

Article

“Exact” and Approximate Methods for Bayesian Inference: Stochastic Volatility Case Study

Yuliya Shapovalova

Institute for Computing and Information Sciences, Radboud University Nijmegen, Toernooiveld 212, 6525 EC Nijmegen, The Netherlands; yuliya.shapovalova@ru.nl

Abstract: We conduct a case study in which we empirically illustrate the performance of different classes of Bayesian inference methods to estimate stochastic volatility models. In particular, we consider how different particle filtering methods affect the variance of the estimated likelihood. We review and compare particle Markov Chain Monte Carlo (MCMC), RMHMC, fixed-form variational Bayes, and integrated nested Laplace approximation to estimate the posterior distribution of the parameters. Additionally, we conduct the review from the point of view of whether these methods are (1) easily adaptable to different model specifications; (2) adaptable to higher dimensions of the model in a straightforward way; (3) feasible in the multivariate case. We show that when using the stochastic volatility model for methods comparison, various data-generating processes have to be considered to make a fair assessment of the methods. Finally, we present a challenging specification of the multivariate stochastic volatility model, which is rarely used to illustrate the methods but constitutes an important practical application.

Keywords: Bayesian inference; Markov Chain Monte Carlo; Sequential Monte Carlo; Riemann Manifold Hamiltonian Monte Carlo; integrated nested laplace approximation; fixed-form variational Bayes; stochastic volatility



Citation: Shapovalova, Y. ‘Exact’ and Approximate Methods for Bayesian Inference: Stochastic Volatility Case Study. *Entropy* **2021**, *23*, 466. <https://doi.org/10.3390/e23040466>

Academic Editor: Pierre Alquier

Received: 28 February 2021
Accepted: 12 April 2021
Published: 15 April 2021

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

The field of Bayesian statistics and machine learning has advanced in recent years quite rapidly. The methods that have been developed do not often find fast assimilation across different fields. In this review, we aim to provide the reader with methodologies that try to solve the estimation problem in models with latent variables and intractable likelihoods. We are in particular interested in the methods that can be used to estimate nonlinear state-space models and in particular stochastic (latent) volatility models. There are multiple studies that conducted review and comparison of the methods of estimation of the stochastic volatility models [1–3]. We briefly mention some of the methods that have been reviewed; however, most of the methods considered in this paper have not entered those reviews. In this paper, we focus in particular on comparing methods that target posterior distribution exactly and the methods that try to approximate it. We also conduct the review from the point of view of estimating multivariate models with these methods and discuss what the bottleneck is in each of them when extending to higher-dimensional stochastic volatility (SV) models. We consider different data-generating processes for simulating data in the empirical studies and conclude that the choice of the data-generating process can heavily affect performance of a method. Thus, illustrating the performance of a method on just one data generating process or one real-world data set is not sufficient.

In financial econometrics literature, GARCH-type models prevail since they are much simpler to estimate. Stochastic (latent) volatility models, however, can be more natural frameworks for modeling asset returns. They can provide flexible and intuitive tools for applications in financial econometrics as well as some other disciplines. In particular, multivariate stochastic volatility models offer an attractive framework for detection and measuring *volatility spillover effects*. Volatility spillovers in this framework can be defined

through Granger-causal links in the latent (unobservable) volatility process, which is modeled with a Vector Autoregressive model (VAR(p)). Insights about the causal structure can help to identify the relationship (Granger-causality or/and contemporaneous correlation) between the financial markets. Such information can be insightful and helpful in the decision-making process of portfolio managers and policymakers. These models are, however, rarely considered in practice. Multiple Bayesian inference methods have been proposed for the estimation of this class of models in recent years. In this paper, we identify the bottlenecks in different classes of methods for the estimation of these models in the multivariate case.

One of the stepping stones of estimation of the nonlinear state-space models in general (and stochastic volatility models in particular) lies in the intractability of the likelihood, which is the result of the presence of an unobservable process in the model and nonlinear dependence between this process and the observed data. The likelihood can be estimated with particle filter methods, also known as Sequential Monte Carlo. This is a computationally intensive procedure; however, depending on the problem and the data, it can provide excellent results. The second stepping stone of the estimation is the intractable posterior distribution. A standard starting point for sampling from the posterior distribution is the Metropolis-Hastings algorithm, which is a general method and can be applied straightforwardly to different models. It works well when the number of parameters in the model is small. However, the convergence of the algorithm can be slow in larger models, due to inefficiency of the sampling with random walk proposals. Particle Metropolis-Hastings [4] combines Sequential Monte Carlo for the likelihood estimation with Metropolis-Hastings for the sampling from the posterior, which results in a state-of-the-art method in terms of the estimation quality since it targets the exact posterior. The downside of this method is that it is computationally extremely demanding. Note that, while we consider particle Metropolis-Hastings in this paper, the class of methods from [4] is more general.

Two main downsides of particle Metropolis-Hastings are random walk behavior of the proposals and computational burden. One of the possible solutions to the first problem are the algorithms that use gradient information for the construction of the proposal distribution and thus explore the parameter space more efficiently. An additional step in improving these algorithms is defining them on a Riemann manifold instead of Euclidean space as proposed in [5]. The resulting algorithm, which we consider for the comparison in this paper, is Riemann Manifold Hamiltonian Monte Carlo. For extensive comparison of the methods that exploit gradient information and Langevin dynamics—such as Metropolis-adjusted Langevin algorithm, Hamiltonian Monte Carlo, Riemann manifold Metropolis adjusted Langevin algorithm, and Riemann Manifold Hamiltonian Monte Carlo—we refer to [5].

