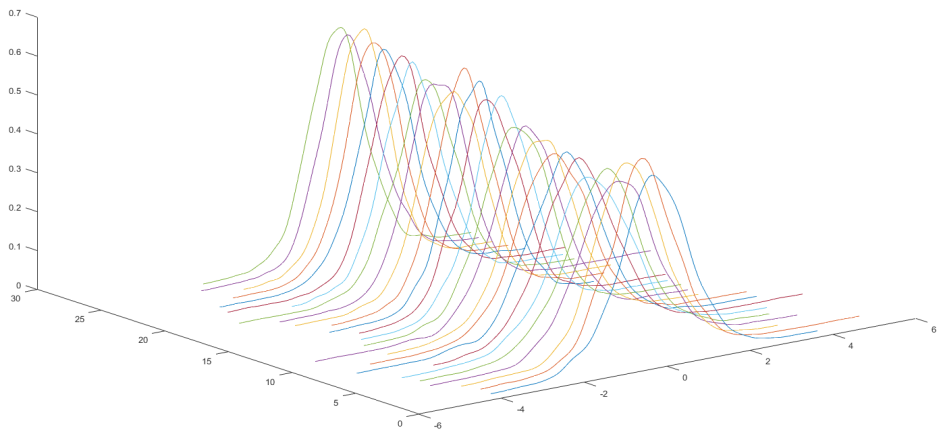


Figure 12: Sequential one-step ahead Value at Risk (VaR) estimates at risk level 0.05 for the equally weighted portfolio of US industry stock return (left) and sequential one-step ahead predictive correlation estimates between banking and automobiles industries and banking and food industries from 25/10/2005 to 29/11/2005 estimated using AISIL-PG

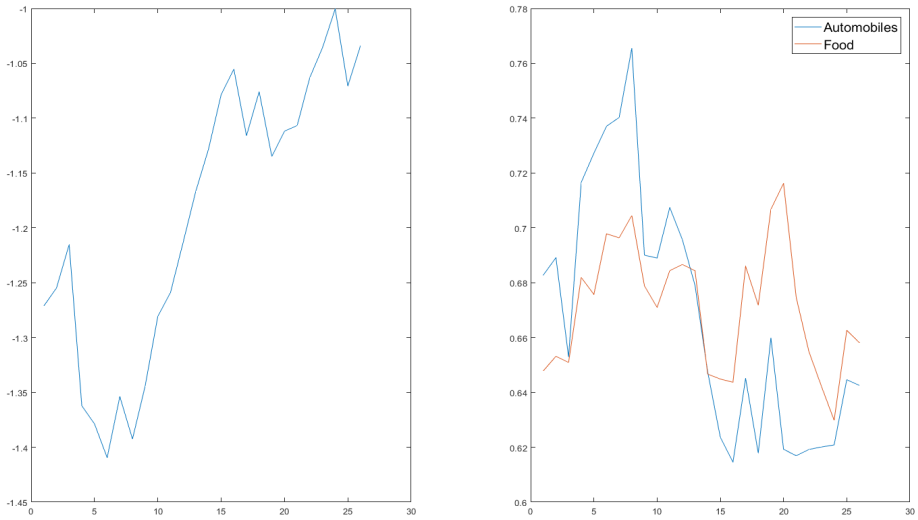


Table 5: Factor SV model estimated using PHS, AISIL-HMC, and AISIL-PG samplers for the US stock returns data with $T = 1000$, $S = 26$, and $K = 1$. Time is in minutes. The results are obtained with 10 independent runs of each algorithm. The table gives: (i) the posterior mean estimates of some of the parameters with posterior standard deviations in brackets. (ii) the estimates of the log of the marginal likelihood $\log \widehat{p}(\boldsymbol{y}_{1:T})$ based on an average of the 10 runs as well as the standard error of the estimate in brackets. (iii) the average value of the number of annealing steps P averaged over the 10 runs.

Param.	Efficient PMMH +PG	HMC-batch	PG-batch	PG-batch-seq
$\phi_{\epsilon 3}$	0.9454 (0.0180)	0.9449 (0.0181)	0.9451 (0.0178)	0.9442 (0.0180)
$\phi_{\epsilon 5}$	0.9483 (0.0178)	0.9484 (0.0178)	0.9468 (0.0183)	0.9477 (0.0182)
$\phi_{\epsilon 19}$	0.9738 (0.0102)	0.9737 (0.0105)	0.9731 (0.0099)	0.9725 (0.0107)
$\phi_{\epsilon 20}$	0.9831 (0.0077)	0.9841 (0.0075)	0.9836 (0.0077)	0.9835 (0.0076)
$\tau^2_{\epsilon 3}$	0.0538 (0.0176)	0.0542 (0.0172)	0.0527 (0.0170)	0.0536 (0.0171)
$\tau^2_{\epsilon 5}$	0.0377 (0.0118)	0.0369 (0.0122)	0.0379 (0.0120)	0.0373 (0.0118)
$\tau^2_{\epsilon 19}$	0.0435 (0.0114)	0.0434 (0.0118)	0.0441 (0.0116)	0.0450 (0.0121)
$\tau^2_{\epsilon 20}$	0.0405 (0.0125)	0.0393 (0.0119)	0.0403 (0.0123)	0.0400 (0.0121)
$\mu_{\epsilon 3}$	−1.7294 (0.1593)	−1.7333 (0.1597)	−1.7355 (0.1545)	−1.7326 (0.1557)
$\mu_{\epsilon 5}$	−0.5392 (0.1436)	−0.5418 (0.1406)	−0.5436 (0.1441)	−0.5329 (0.1386)
$\mu_{\epsilon 19}$	−1.3112 (0.3157)	−1.3042 (0.3246)	−1.3058 (0.2945)	−1.2973 (0.3246)
$\mu_{\epsilon 20}$	−1.0516 (0.5057)	−1.0574 (0.5178)	−1.0697 (0.5071)	−1.0580 (0.5412)
β_{31}	0.9118 (0.0759)	0.9136 (0.0765)	0.9051 (0.0666)	0.9055 (0.0737)
β_{51}	1.0894 (0.0925)	1.0907 (0.0930)	1.0811 (0.0824)	1.0812 (0.0905)
β_{191}	0.5897 (0.0514)	0.5904 (0.0522)	0.5856 (0.0460)	0.5859 (0.0460)
β_{201}	1.0156 (0.0858)	1.0175 (0.0865)	1.0077 (0.0754)	1.0077 (0.0831)
$\log \widehat{p}(\boldsymbol{y}_{1:T})$	-	−26134.94 (58.18)	−26157.25 (22.13)	−26177.94 (40.28)
Total Time	1100	804.17	1572.33	2280.00

Table 6: Selecting number of factors using the log of the estimates of the marginal likelihood (with standard errors in brackets).

K	$\log \widehat{p}(\boldsymbol{y})$
1	−26157.25 (22.13)
2	−25569.64 (41.53)
3	−25289.10 (34.20)
4	−25231.49 (38.74)
5	−25247.06 (47.93)
6	−25228.52 (48.79)

7 Conclusions

The contribution of this article is to propose flexible density tempering or annealing approaches that can be used for both batch and sequential state and parameter estimation problems. Two flexible Markov move steps that are more efficient than the density tempered SMC of Duan and Fulop (2015), the marginalised resample-move (MRM) approach of Fulop and Li (2013) and SMC^2 of Chopin et al. (2013) and that can handle high dimensional parameters and states are proposed. The first is based on particle Gibbs (Andrieu et al., 2010); the second is based on Hamiltonian Monte Carlo (Neal, 2011). The usefullness of the proposed methods is illustrated by applying them to univariate SV models with outliers and structural breaks and a factor SV model having both a large number of parameters and a large number of latent states.

8 Supplementary Material

This article has an online supplement that contains additional technical details and further empirical results.

A Augmented Intermediate Target Density for the Factor SV model

This section provides the augmented tempered target density for the factor SV model; It includes all the random variables produced by the $S + K$ univariate particle filter methods that generate the factor log-volatilities $\boldsymbol{\lambda}_{k,1:T}$, for $k = 1, ..., K$, and the idiosyncratic log-volatilities $\boldsymbol{h}_{s,1:T}$ for $s = 1, ..., S$, as well as the latent factors $\boldsymbol{f}_{1:T}$ and the parameters $\boldsymbol{\theta}$. Eq. (9) is used to specify the particle filters for idiosyncratic