Thus far, we have discussed the methods that target the posterior distribution exactly and have a high computational burden, which makes empirical investigation of their performance in high-dimensional cases infeasible. In the last decade, a large number of methods have been published on approximate posterior inference that allow much faster computations, but lose in terms of precision of the estimation. In this paper, we consider two such methods that deal with different types of approximation. Fixed-form variational Bayes, proposed in [6], assumes hierarchical factorization of the prior and posterior distributions, and the factorized distributions are approximated by an analytically tractable distribution from a certain family of distributions $q(\cdot)$. Then, instead of solving integration problem, one solves the optimization problem of minimizing the Kullback–Leibler divergence between $q(\cdot)$ and $p(\cdot)$, where $p(\cdot)$ is the target distribution. The second approximate method that we consider is the integrated nested Laplace approximation (INLA) [7]. The method relies on the nested version of the classical Laplace approximation. It became very popular in recent years and made computations in many models feasible.

In this review paper, we focus our attention on the following methodologies and provide a comparison for some of the methods via a simulation study. We consider how the variance of the estimated likelihood is affected by choosing different particle-filtering

algorithms. Unlike previous studies, we consider the variance of the estimated likelihood over the whole parameter space and notice that it is affected by some parameters of the model more than by the others. We compare particle Metropolis-Hastings with Riemann Manifold Hamiltonian Monte Carlo as two state-of-the-art sampling methods for this type of problem. We assess how well the INLA method performs in the task of the estimation of the parameters of stochastic volatility model and finally, compare fixed-form variational Bayes methods with sampling by RMHMC. All the between-methods comparisons are performed on multiple simulated data sets with different underlying parameters. We illustrate that, for fair comparison and performance assessment, illustration only on data sets is not sufficient.

The paper is organized as follows. In Section 2, we introduce the model and its different specifications. While in simulation studies we use univariate model, we do introduce multivariate stochastic volatility models with Granger-causal feedback as the model of interest for high-dimensional inference. In Section 3, we review the methods that can be used for the estimation of this class of models. We introduce major ideas behind these methods, and for the details of the derivations we refer to the original papers. In Section 4, we perform empirical case study on different simulated data sets and compare the methods on two real-world time series. We in particular focus on the precision loss of parameter estimation when using approximate methods and how adaptable the methods are to perform multivariate estimation and estimation of various model specifications.

2. Model

2.1. Univariate Stochastic Volatility Model

In this section, we introduce the model of interest that we will use in the simulation studies. Stochastic volatility (SV) models are concerned with modeling asset prices or asset returns depending on how the model is formulated. Let P_t be the price of the asset at time t or the exchange rate at time t (we consider two applications to real data in Section 3.5: one to exchange rate and one to log-returns). Then the log-return y_t is

$$y_t = \log(1 + R_t) = \log \frac{P_t}{P_{t-1}}. \quad (1)$$

Stochastic volatility models are built in such a way that they can mimic *stylized facts* about financial markets and log-returns y_t . Stylized facts are empirically observed statistical properties of asset prices and asset returns. Typical examples of stylized facts are

- *Volatility clustering and persistence*: the big changes in asset returns tend to be followed by big changes, and small changes in asset returns tend to be followed by small changes; in other words, there are periods of large fluctuations and small fluctuations [8].
- *Leverage effect*: the changes in stock prices may be negatively related to the changes in volatility [9].
- *Co-movements*: different stocks tend to exhibit co-movements, which means that if the volatility of one stock changes in a specific direction, volatilities of the other stocks tend to change in the same direction [9].

One of the earlier works that received much attention in the financial literature and proposed a mathematical model that tried to explain the dynamics of financial markets is [10]. Numerous continuous-time stochastic volatility models have been proposed since then, and among the first ones, multiple variants should be mentioned [11–14]. The model we will be considering in this chapter can be viewed as a discrete version of the model in [13] derived by using Euler–Maruyama approximation. The stochastic volatility model in continuous time can be written as

$$ds(t) = \sigma(t)dB_1(t), \quad (2)$$

$$\ln \sigma^2(t) = \mu + \beta \ln \sigma^2(t)dt + \sigma_\eta dB_2(t), \quad (3)$$

where $s(t)$ is log of asset price, $\sigma^2(t)$ is the volatility, $B_1(t)$ and $B_2(t)$ are Brownian motions that satisfy $\text{corr}(B_1(t), B_2(t)) = \rho$. If $\rho < 0$, there is leverage effect present. Thus, log of asset price follows diffusion and its volatility parameter also follows diffusion [15]. As we often get the data in discrete time, usually the discrete time approximation of the model is used in practice. The discrete model then follows by using Euler–Maruyama approximation

$$y_t = \sigma_t \epsilon_t, \quad (4)$$

$$\ln \sigma_{t+1}^2 = \mu + \phi \ln \sigma_t^2 + \sigma_\eta \eta_{t+1}, \quad (5)$$

where y_t is logarithmic return, $\epsilon_t = B_1(t+1) - B_1(t)$, $\eta_{t+1} = B_2(t+1) - B_2(t)$, $\phi = 1 + \beta$. Further, $\epsilon_t \sim N(0, 1)$ and $\eta_t \sim N(0, 1)$, $\text{corr}(\epsilon_t, \eta_{t+1}) = \rho$.