SV log-volatilities $\boldsymbol{h}_{s,1:T}$ for $s = 1, ..., S$, and Eq. (10) to specify the univariate particle filters that generate the factor log-volatilities $\boldsymbol{\lambda}_{k,1:T}$, for $k = 1, ..., K$. The N weighted samples at time t for the factor log-volatilities are denoted by $\left(\boldsymbol{\lambda}_{kt}^{1:N}, \widetilde{\boldsymbol{W}}_{fkt}^{1:N}\right)$ and the idiosyncratic error log-volatilities by $\left(\boldsymbol{h}_{st}^{1:N}, \widetilde{\boldsymbol{W}}_{\epsilon st}^{1:N}\right)$. The corresponding proposal densities are $m_{fk1}\left(\lambda_{k1}|\boldsymbol{\theta}_{fk}, \boldsymbol{y}_1\right)$, $m_{fkt}\left(\lambda_{kt}|\lambda_{kt-1}, \boldsymbol{\theta}_{fk}, \boldsymbol{y}_t\right)$, $m_{\epsilon s1}\left(h_{s1}|\boldsymbol{\theta}_{\epsilon s}, \boldsymbol{y}_1\right)$, and $m_{\epsilon st}\left(h_{st}|h_{st-1}, \boldsymbol{\theta}_{\epsilon s}, \boldsymbol{y}_t\right)$ for the $t = 2, ..., T$. The resampling schemes are denoted by $\mathcal{M}\left(\widetilde{\boldsymbol{a}}_{\epsilon st-1}^{1:N}|\widetilde{\boldsymbol{W}}_{\epsilon st-1}^{1:N}\right)$, where each $\widetilde{a}_{\epsilon st-1}^{j} = k$ indexes a particle in $\left(\boldsymbol{h}_{st-1}^{1:N}, \widetilde{\boldsymbol{W}}_{\epsilon st-1}^{1:N}\right)$ and is chosen with probability $\widetilde{W}_{\epsilon st-1}^{k}$; $\mathcal{M}\left(\widetilde{\boldsymbol{a}}_{fkt-1}^{1:N}|\widetilde{\boldsymbol{W}}_{fkt-1}^{1:N}\right)$ is defined similarly. The vectors of particles are denoted by

$$\boldsymbol{U}_{\epsilon s,1:T} \coloneqq \left(\boldsymbol{h}_{s1}^{1:N}, ..., \boldsymbol{h}_{sT}^{1:N}, \widetilde{\boldsymbol{a}}_{\epsilon s1}^{1:N}, ..., \widetilde{\boldsymbol{a}}_{\epsilon sT-1}^{1:N}\right)$$

and

$$\boldsymbol{U}_{fk,1:T} \coloneqq \left(\boldsymbol{\lambda}_{k1}^{1:N}, ..., \boldsymbol{\lambda}_{kT}^{1:N}, \widetilde{\boldsymbol{a}}_{fk1}^{1:N}, ..., \widetilde{\boldsymbol{a}}_{fkT-1}^{1:N}\right).$$

The joint distribution of the particles given the parameters is

$$\psi_{\epsilon s}\left(\boldsymbol{U}_{\epsilon s,1:T}^{1:N}|\boldsymbol{\theta}_{\epsilon s}\right) \coloneqq \prod_{j=1}^{N} m_{\epsilon s1}\left(h_{s1}^{j}|\boldsymbol{y}_1, \boldsymbol{\theta}_{\epsilon s}\right) \prod_{t=2}^{T}\left\{\mathcal{M}\left(\widetilde{\boldsymbol{a}}_{\epsilon st-1}^{1:N}|\widetilde{\boldsymbol{W}}_{\epsilon st-1}^{1:N}\right) \prod_{j=1}^{N} m_{\epsilon st}\left(h_{st}^{j}|h_{st-1}^{\widetilde{a}_{\epsilon st-1}^{j}}, \boldsymbol{\theta}_{\epsilon s}, \boldsymbol{y}_t\right)\right\},$$

for $s = 1, ..., S$ and

$$\psi_{fk}\left(\boldsymbol{U}_{fk,1:T}^{1:N}|\boldsymbol{\theta}_{fk}\right) \coloneqq \prod_{j=1}^{N} m_{fk1}\left(\lambda_{k1}^{j}|\boldsymbol{y}_1, \boldsymbol{\theta}_{fk}\right) \prod_{t=2}^{T}\left\{\mathcal{M}\left(\widetilde{\boldsymbol{a}}_{fkt-1}^{1:N}|\widetilde{\boldsymbol{W}}_{fkt-1}^{1:N}\right) \prod_{j=1}^{N} m_{fkt}\left(\lambda_{kt}^{j}|\lambda_{kt-1}^{\widetilde{a}_{fkt-1}^{j}}, \boldsymbol{\theta}_{fk}, \boldsymbol{y}_t\right)\right\},$$

for $k = 1, ..., K$. Next, we define indices $\boldsymbol{j}_{\epsilon s,1:T}$ for $s = 1, ..., S$, the selected particle trajectory $\boldsymbol{h}_{s,1:T}^{\boldsymbol{j}_{\epsilon s,1:T}} = \left(h_{s1}^{j_{\epsilon s1}}, ..., h_{sT}^{j_{\epsilon sT}}\right)$, indices $\boldsymbol{j}_{fk,1:T}$ for $k = 1, ..., K$ and the selected particle trajectory $\boldsymbol{\lambda}_{k,1:T}^{\boldsymbol{j}_{fk,1:T}} = \left(\lambda_{k1}^{j_{fk1}}, ..., \lambda_{kT}^{j_{fkT}}\right)$.

The augmented intermediate target density in this case consists of all of the

particle filter variables

$$\tilde{\xi}_{a_p}\left(\boldsymbol{U}_{\epsilon s,1:T}, \boldsymbol{U}_{fk,1:T}, \boldsymbol{\theta}, \boldsymbol{f}_{1:T}\right) \coloneqq \frac{\xi_{a_p}\left(d\boldsymbol{\theta}, d\boldsymbol{h}_{1:T}^{\boldsymbol{j}_{\epsilon s,1:T}}, d\boldsymbol{\lambda}_{1:T}^{\boldsymbol{j}_{fk,1:T}}, d\boldsymbol{f}_{1:T}\right)}{N^{T(S+K)}} \times$$
$$\prod_{s=1}^{S} \frac{\psi_{\epsilon s}\left(\boldsymbol{U}_{\epsilon s,1:T} | \boldsymbol{\theta}_{\epsilon s}\right)}{m_{\epsilon s1}\left(h_{s1}^{j_{\epsilon s1}} | \boldsymbol{\theta}_{\epsilon s}\right) \prod_{t=2}^{T} \widetilde{W}_{\epsilon s t-1}^{\widetilde{a}_{\epsilon s t-1}^{j_{\epsilon s t}}} m_{\epsilon s t}\left(h_{st}^{j_{\epsilon s t}} | h_{st-1}^{\widetilde{a}_{\epsilon s t-1}^{j_{\epsilon s t}}}, \boldsymbol{\theta}_{\epsilon s}\right)} \tag{12}$$
$$\prod_{k=1}^{K} \frac{\psi_{fk}\left(\boldsymbol{U}_{fk,1:T} | \boldsymbol{\theta}_{fk}\right)}{m_{fk1}\left(\lambda_{k1}^{j_{fk1}} | \boldsymbol{\theta}_{fk}\right) \prod_{t=2}^{T} \widetilde{W}_{fkt-1}^{\widetilde{a}_{fkt-1}^{j_{fkt}}} m_{fkt}\left(\lambda_{kt}^{j_{fkt}} | \lambda_{kt-1}^{\widetilde{a}_{fkt-1}^{j_{fkt}}}, \boldsymbol{\theta}_{fk}\right)}.$$

Lemma 2 gives the important properties of the target density. Its proof is in Section S6.

Lemma 2. *The target distribution in Eq. (12) has the marginal distribution*

$$\tilde{\xi}_{a_p}\left(\boldsymbol{h}_{1:T}^{\boldsymbol{j}_{\epsilon s,1:T}}, \boldsymbol{j}_{\epsilon s,1:T}, \boldsymbol{\lambda}_{1:T}^{\boldsymbol{j}_{fk,1:T}}, \boldsymbol{j}_{fk,1:T}, \boldsymbol{\theta}, \boldsymbol{f}_{1:T}\right) = \frac{\xi_{a_p}\left(d\boldsymbol{\theta}, d\boldsymbol{h}_{1:T}^{\boldsymbol{j}_{\epsilon s,1:T}}, d\boldsymbol{\lambda}_{1:T}^{\boldsymbol{j}_{fk,1:T}}, d\boldsymbol{f}_{1:T}\right)}{N^{T(S+K)}}.$$

B Markov moves based on Hamiltonian Monte Carlo for multivariate factor SV model

Algorithm 3 Markov moves based on Hamiltonian Monte Carlo for multivariate factor SV model

for each $i = 1, ..., M$,

1. for $s = 1 : S$
   (a) Sample $\mu_{i,\epsilon s}|\boldsymbol{h}_{is,1:T}, \boldsymbol{\theta}_{i,-\mu_{i,\epsilon s}}$
   (b) Sample $\phi_{i,\epsilon s}|\boldsymbol{h}_{is,1:T}, \boldsymbol{\theta}_{i,-\phi_{i,\epsilon s}}$
   (c) Sample $\tau^2_{i,\epsilon s}|\boldsymbol{h}_{is,1:T}, \boldsymbol{\theta}_{i,-\tau^2_{i,\epsilon s}}$
2. For $k = 1, ..., K$
   (a) Sample $\phi_{i,fk}|\boldsymbol{\lambda}_{ik,1:T}, \boldsymbol{\theta}_{i,-\phi_{i,fk}}$
   (b) Sample $\tau^2_{i,fk}|\boldsymbol{\lambda}_{ik,1:T}, \boldsymbol{\theta}_{i,-\tau^2_{i,fk}}$
3. Sample the factor loading matrix $\boldsymbol{\beta}_i|\boldsymbol{\theta}_{i,-\boldsymbol{\beta}}, \boldsymbol{f}_{i,1:T}, \boldsymbol{h}_{i,1:T}, \boldsymbol{\lambda}_{i,1:T}, \boldsymbol{y}_{1:T}$
4. Sample the latent factors $\boldsymbol{f}_{i,t}|\boldsymbol{\theta}_i, \boldsymbol{h}_{i,1:T}, \boldsymbol{\lambda}_{i,1:T}, \boldsymbol{y}_{1:T}$
5. For $s = 1, ..., S$, sample $\boldsymbol{h}_{is,1:T}|\boldsymbol{\theta}_i, \boldsymbol{f}_{i,1:T}, \boldsymbol{y}_{1:T}$ using Hamiltonian Monte Carlo.
6. For $k = 1, ..., K$, sample $\boldsymbol{\lambda}_{ik,1:T}|\boldsymbol{\theta}_i, \boldsymbol{f}_{i,1:T}, \boldsymbol{y}_{1:T}$ using Hamiltonian Monte Carlo.

C Markov moves based on particle Gibbs for multivariate factor SV model

Algorithm 4 Markov moves based on particle Gibbs for multivariate factor SV model

for $i = 1, ..., M$,

1. For $s = 1, ..., S$,
   (a) Sample $\mu_{i\epsilon s}|\boldsymbol{h}_{is,1:T}^{\boldsymbol{j}_{i\epsilon s,1:T}}, \boldsymbol{j}_{i\epsilon s,1:T}, \boldsymbol{\theta}_{i,-\mu_{i\epsilon s}}, \boldsymbol{f}_{i,1:T}, \boldsymbol{y}_{s,1:T}$
   (b) Sample $\phi_{i\epsilon s}|\boldsymbol{h}_{is,1:T}^{\boldsymbol{j}_{i\epsilon s,1:T}}, \boldsymbol{j}_{i\epsilon s,1:T}, \boldsymbol{\theta}_{i,-\phi_{i\epsilon s}}, \boldsymbol{f}_{i,1:T}, \boldsymbol{y}_{s,1:T}$
   (c) Sample $\tau_{i\epsilon s}^2|\boldsymbol{h}_{is,1:T}^{\boldsymbol{j}_{i\epsilon s,1:T}}, \boldsymbol{j}_{i\epsilon s,1:T}, \boldsymbol{\theta}_{i,-\tau_{i\epsilon s}^2}, \boldsymbol{f}_{i,1:T}, \boldsymbol{y}_{s,1:T}$
2. For $k = 1, ..., K$,
   (a) Sample $\phi_{ifk}|\boldsymbol{\lambda}_{ik,1:T}^{\boldsymbol{j}_{ifk,1:T}}, \boldsymbol{j}_{ifk,1:T}, \boldsymbol{\theta}_{i,-\phi_{ifk}}, \boldsymbol{f}_{ik,1:T}$
   (b) Sample $\tau_{ifk}^2|\boldsymbol{\lambda}_{ik,1:T}^{\boldsymbol{j}_{ifk,1:T}}, \boldsymbol{j}_{ifk,1:T}, \boldsymbol{\theta}_{i,-\tau_{ifk}^2}, \boldsymbol{f}_{ik,1:T}$
3. Sample the factor loading matrix $\boldsymbol{\beta}_i|\boldsymbol{h}_{is,1:T}^{\boldsymbol{j}_{i\epsilon s,1:T}}, \boldsymbol{j}_{i\epsilon s,1:T}, \boldsymbol{\lambda}_{ik,1:T}^{\boldsymbol{j}_{ifk,1:T}}, \boldsymbol{j}_{ifk,1:T}, \boldsymbol{\theta}_{-\boldsymbol{\beta}_i}, \boldsymbol{f}_{i,1:T}, \boldsymbol{y}_{1:T}$.
4. Sample the latent factors $\boldsymbol{f}_{i,t}|\boldsymbol{h}_{is,1:T}^{\boldsymbol{j}_{i\epsilon s,1:T}}, \boldsymbol{j}_{i\epsilon s,1:T}, \boldsymbol{\lambda}_{ik,1:T}^{\boldsymbol{j}_{ifk,1:T}}, \boldsymbol{j}_{ifk,1:T}, \boldsymbol{\theta}_i, \boldsymbol{y}_{1:T}$ for $t = 1, .., T$.
5. For $s = 1, ..., S$, sample
$$\boldsymbol{U}_{i\epsilon s,1:T}^{\left(-\boldsymbol{j}_{i\epsilon s,1:T}\right)} \sim \tilde{\xi}_{a_p}^N \left(\boldsymbol{U}_{i\epsilon s,1:T}^{\left(-\boldsymbol{j}_{i\epsilon s,1:T}\right)} | \boldsymbol{h}_{is,1:T}^{\boldsymbol{j}_{i\epsilon s,1:T}}, \boldsymbol{j}_{i\epsilon s,1:T}, \boldsymbol{\theta}_i, \boldsymbol{f}_{i1:T}, \boldsymbol{y}_{1:T} \right)$$
6. For $s = 1, ..., S$, sample
$J_{i\epsilon s,t} = j_{i\epsilon s,t} \sim \xi_{a_p}\left(j_{i\epsilon s,t}|\boldsymbol{\theta}_i, \boldsymbol{h}_{is,1:t}^{1:N}, \widetilde{\boldsymbol{a}}_{i\epsilon s,1:t-1}^{1:N}, \boldsymbol{h}_{is,t+1:T}^{\boldsymbol{j}_{i\epsilon s,t+1:T}}, \boldsymbol{j}_{i\epsilon s,t+1:T}\right)$
for $t = T-1, ..., 1$ and $J_{i\epsilon s,T} = j_{i\epsilon s,T} \sim \xi_{a_p}\left(j_{i\epsilon s,T}|\boldsymbol{\theta}_i, \boldsymbol{h}_{is,1:T}^{1:N}, \widetilde{\boldsymbol{a}}_{i\epsilon s,1:T-1}^{1:N}\right)$
7. For $k = 1, ..., K$, sample
$$\boldsymbol{U}_{ifk,1:T}^{\left(-\boldsymbol{j}_{ifk,1:T}\right)} \sim \tilde{\xi}_{a_p}^N \left(\boldsymbol{U}_{ifk,1:T}^{\left(-\boldsymbol{j}_{ifk,1:T}\right)} | \boldsymbol{\lambda}_{ik,1:T}^{\boldsymbol{j}_{i,fk,1:T}}, \boldsymbol{j}_{ifk,1:T}, \boldsymbol{\theta}_i, \boldsymbol{f}_{ik1:T} \right)$$
8. For $k = 1, ..., K$, sample
$J_{ifk,t} = j_{ifk,t} \sim \xi_{a_p}\left(j_{ifk,t}|\boldsymbol{\theta}_i, \boldsymbol{\lambda}_{ik,1:t}^{1:N}, \widetilde{\boldsymbol{a}}_{ifk,1:t-1}^{1:N}, \boldsymbol{\lambda}_{ik,t+1:T}^{\boldsymbol{j}_{i\epsilon s,t+1:T}}, \boldsymbol{j}_{ifk,t+1:T}\right)$
for $t = T-1, ..., 1$ and $J_{ifk,T} = j_{ifk,T} \sim \xi_{a_p}\left(j_{ifk,T}|\boldsymbol{\theta}_i, \boldsymbol{\lambda}_{ik,1:T}^{1:N}, \widetilde{\boldsymbol{a}}_{ifk,1:T-1}^{1:N}\right)$.

Steps 1 and 2 are discussed in Section S2.2. Steps 3 and 4 are discussed in Section S3.

Online Supplement: Robust Particle Density Tempering for State Space Models

We use the following notation in the supplement. Eq. (1), Alg. 1, and Sampling Scheme 1, etc, refer to the main paper, while Eq. (S1), Alg. S1, and Sampling Scheme S1, etc, refer to the supplement.

S1 Markov move steps for the univariate SV model

S1.1 Sampling the latent volatilities using Hamiltonian Monte Carlo

This section discusses the Hamiltonian Monte Carlo (HMC) proposal to sample the high dimensional latent state vector $\boldsymbol{x}_{1:T}$. Suppose we want to sample from a T-dimensional distribution with pdf proportional to $\exp(\mathcal{L}(\boldsymbol{x}))$, where $\mathcal{L}(\boldsymbol{x}) = \log p(\boldsymbol{x}|\boldsymbol{\theta}, \boldsymbol{y})$. In Hamiltonian Monte Carlo (Neal, 2011), we augment an auxiliary momentum vector $\boldsymbol{r}$ having the same dimension as the latent state vector $\boldsymbol{x}$ with the density $p(\boldsymbol{r}) = N(\boldsymbol{r}|0, \boldsymbol{M})$, where $\boldsymbol{M}$ is a mass matrix. We define the joint conditional density of $(\boldsymbol{x}, \boldsymbol{r})$ as

$$p(\boldsymbol{x}, \boldsymbol{r}|\text{rest}) \propto \exp(-H(\boldsymbol{x}, \boldsymbol{r})),$$

where $H(\boldsymbol{x}, \boldsymbol{r}) = -\mathcal{L}(\boldsymbol{x}) + \frac{1}{2}\boldsymbol{r}^T M^{-1}\boldsymbol{r}$ is called the Hamiltonian. In an idealised Hamiltonian step, the state vectors $\boldsymbol{x}$ and the momentum variables $\boldsymbol{r}$ move continuously according to the differential equations

$$\frac{d\boldsymbol{x}}{dt} = \frac{\partial H}{\partial \boldsymbol{r}} = \boldsymbol{M}^{-1}\boldsymbol{r}, \frac{d\boldsymbol{r}}{dt} = -\frac{\partial H}{\partial \boldsymbol{x}} = \nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}),$$

where $\nabla_{\boldsymbol{x}}$ denotes the gradient with respect to $\boldsymbol{x}$. In practice, this continuous time Hamiltonian dynamics needs to be approximated by discretizing time, using a step size ϵ. We can then simulate the evolution over time of $(\boldsymbol{x}, \boldsymbol{r})$ via the "leapfrog" integrator using Alg. S1.