We get state-space representation of the model that is commonly used by defining $h_t = \ln \sigma_t^2$ and $\sigma_t^2 = \exp(h_t)$

$$y_t = \exp(h_t/2) \epsilon_t, \quad (6)$$

$$h_{t+1} = \mu + \phi h_t + \eta_{t+1}, \quad (7)$$

where y_t are log-returns that are observed and volatility h_t is latent and drives the dynamics of y_t . Figure 1 illustrates this structure of the model. Note that the latent volatility process has an autoregressive form. However, unlike in the standard autoregressive model, the latent volatility is not observed and thus has to be estimated together with the model parameters μ , ϕ , and σ_η , which are the scale, the volatility persistence and the noise variance of the latent volatility process, respectively. The persistence parameter ϕ reflects one of the stylized facts of financial returns, namely volatility persistence. The intuition is as follows: if $\phi > 0$ and $\exp(h_{t-1}/2)$ is large, then $\exp(h_t/2)$ will tend to be large too. Hence, the model can account for volatility clustering. In this paper, we consider stationary volatility cases with $|\phi| < 1$. Finally, one can also incorporate leverage effects by defining negative correlation between noise terms ϵ_t and η_{t+1} . Intuitive interpretation of the leverage effect goes as follows: bad news tends to decrease the price, which means that financial leverage increases, the firm becomes riskier, and thus expected volatility also increases. The leverage effects in this model have been studied in [16]. The stochastic volatility model can be parametrized in multiple ways; often, the following alternatives are considered [2]. Other ways to parametrize this model are presented in Equations (8) and (9). The left-hand side version of the model corresponds to that of [17]. The right-hand side version is a different way to define the scaling parameter; in this case, it is β . For identifiability reasons, only β or μ as in Equation (7) should be included in the model.

$$\begin{aligned} y_t &= \sqrt{h_t} \epsilon_t & y_t &= \beta \exp(h_t/2) \epsilon_t & (8) \\ \log h_t &= \mu + \phi \log h_{t-1} + \eta_t & h_t &= \phi h_{t-1} + \eta_t & (9) \end{aligned}$$

Note that the authors of [17] define the leverage effect as correlation between ϵ_t and η_t , so the correlation between noise terms is contemporaneous while [16] model correlation between ϵ_t and η_{t+1} , which corresponds to correlation of the returns with one-step-ahead volatility. Reference Yu [18] shows that the approach of [16] is preferable. In particular, while in case of [16] the model is a martingale difference sequence, i.e., the past does not help to predict the future of the time series, in the case of [17], it is not. Hence, in the latter case, the efficient market hypothesis is violated.

In the remainder of this manuscript, we will work with either specification of the model defined in Equations (6) and (7) or in right-hand side of Equations (8) and (9). These models are equivalent, and we interchange the representation either for the convenience of using some of the methods or for comparison with other work. In the literature, both specifications are frequently used, and in some papers (for example, ref. [19]) the transition from one specification to another is conducted by observing that $\beta = \exp(\mu/2)$.

Under the assumption that $|\phi| < 1$, the unconditional first and second moments of the latent process h_t are

$$\mathbb{E}(h_t) = \frac{\mu}{1 - \phi}, \quad \text{Var}(h_t) = \frac{\sigma_\eta^2}{1 - \phi^2}. \tag{10}$$

The challenge of the estimation of the model lies in the intractability of the likelihood and posterior distribution. The likelihood factorizes as

$$L(y|\theta) = \prod_{t=1}^T p(y_t|y_{1:t-1}, \theta), \tag{11}$$

where the terms in the product can be computed recursively, and it becomes clear that the likelihood is a high-dimensional integral

$$p(y_t|y_{1:t-1}, \theta) = \int p(y_t|h_t, \theta)p(h_t|y_{1:t-1}, \theta)dh_t. \tag{12}$$

There is no analytical solution to the integral in Equation (12), and in this paper, we consider methods to estimate it using sequential Monte Carlo methods.

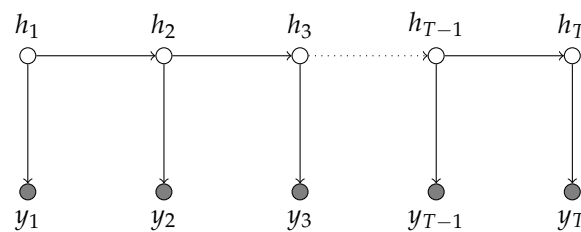


Figure 1. Graphical representation of stochastic volatility model. Observations y_t represented by shaded edges depend at each time point on the state of the latent volatility process h_t .