Algorithm S1 The Hamiltonian Monte Carlo part of a single Markov move step

Given initial values of $\boldsymbol{x}$, ϵ, L, where L is the number of leapfrog updates
Sample $\boldsymbol{r} \sim N(0, M)$
For $i = 1$ to L
    Set $(\boldsymbol{x}^*, \boldsymbol{r}^*) \leftarrow \text{Leapfrog}(\boldsymbol{x}, \boldsymbol{r}, \epsilon)$ (See Alg. S2)
end for
With probability

$$\alpha = \min\left(1, \frac{\exp\left(\mathcal{L}(\boldsymbol{x}^*) - \frac{1}{2}\boldsymbol{r}^{*T}M^{-1}\boldsymbol{r}^*\right)}{\exp\left(\mathcal{L}(\boldsymbol{x}) - \frac{1}{2}\boldsymbol{r}^{T}M^{-1}\boldsymbol{r}\right)}\right),$$

set $\boldsymbol{x} = \boldsymbol{x}^*$, $\boldsymbol{r}^* = -\boldsymbol{r}$, else retain current x.

Algorithm S2 One step of the leapfrog algorithm

Input i, L, ϵ.

$$\begin{aligned} &\text{if } i = 1, \quad \boldsymbol{r} = \boldsymbol{r} + \epsilon\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})/2, \\ &\qquad\quad \boldsymbol{x} = \boldsymbol{x} + \epsilon\boldsymbol{M}^{-1}\boldsymbol{r}, \\ &\text{if } i = L \quad \boldsymbol{r} = \boldsymbol{r} + \epsilon\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})/2, \quad \text{else} \quad \boldsymbol{r} = \boldsymbol{r} + \epsilon\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x}) \end{aligned}$$

For the univariate SV model Eq. (4), we need the gradient of the log-likelihood $\nabla_{\boldsymbol{x}}\mathcal{L}(\boldsymbol{x})$ with respect to each of the latent volatilities. The required gradient for $t = 1$ is

$$\nabla_{x_1}\mathcal{L}(\boldsymbol{x}) = a_p\left(-0.5 + 0.5y_1^2\exp(-x_1)\right) - \frac{(1-\phi^2)}{\tau^2}(x_1 - \mu) + \frac{\phi}{\tau^2}(x_2 - \mu - \phi(x_1 - \mu));$$

the gradient for $1 < t < T$ is

$$\begin{aligned} \nabla_{x_t}\mathcal{L}(\boldsymbol{x}) = a_p\left(-0.5 + 0.5y_t^2\exp(-x_t)\right) + \frac{\phi}{\tau^2}(x_{t+1} - \mu - \phi(x_t - \mu)) - \\ \frac{1}{\tau^2}(x_t - \mu - \phi(x_{t-1} - \mu)); \end{aligned}$$

and, for $t = T$, the gradient is

$$\nabla_{x_T}\mathcal{L}(\boldsymbol{x}) = a_p\left(-0.5 + 0.5y_T^2\exp(-x_T)\right) - \frac{1}{\tau^2}(x_T - \mu - \phi(x_{T-1} - \mu)).$$

S1.2 Sampling Univariate SV parameters

For $i = 1, ..., M$, we sample $\mu_i|\boldsymbol{x}_{i1:T}, \boldsymbol{y}_{1:T}, \boldsymbol{\theta}_{-\mu}$ from $N\left(\mu_\mu, \sigma_\mu^2\right)$ truncated within $(-10, 10)$, where

$$\sigma_\mu^2 = \frac{\tau_i^2}{1 - \phi_i^2 + (T-1)(1-\phi_i)^2}$$

and

$$\mu_\mu = \sigma_\mu^2 \frac{x_{i1}\left(1-\phi_i^2\right) + \sum_{t=2}^{T} x_{it} - \phi_i x_{it} + \phi_i^2 x_{it-1} - \phi_i x_{it-1}}{\tau_i^2}.$$

We sample the persistence parameter ϕ_i by drawing a proposed value ϕ_i^* from $N\left(\mu_\phi, \sigma_\phi^2\right)$ truncated within $(-1,1)$, where

$$\sigma_\phi^2 = \frac{\tau_i^2}{\sum_{t=2}^{T}\left(x_{it-1}-\mu_i\right)^2 - \left(x_{i1}-\mu_i\right)^2}$$

and

$$\mu_\phi = \sigma_\phi^2 \frac{\sum_{t=2}^{T}\left(x_{it}-\mu_i\right)\left(x_{it-1}-\mu_i\right)}{\tau_i^2},$$

and accept with probability

$$\min\left(1, \frac{p\left(\phi_i^*\right)\sqrt{1-\phi_i^{2*}}}{p\left(\phi_i\right)\sqrt{1-\phi_i^2}}\right).$$

We sample τ_i^2 from $\mathrm{IG}\left(v_1/2, s_1/2\right)$, where $v_1 = v_0 + T$ and $s_1 = s_0 + \left(1-\phi_i^2\right)\left(x_{i1}-\mu_i\right)^2 + \sum_{t=2}^{T}\left(x_{it}-\mu_i-\phi_i\left(x_{it-1}-\mu_i\right)\right)^2$.

S2 Markov moves for Factor SV model

S2.1 Sampling the latent volatilities using Hamiltonian Monte Carlo

For $s = 1, ..., S$, the gradient of the log-likelihood $\nabla_{\boldsymbol{h}_s}\mathcal{L}\left(\boldsymbol{h}_s\right)$ with respect to each of the latent volatilities $\boldsymbol{h}_{s,1:T}$ is required. The required gradient is, for $t = 1$,

$$\nabla_{\boldsymbol{h}_{s1}}\mathcal{L}\left(\boldsymbol{h}_s\right) = a_p\left(-0.5 + 0.5\left(y_{s1} - \beta_s \boldsymbol{f}_1\right)^2 \exp\left(-h_{s1}\right)\right) - \frac{\left(1-\phi_{\epsilon s}^2\right)}{\tau_{\epsilon s}^2}\left(h_{s1} - \mu_{\epsilon s}\right) + \frac{\phi_{\epsilon s}}{\tau_{\epsilon s}^2}\left(h_{s2} - \mu_{\epsilon s} - \phi_{\epsilon s}\left(h_{s1} - \mu_{\epsilon s}\right)\right),$$

for $1 < t < T$, the required gradient is

$$\nabla_{h_{st}}\mathcal{L}\left(\boldsymbol{h}_s\right) = a_p\left(-0.5 + 0.5\left(y_{st} - \beta_s \boldsymbol{f}_t\right)^2 \exp\left(-h_{st}\right)\right) + \frac{\phi_{\epsilon s}}{\tau_{\epsilon s}^2}\left(h_{st+1} - \mu_{\epsilon s} - \phi_{\epsilon s}\left(h_{st} - \mu_{\epsilon s}\right)\right) - \frac{1}{\tau_{\epsilon s}^2}\left(h_{st} - \mu_{\epsilon s} - \phi_{\epsilon s}\left(h_{st-1} - \mu_{\epsilon s}\right)\right).$$

and, for $t = T$, the required gradient is

$$\nabla_{h_{sT}} \mathcal{L}(\boldsymbol{h}_s) = a_p\left(-0.5 + 0.5\left(y_{sT} - \beta_s \boldsymbol{f}_T\right)^2 \exp\left(-h_{sT}\right)\right) - \frac{1}{\tau_{\epsilon s}^2}\left(h_{sT} - \mu_{\epsilon s} - \phi_{\epsilon s}\left(h_{sT-1} - \mu_{\epsilon s}\right)\right).$$

For $k = 1, ..., K$, the gradient of the log-likelihood $\nabla_{\boldsymbol{\lambda}_{k,1:T}} \mathcal{L}(\boldsymbol{\lambda}_k)$ with respect to each of the latent volatilities $\boldsymbol{\lambda}_{k,1:T}$ is required. The required gradient is, for $t = 1$,

$$\nabla_{\boldsymbol{\lambda}_{k1}} \mathcal{L}(\boldsymbol{\lambda}_k) = a_p\left(-0.5 + 0.5 f_{k1}^2 \exp\left(-\lambda_{k1}\right)\right) - \frac{\left(1 - \phi_{fk}^2\right)}{\tau_{fk}^2}\lambda_{k1} + \frac{\phi_{fk}}{\tau_{fk}^2}\left(\lambda_{k2} - \phi_{fk}\lambda_{k1}\right),$$

for $1 < t < T$, the required gradient is

$$\nabla_{\lambda_{kt}} \mathcal{L}(\boldsymbol{\lambda}_k) = a_p\left(-0.5 + 0.5 f_{kt}^2 \exp\left(-\lambda_{kt}\right)\right) + \frac{\phi_{fk}}{\tau_{fk}^2}\left(\lambda_{kt+1} - \phi_{fk}\lambda_{kt}\right) - \frac{1}{\tau_{fk}^2}\left(\lambda_{kt} - \phi_{fk}\lambda_{kt-1}\right).$$

and, for $t = T$, the required gradient is

$$\nabla_{\lambda_{kT}} \mathcal{L}(\boldsymbol{\lambda}_k) = a_p\left(-0.5 + 0.5 f_{kT}^2 \exp\left(-\lambda_{kT}\right)\right) - \frac{1}{\tau_{fk}^2}\left(\lambda_{kT} - \phi_{fk}\lambda_{kT-1}\right).$$

S2.2 Sampling the idiosyncratic and factor log-volatilities parameters

For $s = 1, ..., S$, sample $\mu_{\epsilon s}$ from $N\left(\mu_{\mu,\epsilon s}, \sigma_{\mu,\epsilon s}^2\right)$ truncated within $(-10, 10)$, where

$$\sigma_{\mu,\epsilon s}^2 = \frac{\tau_{\epsilon s}^2}{1 - \phi_{\epsilon s}^2 + (T-1)\left(1 - \phi_{\epsilon s}\right)^2}$$

and

$$\mu_{\mu,\epsilon s} = \sigma_{\mu,\epsilon s}^2 \frac{h_{s1}\left(1 - \phi_{\epsilon s}^2\right) + \sum_{t=2}^{T} h_{st} - \phi_{\epsilon s} h_{st} + \phi_{\epsilon s}^2 h_{st-1} - \phi_{\epsilon s} h_{st-1}}{\tau_{\epsilon s}^2}.$$

For $s = 1, ..., S$, we sample $\phi_{\epsilon s}$ by drawing a proposed value $\phi_{\epsilon s}^*$ from $N\left(\mu_{\phi,\epsilon s}, \sigma_{\phi,\epsilon s}^2\right)$ truncated within $(-1, 1)$, where

$$\sigma_{\phi,\epsilon s}^2 = \frac{\tau_{\epsilon s}^2}{\sum_{t=2}^{T}\left(h_{st-1} - \mu_{\epsilon s}\right)^2 - \left(h_{s1} - \mu_{\epsilon s}\right)^2}$$

and

$$\mu_{\phi,\epsilon s} = \sigma_{\phi,\epsilon s}^2 \frac{\sum_{t=2}^{T}\left(h_{st} - \mu_{\epsilon s}\right)\left(h_{st-1} - \mu_{\epsilon s}\right)}{\tau_{\epsilon s}^2}.$$

The candidate is accepted with probability

$$\min\left(1, \frac{p\left(\phi_{\epsilon s}^{*}\right)\sqrt{1-\phi_{\epsilon s}^{2*}}}{p\left(\phi_{\epsilon s}\right)\sqrt{1-\phi_{\epsilon s}^{2}}}\right).$$

For $s = 1, ..., S$, we sample $\tau_{\epsilon s}^2$ from $\text{IG}\left(v_{1,\epsilon s}/2, s_{1,\epsilon s}/2\right)$, where $v_{1,\epsilon s} = v_{0,\epsilon s} + T$ and $s_{1,\epsilon s} = s_{0,\epsilon s} + \left(1-\phi_{\epsilon s}^2\right)\left(h_{s1} - \mu_{\epsilon s}\right)^2 + \sum_{t=2}^{T}\left(h_{st} - \mu_{\epsilon s} - \phi_{\epsilon s}\left(h_{st-1} - \mu_{\epsilon s}\right)\right)^2$.

For $k = 1, ..., K$, we sample ϕ_{fk} by drawing a proposed value ϕ_{fk}^{*} from $N\left(\mu_{\phi,fk}, \sigma_{\phi,fk}^2\right)$ truncated within $(-1, 1)$, where

$$\sigma_{\phi,fk}^2 = \frac{\tau_{fk}^2}{\sum_{t=2}^{T-1}\lambda_{kt}^2}, \quad \text{and} \quad \mu_{\phi,fk} = \frac{\sum_{t=2}^{T}\lambda_{kt}\lambda_{kt-1}}{\sum_{t=2}^{T-1}\lambda_{kt}^2}.$$

For $k = 1, ..., K$, we sample τ_{fk}^2 from $\text{IG}\left(v_{1,fk}/2, s_{1,fk}/2\right)$, where $v_{1,fk} = v_{0,fk} + T$ and $s_{1,fk} = s_{0,fk} + \left(1-\phi_{fk}^2\right)\lambda_{k1}^2 + \sum_{t=2}^{T}\left(\lambda_{kt} - \phi_{\epsilon s}\lambda_{kt-1}\right)^2$.

S3 Sampling the factor loading matrix β and the latent factors $\boldsymbol{f}_{1:T}$

In this section, we discuss the parameterization of the factor loading matrix and the latent factors, and how they are sampled from their full conditional distribution.

To identify the parameters of the model for the factor loading matrix β, it is necessary to impose some further constraints. Usually, the factor loading matrix β is assumed lower triangular in the sense that $\beta_{sk} = 0$ for $k > s$ and furthermore, one of two constraints are used. i) The first is that the f_{kt} have unit variance (Geweke and Zhou, 1996); ii) or, alternatively, assume that $\beta_{ss} = 1$, for $s = 1, ..., S$, and the variance of $\boldsymbol{f}_t$ is diagonal but unconstrained. The main drawback of the lower triangular assumption on β is that the resulting inference can depend on the order in which the components of $\boldsymbol{y}_t$ are chosen (Chan et al., 2017). However, a unique identification of the loading matrix is not necessary if the focus is to estimate and predict the covariance structure. This allows leaving the factor loading matrix completely unrestricted. In our empirical application, we leave the factor loading matrix completely unrestricted such that the results invariance with respect to the ordering of the series.

Let z_s denote the number of unrestricted elements in row s and define

$$F_s = \begin{bmatrix} f_{11} & \cdots & f_{z_s 1} \\ \vdots & & \vdots \\ f_{1T} & \cdots & f_{z_s T} \end{bmatrix},$$

and

$$V_s = \begin{bmatrix} \exp(h_{s1}) & \cdots & 0 \\ 0 & \ddots & \vdots \\ 0 & \cdots & \exp(h_{sT}) \end{bmatrix},$$

then sampling the factor loadings $\boldsymbol{\beta}_{s,\cdot}^{'} = (B_{s1}, ..., B_{sz_s})$ for $s = 1, ..., S$, conditionally on $\boldsymbol{f}_{1:T}$ can be done independently for each s, by performing a Gibbs-update from

$$\boldsymbol{\beta}_{s,\cdot}^{'} | \boldsymbol{f}_{1:T}, \boldsymbol{y}_{s,1:T} \sim N_{z_s}(a_{sT}, b_{sT}), \tag{S1}$$

where

$$b_{sT} = \left(a_p \left(F_s^{'} V_s^{-1} F_s \right) + B_0^{-1} I_{z_s} \right)^{-1},$$

and

$$a_{sT} = b_{sT} F_s^{'} \left(a_p V_s^{-1} \boldsymbol{y}_{s,1:T} \right).$$

Sampling of $\{\boldsymbol{f}_t\} | \boldsymbol{y}_t, \{\boldsymbol{h}_t\}, \{\boldsymbol{\lambda}_t\}, \boldsymbol{\beta}$. After completing some algebra, we can show that $\{\boldsymbol{f}_t\}$ can be sampled from

$$\{\boldsymbol{f}_t\} | \boldsymbol{y}_t, \{\boldsymbol{h}_t\}, \{\boldsymbol{\lambda}_t\}, \beta \sim N(a_t, b_t), \tag{S2}$$

where

$$b_t = \left(a_p \left(\beta^{'} V_t^{-1} \beta \right) + D_t^{-1} \right)^{-1},$$

and

$$a_t = b_t \beta^{'} \left(a_p V_t^{-1} \boldsymbol{y}_t \right).$$

S4 Particle Filter, Conditional Particle Filter, and backward simulation Algorithm

Algorithm S3 Generic Particle Filter Algorithm

Inputs: $\boldsymbol{y}_{1:T}$, N, $\boldsymbol{\theta}$
Outputs: $\boldsymbol{x}_{1:T}^{1:N}$, $\boldsymbol{a}_{1:T-1}^{1:N}$,$\widetilde{\boldsymbol{w}}_{1:T}^{1:N}$

1. For $t = 1$

   (a) Sample $\boldsymbol{x}_1^j$ from $m_1\left(\boldsymbol{x}_1|\boldsymbol{y}_1, \boldsymbol{\theta}\right)$, for $j = 1, ..., N$

   (b) Calculate the importance weights

   $$\widetilde{w}_1^j = \frac{p\left(\boldsymbol{y}_1|\boldsymbol{x}_1^j, \boldsymbol{\theta}\right)^{a_p} p\left(\boldsymbol{x}_1^j|\boldsymbol{\theta}\right)}{m_1\left(\boldsymbol{x}_1^j|\boldsymbol{y}_1, \boldsymbol{\theta}\right)}, j = 1, ..., N.$$

   and normalise them to obtain $\widetilde{\boldsymbol{W}}_1^{1:N}$.

2. For $t > 1$

   (a) Sample the ancestral indices $\widetilde{\boldsymbol{a}}_{t-1}^{1:N} \sim \mathcal{M}\left(\widetilde{\boldsymbol{a}}_{t-1}^{1:N}|\widetilde{\boldsymbol{W}}_{t-1}^{1:N}\right)$.

   (b) Sample $\boldsymbol{x}_t^j$ from $m_t\left(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^j}, \boldsymbol{\theta}\right)$, $j = 1, ..., N$.

   (c) Calculate the importance weights

   $$\widetilde{w}_t^j = \frac{p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^j, \boldsymbol{\theta}\right)^{a_p} p\left(\boldsymbol{x}_t^j|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^j}, \boldsymbol{\theta}\right)}{m_t\left(\boldsymbol{x}_t^j|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^j}, \boldsymbol{\theta}\right)}, j = 1, ..., N.$$

   and normalised to obtain $\widetilde{\boldsymbol{W}}_t^{1:N}$.

Algorithm S4 Conditional Sequential Monte Carlo algorithm

Inputs: N, $\boldsymbol{\theta}$, $\boldsymbol{y}_{1:T}$, $\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}$, and $\boldsymbol{j}_{1:T}$
Outputs: $\boldsymbol{x}_{1:T}^{1:N}$, $\boldsymbol{a}_{1:T-1}^{1:N}$,$\widetilde{\boldsymbol{w}}_{1:T}^{1:N}$

1. For $t = 1$

   (a) Sample $\boldsymbol{x}_1^j$ from $m_1\left(\boldsymbol{x}_1|\boldsymbol{y}_1,\boldsymbol{\theta}\right)$, for $j \in \{1,...,N\} \setminus \{j_1\}$.