2.2. Multivariate Stochastic Volatility Model

In this section, we introduce the multivariate stochastic volatility model, which is rarely used in practice due to the challenges of estimation. One of the objectives of this paper is to assess whether modern methods in Bayesian inference are capable of the estimation of these models in high-dimensional case. Multivariate or high dimensional application of this class of models can give insightful information to practitioners. We deal with the same set-up as before; however, we now consider multiple time series of logarithmic returns that are interconnected through the latent volatility process

$$y_t = \Omega_t \epsilon_t, \tag{13}$$

where $\epsilon_t \sim N(\mathbf{0}, \mathbf{R})$ and \mathbf{R} is a correlation matrix with entries $r_{ii} = 1, i = 1, \dots, n$ on the diagonal. Furthermore, Ω_t is a diagonal matrix that contains time-varying volatilities that are driven by an independent stochastic process h_t ,

$$\Omega_t = \text{diag}(\exp(h_t/2)).$$

The process h_t of log-volatilities follows a VAR(p) process

$$h_t = \mu + \sum_{k=1}^p \Phi_k (h_{t-k} - \mu) + \eta_t, \tag{14}$$

where $\Phi_k = (\phi_{ij,k})_{i,j=1,\dots,n}$ are $n \times n$ coefficient matrices. Introducing the matrices $\Phi_k = (\phi_{ij,k})_{i,j=1,\dots,n}$ allows us to model connectivity in financial time series through the concept of Granger-causality in latent volatility process. We say that h_i does not Granger-cause h_j

if all $(\phi_{ij,k})_{k=1,\dots,p} = 0$. The standard conditions on stationarity of a vector autoregressive model apply: the root of $|\mathbb{I} - \lambda\Phi| = 0$ should lie outside the unit circle, and the errors η_t are independent and identically normally distributed with mean zero and variance-covariance matrix $\Sigma = \text{diag}(\sigma_1^2, \dots, \sigma_n^2)$. Equations (13) and (14) are multivariate extensions of the model described in Equations (6) and (7). The representation from the right-hand side of Equations (8) and (9) can be obtained by including a vector of parameters β into Ω_t and removing μ from Equation (14). As before, for identifiability, only one vector of the scale parameters—either μ or β —should be included in the model.

The above MSV model can also be viewed as a non-linear state-space model where (14) is the state equation of the latent process h_t and (13) is the observation equation that depends non-linearly on the latent state. Note that, in this model, the time series are interconnected and the relationship between them can be interpreted through the concept of Granger-causality in latent volatility processes.

3. Methods

3.1. Bayesian Inference

In this paper, we review various methods that sample from or approximate the posterior distribution of the parameters of the model θ . The sampling or approximate methods are necessary since we are working in the framework when the posterior distribution and the likelihood are analytically intractable. The Bayes' rule allows us to write posterior distribution in the form

$$p(\theta|\mathbf{y}) = \frac{\pi(\theta)g(\mathbf{y}|\theta)}{m(\mathbf{y})}, \quad (15)$$

where $\pi(\theta)$ is the prior distribution of the parameters of the model, $g(\mathbf{y}|\theta)$ is the likelihood of the data given parameters of the model, and $m(\mathbf{y})$ is the marginal density of \mathbf{y} , which can be viewed as normalizing constant and which we will ignore in this paper. In the remainder of the paper we will work with the Bayes' rule in proportionality terms:

$$p(\theta|\mathbf{y}) \propto \pi(\theta)g(\mathbf{y}|\theta). \quad (16)$$

Note that in the stochastic volatility model we have to estimate parameters of the model $\theta = (\mu, \phi, \sigma^2)$ and the latent vector of volatilities \mathbf{h} . Thus, we are interested in the following form of the Bayes' rule

$$p(\theta, \mathbf{h}|\mathbf{y}) \propto g(\mathbf{y}|\theta, \mathbf{h})f(\mathbf{h}|\theta)\pi(\theta). \quad (17)$$

Multiple approaches can be used for the estimation of $p(\theta, \mathbf{h}|\mathbf{y})$. One of the challenges is that neither posterior $p(\theta, \mathbf{h}|\mathbf{y})$ nor the likelihood $g(\mathbf{y}|\theta, \mathbf{h})$ is tractable. We start our review by considering sequential Monte Carlo methods, also known as particle filtering, for the estimation of the likelihood $g(\mathbf{y}|\theta, \mathbf{h})$. We then discuss Metropolis-Hastings algorithm for sampling from the posterior and how these two algorithm can be combined into particle Metropolis-Hastings for sampling from the posterior distribution. We continue the review of the methods by considering RMHMC method in which the parameters and volatilities are sampled within the same framework. Finally, we review two approximate methods: integrated nested Laplace approximation and fixed-form variational Bayes, two different ways of approximating posterior distribution.

3.2. Sequential Monte Carlo for the Estimation of the Likelihood

The Sequential Monte Carlo (SMC) method, also known in the literature as *particle filtering*, is considered a state-of-the-art method for estimation of the intractable likelihoods in nonlinear state-space models. The general idea behind this method lies in the estimation of the latent states by drawing multiple samples (particles) and then propagating them in time according to corresponding importance weights. By combining the weights over all time steps, one obtains a marginal likelihood estimate. Standard and well-known schemes are *Bootstrap particle filter* (BPF) [20], *Sequential Importance Sampling* (SIS), and *Sequential Importance Resampling* (SIR) [21]. Sequential Monte Carlo methods were elegantly