   (b) Calculate the weights

$$\widetilde{w}_1^j = \frac{p\left(\boldsymbol{y}_1|\boldsymbol{x}_1^j,\boldsymbol{\theta}\right)^{a_p} p\left(\boldsymbol{x}_1^j|\boldsymbol{\theta}\right)}{m_1\left(\boldsymbol{x}_1^j|\boldsymbol{y}_1,\boldsymbol{\theta}\right)}, j = 1,...,N.$$

   and normalised to obtain $\widetilde{\boldsymbol{W}}_1^{1:N}$.

2. For $t > 1$

   (a) Sample the ancestral indices $\widetilde{\boldsymbol{a}}_{t-1}^{(-j_t)} \sim \mathcal{M}\left(\widetilde{\boldsymbol{a}}_{t-1}^{(-j_t)}|\widetilde{\boldsymbol{W}}_{t-1}^{1:N}\right)$.

   (b) Sample $\boldsymbol{x}_t^j$ from $m_t\left(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^j},\boldsymbol{\theta}\right)$, $j = 1,...,N \setminus \{j_t\}$.

   (c) Calculate the weights

$$\widetilde{w}_t^j = \frac{p\left(\boldsymbol{y}_t|\boldsymbol{x}_t^j,\boldsymbol{\theta}\right)^{a_p} p\left(\boldsymbol{x}_t^j|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^j},\boldsymbol{\theta}\right)}{m_t\left(\boldsymbol{x}_t^j|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^j},\boldsymbol{\theta}\right)}, j = 1,...,N.$$

   and normalised to obtain $\widetilde{\boldsymbol{W}}_t^{1:N}$.

Algorithm S5 The Backward simulation algorithm

1. Sample $J_T = j_T$ conditional on $(\boldsymbol{U}_{1:T},\theta)$, with probability proportional to $w_T^{j_T}$, and choose $x_T^{j_T}$;

2. For $t = T-1,...,1$, sample $J_t = j_t$ conditional on $\left(\boldsymbol{u}_{1:t},\boldsymbol{j}_{t+1:T},x_{t+1}^{j_{t+1}},...,x_T^{j_T}\right)$, and with probability proportional to $w_t^{j_t} f_\theta\left(x_{t+1}^{j_{t+1}}|x_t^{j_t}\right)$, and choose $x_t^{j_t}$.

S5 Assumptions

In Section S4, we use the particle filter to approximate the joint filtering densities $\{p\left(\boldsymbol{x}_t|\boldsymbol{y}_{1:t}\right) : t = 1,...,T\}$ sequentially using N particles, $\left\{\boldsymbol{x}_t^{1:N},\widetilde{W}_t^{1:N}\right\}$, drawn from some proposal densities $m_1\left(\boldsymbol{x}_1|\boldsymbol{y}_1,\boldsymbol{\theta}\right)$ and $m_t\left(\boldsymbol{x}_t|\boldsymbol{x}_{t-1},\boldsymbol{y}_{1:t},\boldsymbol{\theta}\right)$ for $t \geq 2$. For $t \geq 1$,

we follow (Andrieu et al., 2010) and define,

$$S_t^{\boldsymbol{\theta}} \coloneqq \left(\boldsymbol{x}_{1:t} \in \boldsymbol{\chi}^t : \pi\left(\boldsymbol{x}_{1:t}|\boldsymbol{\theta}\right) > 0\right) \quad \text{and} \quad Q_t^{\boldsymbol{\theta}} \coloneqq \left\{\boldsymbol{x}_{1:t} \in \boldsymbol{\chi}^t : \pi\left(\boldsymbol{x}_{1:t-1}|\boldsymbol{\theta}\right) m_t\left(\boldsymbol{x}_t|\boldsymbol{\theta}, \boldsymbol{x}_{1:t-1}, \boldsymbol{y}_{1:t}\right) > 0\right\}.$$

Assumption S1. *(Andrieu et al., 2010) We assume that* $S_t^{\boldsymbol{\theta}} \subseteq Q_t^{\boldsymbol{\theta}}$ *for any* $\boldsymbol{\theta} \in \boldsymbol{\Theta}$ *and* $t = 1, ..., T$

Assumption S1 is always satisfied in our implementation because we use the bootstrap filter with $p\left(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{\theta}\right)$ as a proposal density, and $p\left(\boldsymbol{y}_t|\boldsymbol{x}_t, \boldsymbol{\theta}\right) > 0$ for all $\boldsymbol{\theta}$.

Assumption S2. *For any* $j = 1, ..., N$ *and* $t = 1, .., T$*, the resampling scheme* $\mathcal{M}\left(\widetilde{\boldsymbol{a}}_{t-1}^{1:N}|\widetilde{\boldsymbol{W}}_{t-1}^{1:N}\right)$ *satisfies* $\Pr\left(\widetilde{A}_{t-1}^k = j|\widetilde{\boldsymbol{W}}_{t-1}^{1:N}\right) = \widetilde{W}_{t-1}^j$*. (Chopin and Singh, 2015; Andrieu et al., 2010).*

Assumption 2 is satisfied by the popular resampling schemes, such as multinomial, systematic, residual resampling.

S6 Proofs

Proof of Lemma 1. We prove the lemma by carrying out the marginalisation. The marginal distribution $\xi_{a_p}\left(\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, \boldsymbol{\theta}\right)$ is obtained by integrating $\widetilde{\xi}_{a_p}\left(\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, \boldsymbol{U}_{1:T}^{-\boldsymbol{j}_{1:T}}, \boldsymbol{\theta}\right)$ over $\left(\boldsymbol{x}_{1:T}^{(-\boldsymbol{j}_{1:T})}, \widetilde{\boldsymbol{a}}_{1:T-1}\right)$. We begin by integrating over $\left(\boldsymbol{x}_T^{-j_T}, \widetilde{\boldsymbol{a}}_{T-1}\right)$ to obtain

$$\widetilde{\xi}_{a_p}\left(d\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, d\boldsymbol{x}_{1:T-1}^{(-\boldsymbol{j}_{1:T-1})}, d\widetilde{\boldsymbol{a}}_{1:T-2}, d\boldsymbol{\theta}\right) = \frac{\xi_{a_p}\left(d\boldsymbol{\theta}, d\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}\right)}{N^T} \frac{\psi\left(d\boldsymbol{x}_{1:T-1}^{1:N}, d\widetilde{\boldsymbol{a}}_{1:T-2}|\boldsymbol{\theta}\right)}{m_1\left(d\boldsymbol{x}_1^{j_1}|\boldsymbol{\theta}, \boldsymbol{y}_1\right) \prod_{t=2}^{T-1} \widetilde{W}_{t-1}^{\widetilde{a}_{t-1}^{jt}} m_t\left(d\boldsymbol{x}_t^{jt}|\boldsymbol{x}_{t-1}^{\widetilde{a}_{t-1}^{jt}}, \boldsymbol{\theta}\right)}. \tag{S3}$$

We repeat this for $t = T-2, ..., 2$, to obtain,

$$\widetilde{\xi}_{a_p}\left(d\boldsymbol{x}_1^{(-j_1)}, d\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, d\boldsymbol{\theta}\right) = \frac{\xi_{a_p}\left(d\boldsymbol{\theta}, d\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}\right)}{N^T} \times \frac{\psi\left(d\boldsymbol{x}_1^{1:N}\right)}{m_1\left(d\boldsymbol{x}_1^{j_1}|\boldsymbol{\theta}, \boldsymbol{y}_1\right)}.$$

Finally, we integrate over $\boldsymbol{x}_1^{j_1}$, to obtain

$$\widetilde{\xi}_{a_p}\left(d\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}, \boldsymbol{j}_{1:T}, d\boldsymbol{\theta}\right) = \frac{\xi_{a_p}\left(d\boldsymbol{\theta}, d\boldsymbol{x}_{1:T}^{\boldsymbol{j}_{1:T}}\right)}{N^T}.$$

□

Proof of Lemma 2. Similar to the proof of Lemma 1, we can show that the marginal distribution

$$\tilde{\xi}_{a_p}\left(d\boldsymbol{h}_{1:T}^{\boldsymbol{j}_{\epsilon s,1:T}}, \boldsymbol{j}_{\epsilon s,1:T}, d\boldsymbol{\lambda}_{1:T}^{\boldsymbol{j}_{fk,1:T}}, \boldsymbol{j}_{fk,1:T}, d\boldsymbol{\theta}, d\boldsymbol{f}_{1:T}\right) = \frac{\xi_{a_p}\left(d\boldsymbol{\theta}, d\boldsymbol{h}_{1:T}^{\boldsymbol{j}_{\epsilon s,1:T}}, d\boldsymbol{\lambda}_{1:T}^{\boldsymbol{j}_{fk,1:T}}, d\boldsymbol{f}_{1:T}\right)}{N^{T(S+K)}},$$

is obtained by integrating $\tilde{\xi}_{a_p}\left(\boldsymbol{U}_{\epsilon s,1:T}, \boldsymbol{U}_{fk,1:T}, \boldsymbol{\theta}, \boldsymbol{f}_{1:T}\right)$ over $\left(\boldsymbol{h}_{s1:T}^{(-\boldsymbol{j}_{\epsilon s1:T})}, \widetilde{\boldsymbol{a}}_{\epsilon s1:T-1}\right)$ for $s = 1, ..., S$, and $\left(\boldsymbol{\lambda}_{k1:T}^{(-\boldsymbol{j}_{fk1:T})}, \widetilde{\boldsymbol{a}}_{fk1:T-1}\right)$ for $k = 1, ..., K$. □

S7 List of Industry Portfolios

Table S1: The list of industry portfolios

	Stocks
1	Coal
2	Health Care and Equipment
3	Retail
4	Tobacco
5	Steel Works
6	Food Products
7	Recreation
8	Printing and Publishing
9	Consumer Goods
10	Apparel
11	Chemicals
12	Textiles
13	Fabricated Products
14	Electrical Equipment
15	Automobiles and Trucks
16	Aircraft, ships, and Railroad Equipment
17	Industrial Mining
18	Petroleum and Natural Gas
19	Utilities
20	Telecommunication
21	Personal and Business Services
22	Business Equipment
23	Transportation
24	Wholesale
25	Restaurants, Hotels, and Motels
26	Banking, Insurance, Real Estate

S8 Additional Empirical Results for Factor SV model

Figure S1: Plot of log-volatilities $\boldsymbol{h}_{1:T}$ for US stock returns data estimated using the PMCMC, AISIL-HMC, and AISIL-PG methods

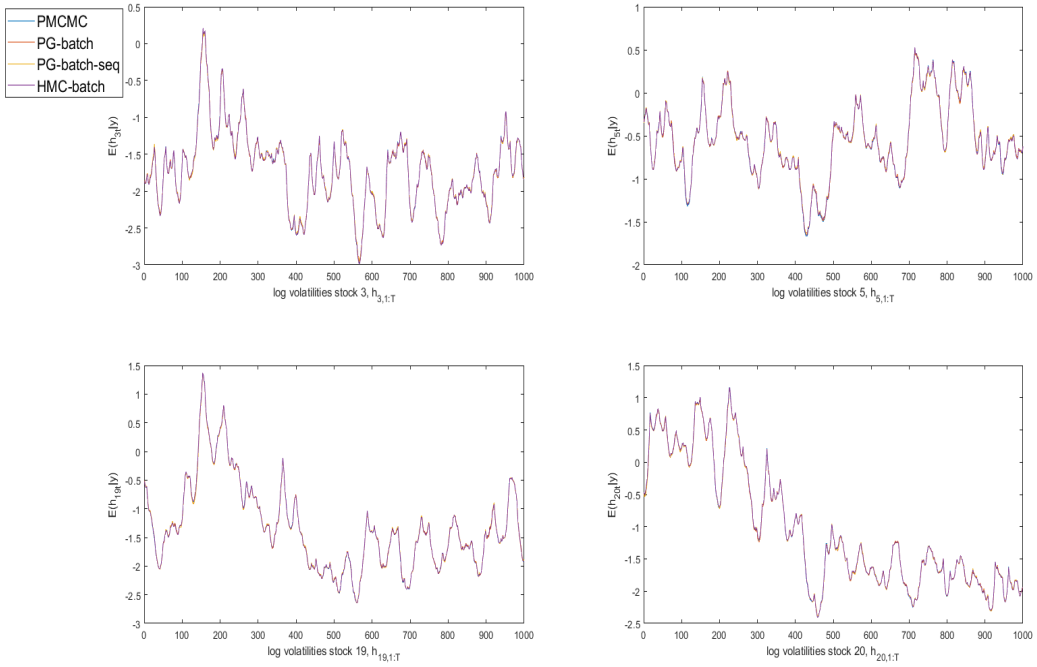


Figure S2: The kernel density estimates of the marginal posterior densities of $\phi_{\epsilon s}$ for the US stock returns data for four representative $\phi_{\epsilon s}$. The density estimates are for the Efficient PMMH +PG, AISIL-HMC, and AISIL-PG methods.

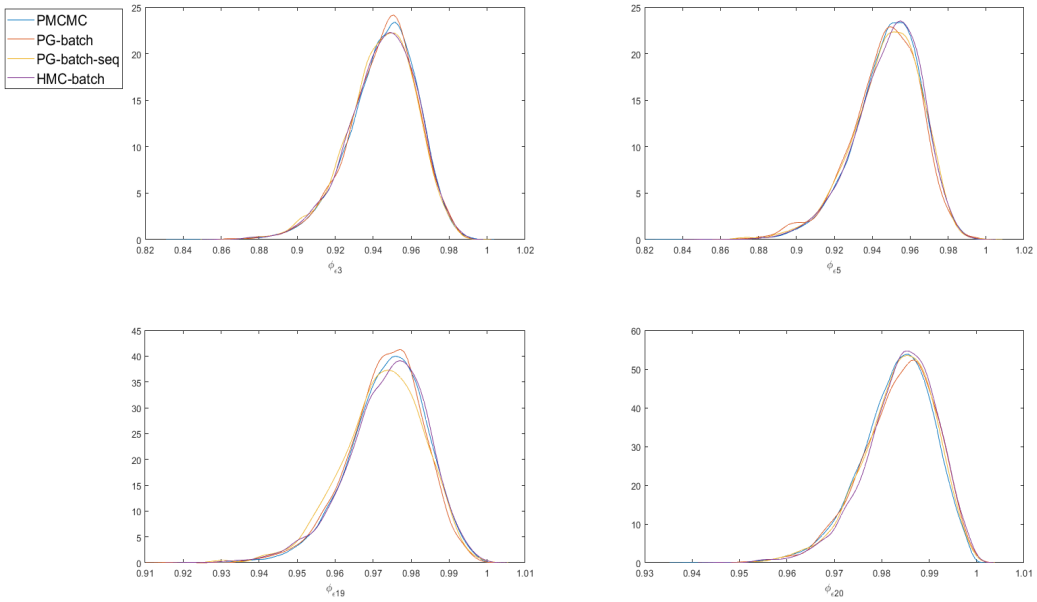


Figure S3: The kernel density estimates of the marginal posterior densities of $\tau^2_{\epsilon s}$ for the US stock returns data for four representative $\tau^2_{\epsilon s}$. The density estimates are for the Efficient PMMH +PG, AISIL-HMC, and AISIL-PG methods.

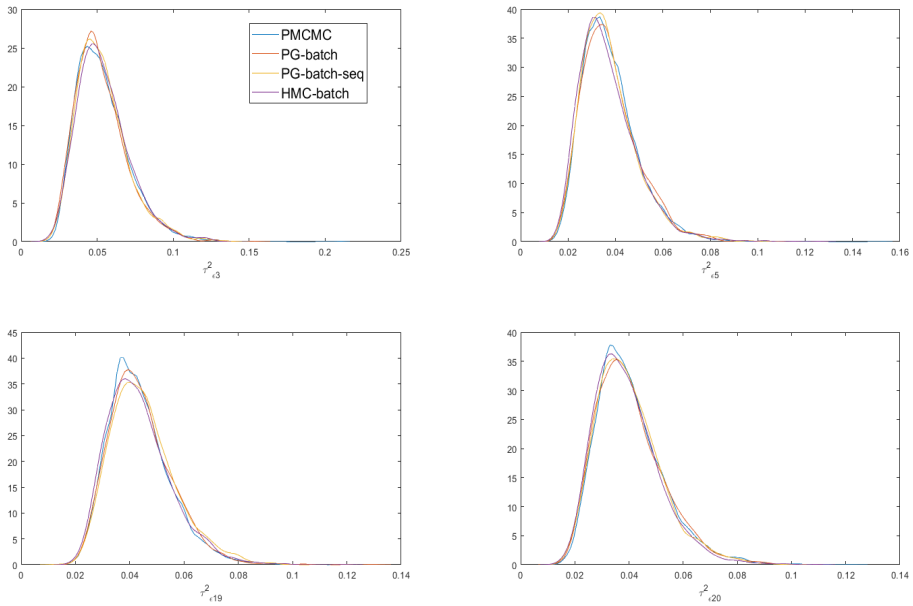


Figure S4: The kernel density estimates of the marginal posterior densities of β for the US stock returns data for four representative β. The density estimates are for the PMCMC, AISIL-HMC, and AISIL-PG methods.