combined with Markov Chain Monte Carlo in [4], and the method was named particle Markov Chain Monte Carlo (PMCMC). This method provides a powerful and coherent approach for Bayesian inference in a wide range of complex models. In the later subsections, we will discuss how sequential Monte Carlo methods are combined with Markov Chain Monte Carlo for fully Bayesian inference in stochastic volatility models. One of the concerns when using and implementing SMC for the likelihood estimation is the variance of the estimated likelihood. Standard SMC techniques such as SIS are prone to have high variance of the estimated likelihood once the dimensionality of the problem increases [22]. A number of studies have tried to address this problem. The common choice of proposal for sample of particles in standard schemes is $f(h_t|h_{t-1})$. Pitt and Shephard [23] propose an *auxiliary particle filter* as a solution that is using proposal for particles which takes into account the current observation $q(h_t|h_{t-1}, y_t)$ and not only the dynamics of the latent process itself. Scharth and Kohn [24] suggest using efficient importance sampling [25] inside the PMCMC procedure. Guarniero et al. [26] use twisted representation of the model and use the look-ahead type of particle filtering to address the issue of high variance of the estimated likelihood. Johansen and Doucet [27] compare sequential importance resampling (SIR) with auxiliary particle filter and find that APF does not always outperform SIR. Often, the variance of the estimated likelihood is analyzed in the true value of the parameters, such as in [24]. However, when using particle Markov Chain Monte Carlo, it is also of interest whether the same conclusions hold in different points of the parameter space. In particular, we never start running the algorithm at the point of the true parameter values. This means that if the variance of the estimated likelihood is much larger in some areas of the parameter space, the convergence of the algorithm can be affected. Having insights into how the variance of the estimated likelihood is different in the parameter space can help to make a more efficient choice of the starting point for the algorithm.

We first review the sequential Monte Carlo methods for the estimation of the likelihood. After that, we discuss Metropolis-Hastings algorithm and how SMC and Metropolis-Hastings can be combined for Bayesian inference in general and stochastic volatility models in particular.

3.2.1. Sequential Monte Carlo

Assume that we are in the framework with an observed time series process y_t and a latent Markovian process h_t . Since we never observe the latent process, we need to infer it. The objective that can be achieved with *Sequential Monte Carlo* (SMC) is also known as *particle filtering*. The method operates in sequential manner with arriving observations y_t . The posterior distribution of the latent process can be computed sequentially

$$p(h_{0:t}|y_{1:t}) = p(h_{0:t-1}|y_{0:t-1}) \frac{g(y_{1:t}|h_{0:t})f(h_t|h_{t-1})}{p(y_t|y_{t-1})}. \quad (18)$$

The denominator of Equation (18) is not analytically tractable, which can be also seen from Equation (12) earlier. SMC allows us to estimate the posterior distribution $p(h_{0:t}|y_{1:t})$ and additionally get the estimate of the likelihood

$$\begin{aligned} L(y_{1:T}) &= \int p(y_{1:T}, h_{1:T}) dh_{1:T} = \int g(y_{1:T}|h_{1:T}) p(h_{1:T}) dh_{1:T} \\ &= \int g(y_1|h_1) p(h_1) \prod_{t=2}^T g(y_t|h_t) f(h_t|h_{t-1}) dh_1 \dots h_T. \end{aligned} \quad (19)$$

The basic procedure of particle filtering in this setting can be summarized by three crucial steps: prediction, updating, and resampling. The outline of a basic particle filter can be summarized in the following way.

- Initialization: given the prior distribution $\pi(\theta_0)$, we draw N independent random samples $\{h_i^0\}_{i=1}^N$; these samples we call *particles*.

- Prediction: we sample particles according to the importance density

$$h_t^{(i)} \sim q(h_t|h_{t-1}^{(i)}, y_t). \quad (20)$$

- Updating: During updating, we assign a weight $w_t^{(i)}$ to every particle

$$w_t^{(i)} = \frac{p(y_t|h_t^{(i)})f(h_t^{(i)}|h_{t-1}^{(i)})}{p(y_t|y_{1:t-1})q_t(h_t^{(i)}|h_{0:t-1}^{(i)})} \quad (21)$$

and normalize these weights to sum to 1. Every weight can be interpreted as our “confidence” about a particle.

- Resampling: resample the particles if the effective number of particles,

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\omega_t^{(i)})^2}, \quad (22)$$

is too low. In Equation (22), $\omega_t^{(i)}$ is the normalized weight of particle i at the time step k . The threshold for the resampling step is set depending on whether particle degeneracy is a problem. In general, we perform resampling when $N_{eff} < N/c$, where c is a constant.

The resampling step is performed to find the trade-off between two well-documented problems: particle *degeneracy* and particle *impoverishment* [28]. The former happens when the resampling step is ignored or is not performed frequently enough. In this case, one ends up with a particle set that has zero weights. The latter problem happens when the particle set is resampled too frequently; then, eventually one gets one particle with a large weight and hence the particle set lacks the diversity. The way to find the balance between these two problems is resampling when the efficient number of particles is smaller than a certain threshold.

In this paper, we consider two particle filters: bootstrap and auxiliary particle filters. A generic particle filter is presented in Algorithm A1 [28]. The bootstrap filter is a variation of a more general approach—sequential importance sampling (resampling). The distinction of the bootstrap filter is the proposal mechanism for the particles. In the bootstrap particle filter the proposals for the particles are made on the basis of the dynamics of the model $f(h_t|h_{t-1})$. If $q(h_t|y_t, h_{t-1}) = f(h_t|h_{t-1})$, then the term $\frac{f(h_t|h_{t-1})}{q(h_t|y_t, h_{t-1})}$ is equal to 1. In the case of the auxiliary particle filter, we also incorporate the current observation into the proposal mechanism $q(h_t|h_{t-1}, y_t)$. Incorporating the current observation into the proposal for the particles in some cases allows us to reduce the variance of the estimated likelihood. In our case, there is no analytical expression for the proposal density. In the next subsection, we discuss how it can be approximated as proposed in [23].

3.2.2. Auxiliary Particle Filter for SV Model

Incorporating knowledge of y_t into proposals for particles $q(h_t|h_{t-1}, y_t)$ can help to reduce the variance of the estimated likelihood and improve the approximation of the filtering distribution $p(h_t|y_{1:t})$. Note, however, that it is not always the case as has been shown in [27]. Only in the case of linear Gaussian state-space models does the proposal density from Equation (A2) have an analytical expression. Hence, for the stochastic volatility models, this term must be approximated. Pitt and Shephard [23] propose using non-

blind proposals for the next generation of particles by first expanding $\log g(y_{t+1}|h_{t+1})$ to a second-order term around μ_{t+1}^k via Taylor expansion

$$\begin{aligned} \log g(y_{t+1}|h_{t+1}, \mu_{t+1}^k) &= \log p(y_{t+1}|\mu_{t+1}^k)' \times \frac{\partial \log p(y_{t+1}|\mu_{t+1}^k)}{\partial h_{t+1}} + \\ &\frac{1}{2} \times (h_{t+1} - \mu_{t+1}^k)' \times \frac{\partial^2 \log p(y_{t+1}|\mu_{t+1}^k)}{\partial h_{t+1} h_{t+1}'} \times (h_{t+1} - \mu_{t+1}^k) \end{aligned} \tag{23}$$

For deriving the expression for $\log g(y_{t+1}|h_{t+1}, \mu_{t+1}^k)$, recall that $y_t \sim N(0, \exp(h_t))$ and hence

$$\begin{aligned} g(y_t|h_t) &= \frac{1}{\sqrt{2\pi \exp(h_t)}} \exp\left\{-\frac{y_t^2}{2 \exp(h_t)}\right\} = \\ &\frac{1}{\sqrt{2\pi}} \exp\left\{-\frac{y_t^2}{\exp(h_t)} - \frac{h_t}{2}\right\} \end{aligned} \tag{24}$$

and further note that $f(h_t|h_{t-1}) = N(\mu + \phi(h_{t-1} - \mu), \sigma_\eta^2)$; thus

$$f(h_t|h_{t-1}) = \frac{1}{\sqrt{2\pi\sigma_\eta^2}} \exp\left\{-\frac{(h_t - \mu - \phi(h_{t-1} - \mu))^2}{2\sigma_\eta^2}\right\}. \tag{25}$$

It follows that the proposal for particles at time $t + 1$ when taking into account the observation of the same period is

$$q(h_{t+1} | h_t^{(k)}, y_{t+1}; \mu_{t+1}^{(k)}) = N\left(\mu_{t+1}^{(k)} + \frac{\sigma^2}{2} \left(\frac{y_t^2}{\beta^2} \exp(-\mu_{t+1}^{(k)}) - 1\right), \sigma^2\right). \tag{26}$$

3.2.3. Metropolis–Hastings

In this section, we consider the problem of sampling from the posterior distribution and a general algorithm to construct such a sampling scheme. With the Metropolis–Hastings algorithm, we sample from the posterior distribution by proposing a transition $\theta \rightarrow \theta^*$ with the density $q(\theta^*|\theta)$, which we accept with probability

$$\alpha(\theta, \theta^*) = \min\left\{1, \frac{\tilde{p}(\theta^*) q(\theta|\theta^*)}{\tilde{p}(\theta) q(\theta^*|\theta)}\right\}, \tag{27}$$

where $\tilde{p}(\cdot)$ is a function proportional to our target distribution. A common choice for the proposal distribution is a random-walk, which we also use when applying PMCMC later in this paper, $q(\theta^*|\theta) = N(\theta^*|\theta, \Sigma)$. The Metropolis–Hastings algorithm is one of the off-the-shelf MCMC methods in the statistical community. It is quite general and can be applied to various problems. The implementation of the Metropolis–Hastings algorithm requires specification of multiple quantities. We need to specify a conditional density $q(\theta^*|\theta)$ that is a proposal distribution, generally $q(\theta^*|\theta)$ should be such that we can easily simulate from it. In many applications, including ours, it is reasonable to take the Gaussian distribution as proposal distribution. In this case, it is also symmetric, meaning $q(\theta^*|\theta) = q(\theta|\theta^*)$. The Metropolis–Hastings iteration is outlined in the Algorithm 1.

In this algorithm, $\alpha(\theta, \theta^*)$ is the Metropolis–Hastings acceptance probability, where θ is the current state of the chain and θ^* is the candidate state of the parameter vector. Generally, in the simulations, it is desired to have around 25% of proposed candidate values accepted [29]. The idea is that when the proposal steps are too large (we make a proposal that is far away from the current state, θ , in the Markov chain), we do not explore local regions sufficiently well; moreover many of the candidates are then very likely to be rejected. When the proposal steps are very small, the acceptance rate will be very high,

however, then we are not likely to leave regions of the local maximum or the convergence will happen very slowly.

Algorithm 1 Metropolis-Hastings Algorithm.