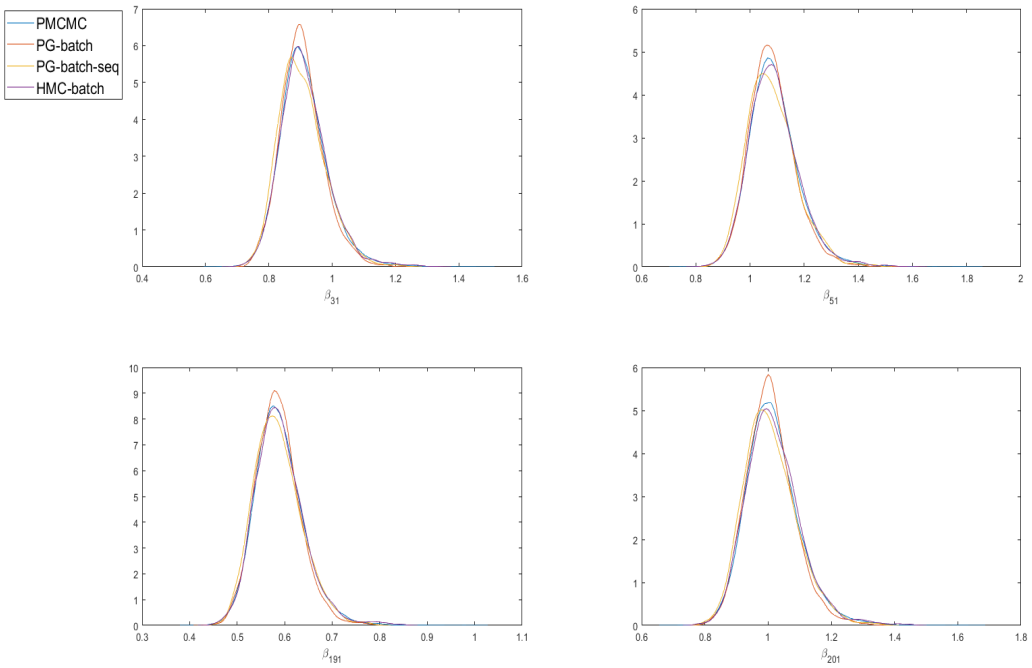
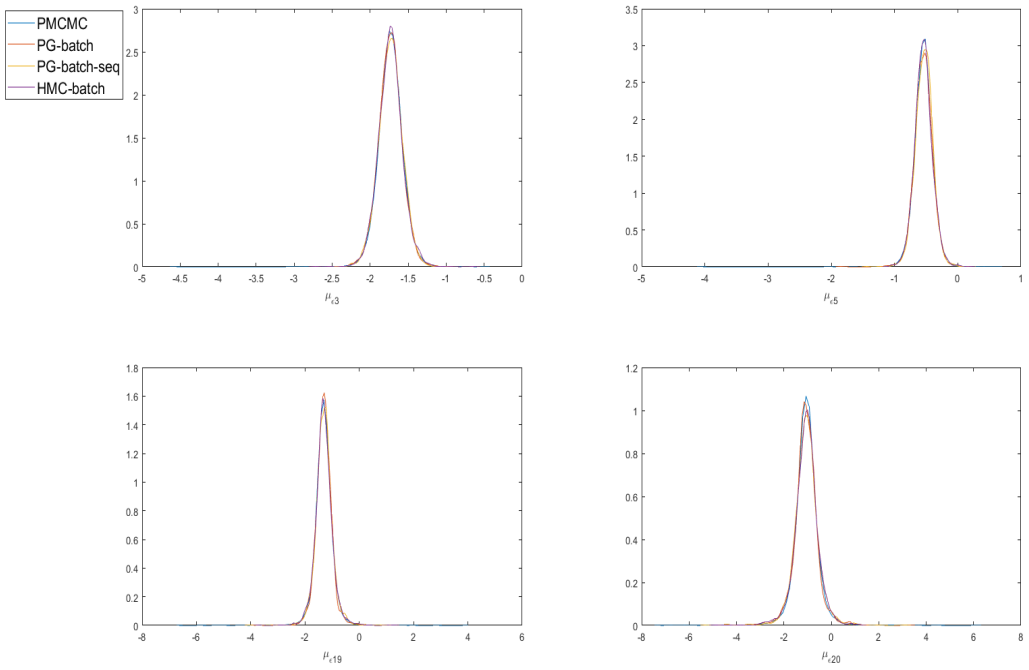


Figure S5: The Kernel Density Estimates of marginal posterior densities of μ for the US stock returns data for four representative μ. The density estimates are for the PMCMC, AISIL-HMC, and AISIL-PG methods.



References

Aguilar, O. and West, M. (2000). Bayesian dynamic factor models and portfolio allocation. *Journal of Business and Economics Statistics*, 18(3):338–357.

Andrieu, C., Doucet, A., and Holenstein, R. (2010). Particle Markov chain Monte Carlo methods. *Journal of the Royal Statistical Society, Series B*, 72:1–33.

Beskos, A., Jasra, A., Kantas, N., and Thiery, A. H. (2016). On the convergence of adaptive sequential Monte Carlo methods. *The Annals of Applied Probability*, 26(2):1111–1146.

Betancourt, M. (2017). A conceptual introduction to Hamiltonian Monte Carlo. *arXiv preprint arXiv:1701.02434*.

Buchholz, A., Chopin, N., and Jacob, P. E. (2020). Adaptive tuning of Hamiltonian Monte Carlo within sequential Monte Carlo. *Bayesian Analysis*, pages 1–27.

Chan, J., Gonzalez, R. L., and Strachan, R. W. (2017). Invariant inference and efficient computation in the static factor model. *Journal of American Statistical Association*.

Chib, S. (1998). Estimation and comparison of multiple change-point models. *Journal of Econometrics*, 86(2):221–241.

Chib, S. and Jeliazkov, I. (2001). Marginal likelihood from the Metropolis-Hastings output. *Journal of American Statistical Association*, 96(453):270–281.

Chib, S., Nardari, F., and Shephard, N. (2006). Analysis of high dimensional multivariate stochastic volatility models. *Journal of Econometrics*, 134:341–371.

Chopin, N. (2004). Central limit theorem for sequential Monte carlo methods and its application to Bayesian inference. *The Annals of Statistics*, 32:2385–2411.

Chopin, N., Jacob, P. E., and Papaspiliopoulos, O. (2013). SMC2: an efficient algorithm for sequential analysis of state space models. *Journal of Royal Statistical Society Series B*, 75(3):397–426.

Chopin, N. and Singh, S. S. (2015). On particle Gibbs sampling. *Bernoulli*, 21(3):1855–1883.

Del Moral, P. (2004). *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*. Springer, New York.

Del Moral, P., Doucet, A., and Jasra, A. (2006). Sequential Monte Carlo samplers. *Journal of the Royal Statistical Society, Series B*, 68:411–436.

Del Moral, P., Doucet, A., and Jasra, A. (2012). An adaptive Sequential Monte Carlo for approximate Bayesian computation. *Statistics and Computing*, pages 1009–1020.

Deligiannidis, G., Doucet, A., and M.Pitt (2018). the correlated pseudomarginal method. *Journal of Royal Statistical Society Series B*, 80(5):839–870.

Duan, J. C. and Fulop, A. (2015). Density-tempered marginalised sequential Monte Carlo samplers. *Journal of Business and Economics Statistics*, 33(2):192–202.

Duane, S., Kennedy, A. D., Pendleton, B. J., and Roweth, D. (1987). Hybrid Monte Carlo. *Physics Letters B*, 195(2):216–222.

Fulop, A. and Li, J. (2013). Efficient learning via simulation: A marginalised resample-move approach. *Journal of Econometrics*, 176:146–161.

Garthwaite, P. H., Fan, Y., and Sisson, S. A. (2015). Adaptive optimal scaling of Metropolis-Hastings algorithms using the Robbins-Monro process. *Communications in Statistics - Theory and Methods*, 45(17):5098–5111.

Gerlach, R., Carter, C., and Kohn, R. (1998). Diagnostics for time series analysis. *Journal of Time Series Analysis*, 20(3).

Geweke, J. F. and Zhou, G. (1996). Measuring the pricing error of the arbitrage pricing theory. *Review of Financial Studies*, 9:557–587.

Girolami, M. and Calderhead, B. (2011). Riemannian manifold Langevin and Hamiltonian Monte Carlo methods. *Journal of Royal Statistician Society Series B*, 73(2):123–214.

Godsill, S., Doucet, A., and West, M. (2004). Monte Carlo smoothing for nonlinear time series. *Journal of American Statistical Association*, 99(465):156–168.

Gunawan, D., Carter, C., and Kohn, R. (2020). On scalable particle Markov chain Monte Carlo. *arXiv:1804.04359v3*.

Hoffman, M. D. and Gelman, A. (2014). The No-U-Turn sampler: adaptively setting path length in Hamiltonian Monte Carlo. *Journal of Machine Learning Research*, 15:1593–1623.

Holton, G. A. (2003). *Value-at-risk: theory and practice*. Academic Press.

Jensen, M. J. and Maheu, J. M. (2010). Bayesian semiparametric stochastic volatility modeling. *Journal of Econometrics*, 157(2):306–316.

Kass, R. E. and Raftery, A. E. (1995). Bayes factors. *Journal of American Statistical Association*, 90(430):773–795.

Kastner, G., Schnatter, S. F., and Lopes, H. F. (2017). Efficient bayesian inference for multivariate factor stochastic volatility models. *Journal of Computational and Graphical Statistics*, 26(4):905–917.

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., Jordan, M. I., and Schon, T. B. (2014). Particle Gibbs with ancestor sampling. *Journal of Machine Learning Research*, 15:2145–2184.

Lindsten, F. and Schon, T. B. (2013). Backward simulation methods for Monte Carlo statistical inference. *Foundation and Trends in machine learning*, 6(1):1–43.

Mendes, E. F., Gunawan, D., Carter, C. K., and Kohn, R. (2020). A flexible particle Markov chain Monte Carlo method. *Statistics and Computing*, 30:783–798.

Nardari, F. and Scruggs, J. T. (2007). Bayesian analysis of linear factor models with latent factors, multivariate stochastic volatility, and apt pricing restrictions. *Journal of Financial and Quantitative Analysis*, 42(4):857–891.

Neal, R. (2001). Annealed importance sampling. *Statistics and Computing*, 11:125–139.

Neal, R. M. (2011). MCMC using Hamiltonian dynamics. *Handbook of Markov chain Monte Carlo, Chapman and Hall/CRC*.

Olsson, J. and Ryden, T. (2011). Rao-Blackwellisation of particle Markov chain Monte Carlo methods using forward filtering backward sampling. *IEEE Transactions on Signal Processing*, 59(10):4606–4619.

Pitt, M. K., Silva, R. S., Giordani, P., and Kohn, R. (2012). On some properties of Markov chain Monte Carlo simulation methods based on the particle filter. *Journal of Econometrics*, 171(2):134–151.

Shephard, N. E. (1994). Partial non-Gaussian state space models. *Biometrika*, 81:115–131.

Sherlock, C., Thiery, A., Roberts, G., and Rosenthal, J. (2015). On the efficiency of pseudo-marginal random walk Metropolis algorithms. *Annals of Statistics*, 43(1):238–275.

Smith, J. Q. (1985). Diagnostic checks of non-standard time series models. *Journal of Forecasting*, 4:283–291.

Stephens, M. A. (1974). EDF statistics for goodness of fit and some comparisons. *Journal of American Statistical Association*, 69:730–737.

Zhou, X., Nakajima, J., and West, M. (2014). Bayesian forecasting and portfolio decisions using dynamic dependent sparse factor models. *International Journal of Forecasting*, 30(4):963–980.