- 1: Given $\theta^{(t)}$,
- 2: Generate $\theta_t^* \sim q(\theta^* | \theta^{(t)})$,
- 3: Take

$$\theta^{(t+1)} = \begin{cases} \theta_t^*, & \text{with probability } \alpha(\theta^{(t)}, \theta_t^*) \\ \theta^{(t)} & \text{with probability } 1 - \alpha(\theta^{(t)}, \theta_t^*), \end{cases}$$

where

$$\alpha(\theta, \theta^*) = \min\left(1, \frac{\tilde{p}(\theta^*) q(\theta | \theta^*)}{\tilde{p}(\theta) q(\theta^* | \theta)}\right)$$

The performance of Metropolis–Hastings depends on the choice of $q(\cdot)$ proposal distribution. In the simulation studies, we consider random-walk proposals of the form $\theta_{i+1}^* = \theta_i + \epsilon_i$, where i is iteration of the algorithm and ϵ_i is assumed to be Gaussian. More information on the theoretical properties of this algorithm can be found in [30].

3.2.4. Particle Metropolis-Hastings

Particle Markov Chain Monte Carlo (PMCMC) methods were introduced in [4]. The basic idea is that MCMC methods, and in particular, Metropolis–Hastings algorithm, which is of interest to us, can be combined with Sequential Monte Carlo to make draws from the posterior distributions of the parameters. Algorithm 2 presents the particle Metropolis–Hastings algorithm. The difference from the standard Metropolis-Hastings is in the quantity $\hat{p}_{\theta^*}(y_{1:T})$, which is the estimate of the likelihood obtained with a particle filter conditioning on the parameters vector θ . In this algorithm, $q(\theta(i-1)|\theta^*)$ is the proposal distribution (which cancels out when it is symmetric), and $\pi(\cdot)$ is prior distribution.

Algorithm 2 Particle Metropolis-Hastings.

- 1: Initialize algorithm at $i = 0$ and initialize parameters $\theta(0)$
- 2: Run an SMC algorithm targeting $p_{\theta(0)}(h_{1:T} | y_{1:T})$, sample $h_{1:T}^{(0)} \sim \hat{p}_{\theta(0)}(\cdot | y_{1:T})$ and let $\hat{p}_{\theta}^{(0)}(y_{1:T})$ denote the marginal likelihood estimate for the initialized parameters
- 3: **for** $i = 1, \dots, M$ **do**
- 4: Generate $\theta^* \sim q(\theta^* | \theta^{(i-1)})$,
- 5: Run an SMC algorithm targeting $p_{\theta^*}(h_{1:T} | y_{1:T})$, sample $h_{1:T}^* \sim \hat{p}_{\theta^*}(\cdot | y_{1:T})$ and let $\hat{p}_{\theta^*}(y_{1:T})$ denote the marginal likelihood estimate for the proposed parameters θ^*
- 6: With probability

$$\min\left\{1, \frac{\hat{p}_{\theta^*}(y_{1:T}) \pi(\theta^*) q(\theta^{(i-1)} | \theta^*)}{\hat{p}_{\theta^{(i-1)}}(y_{1:T}) \pi(\theta^{(i-1)}) q(\theta^* | \theta^{(i-1)})}\right\} \quad (28)$$

- 7: Set $\theta^{(i)} = \theta^*$, $h_{1:T}^{(i)} = h_{1:T}^*$ and $\hat{p}_{\theta^{(i)}}(y_{1:T}) = \hat{p}_{\theta^*}(y_{1:T})$;
 - 8: Otherwise set $\theta^{(i)} = \theta^{(i-1)}$, $h_{1:T}^{(i)} = h_{1:T}^{(i-1)}$ and $\hat{p}_{\theta^{(i)}}(y_{1:T}) = \hat{p}_{\theta^{(i-1)}}(y_{1:T})$.
 - 9: **end for**
-

3.3. MCMC with Gradient Information

In this section, we discuss Riemann Manifold Langevin Hamiltonian Monte Carlo methods that are introduced in [5] and in particular can be applied to stochastic volatility models.

The method originates in physics statistical literature and provides a tool that allows one to make large transitions with high acceptance probability, something that standard

methods such as Metropolis–Hastings fail to achieve. The idea of HMC is based on relation between differential geometry and statistical theory (MCMC in particular). Girolami and Calderhead [5] propose the Metropolis-adjusted Langevin algorithm and Hamiltonian Monte Carlo sampling algorithms that are defined on the Riemann manifold. Their methods allow us to overcome the problem of sampling from high-dimensional densities that may show strong correlation. We further provide the general background and summary of the algorithms together with the necessary quantities for their implementation in the case of stochastic volatility models. It is not our goal to provide theoretical foundations of these methods in this article. For deeper theoretical foundations, see [31–33].

In standard MCMC setting, one uses probability distribution to make a proposal for the next state of the Markov chain. Hamiltonian Monte Carlo methods exploit physical system dynamics to make proposals for the next state. It can improve the mixing drastically and result in a more efficient algorithm. Especially since we are interested in multivariate modeling, a more efficient exploration of the posterior distribution is of interest. Once the dimension of the model grows with standard random walk, it is very hard to make proposals that would be accepted frequently enough and result in a good mixing Markov chain. We first introduce some basic ideas on which Hamiltonian Monte Carlo method is built; for an extensive introduction, we refer to [33].

3.3.1. Metropolis-Adjusted Langevin Algorithm

Previously we have discussed the Metropolis-Hastings algorithm. The idea of the Metropolis-Hastings algorithm is to make a new proposal θ^* using random walk. Then this proposal is accepted with probability.

$$\alpha(\theta, \theta^*) = \min \left\{ 1, \frac{\tilde{p}(\theta^*) q(\theta|\theta^*)}{\tilde{p}(\theta) q(\theta^*|\theta)} \right\}. \quad (29)$$

Although this algorithm benefits from desirable theoretical guarantees, the random walk proposal is not efficient, especially when the number of parameters in the model becomes large. Metropolis-adjusted Langevin algorithm (MALA), originally proposed in [34], is designed to solve the same problem—sample from the target distribution. The big advantage of MALA in comparison to Metropolis–Hastings is the construction for the proposal of the candidate parameter θ^* . The proposal mechanism for MALA originates from the stochastic differential equation based on Langevin diffusion; the proposal mechanism reads

$$\theta^* = \theta^n + \epsilon^2 \nabla_{\theta} L(\theta^n) / 2 + \epsilon z^n, \quad (30)$$

where we define $L(\theta^n) = \log(p(\theta))$ and $z \sim N(z|0, I)$ and ϵ —integration step size. Convergence for this proposal is not guaranteed unless we employ a Metropolis acceptance probability after every integration step. For convenience, let us define

$$\mu(\theta^n, \epsilon) = \theta^n + \frac{\epsilon^2}{2} \nabla_{\theta} L(\theta^n); \quad (31)$$

then the proposal density can be written as $q(\theta^*|\theta^n) = N(\theta^*|\mu(\theta^n, \epsilon), \epsilon^2 I)$. The standard acceptance probability follows

$$\min \{ 1, p(\theta^*) q(\theta^n|\theta^*) / p(\theta^n) q(\theta^*|\theta^n) \}. \quad (32)$$

The type of proposal in Equation (29) is inefficient for strongly correlated parameters θ . To solve this issue, one can use a preconditioning matrix M

$$\theta^* = \theta^n + \epsilon^2 M \nabla_{\theta} L(\theta^n) / 2 + \epsilon \sqrt{M} z^n. \quad (33)$$

Unfortunately there is no principled way to choose matrix M . As we will see later, HMC encounters the same problem. Generally, MALA iterates between two general steps.

First, Langevin dynamics is used for the proposals, and it exploits the gradients of the target. Second, the proposals are accepted or rejected similarly to those of the Metropolis–Hastings algorithm.

3.3.2. Hamiltonian Monte Carlo Algorithm

The HMC algorithm [31] also uses gradient information for constructing the proposal of the parameters in the MCMC scheme. In particular, it exploits the ideas from simulating the behavior of the physical systems. Similarly to describing the behavior of the physical system, HMC performs sampling by exploiting *Hamiltonian dynamics*. A conceptual introduction to this class of methods and its relationship to differential geometry can be found in [33]. In this section, we discuss the general idea behind the algorithm without performing detailed derivations. We focus on the final proposal machinery that can be used in practice and investigate which quantities need to be manually computed before implementing the algorithm and which variables need to be calibrated for the successful performance of the algorithm. First, let us consider a general set-up. In Hamiltonian Monte Carlo, we consider a Hamiltonian function

$$H(\boldsymbol{\theta}, \mathbf{p}) = -\log p(\boldsymbol{\theta}) + \frac{1}{2} \log\{(2\pi)^D |\mathbf{M}|\} + \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}, \quad (34)$$

which consists of potential energy in the system $E(\boldsymbol{\theta}) = -L(\boldsymbol{\theta})$ and kinetic energy $K(\mathbf{p}) = \frac{1}{2} \log\{(2\pi)^D |\mathbf{M}|\} + \frac{1}{2} \mathbf{p}^T \mathbf{M}^{-1} \mathbf{p}$; variables \mathbf{p} are called momentum variables. The dynamics of the system then evolves according to Hamiltonian equations

$$\frac{d\boldsymbol{\theta}}{d\tau} = \frac{\partial H}{\partial \mathbf{p}} = \mathbf{M}^{-1} \mathbf{p}, \quad (35)$$

$$\frac{d\mathbf{p}}{d\tau} = -\frac{\partial H}{\partial \boldsymbol{\theta}} = \nabla_{\boldsymbol{\theta}} L(\boldsymbol{\theta}), \quad (36)$$

where by τ in physical interpretation of the system we denote continuous time. Practical implementation requires discretization, and the commonly used scheme for this purpose is the leapfrog discretization:

$$\mathbf{p}(\tau + \epsilon/2) = \mathbf{p}(\tau) + \epsilon \nabla_{\boldsymbol{\theta}} L\{\boldsymbol{\theta}(\tau)\} / 2, \quad (37)$$

$$\boldsymbol{\theta}(\tau + \epsilon) = \boldsymbol{\theta}(\tau) + \epsilon \mathbf{M}^{-1} \mathbf{p}(\tau + \epsilon/2), \quad (38)$$

$$\mathbf{p}(\tau + \epsilon) = \mathbf{p}(\tau + \epsilon/2) + \epsilon \nabla_{\boldsymbol{\theta}} L\{\boldsymbol{\theta}(\tau + \epsilon)\} / 2. \quad (39)$$

This scheme does not sample from the target distribution and to correct for that, implementation of Metropolis acceptance probability is necessary. For a proposal $(\boldsymbol{\theta}, \mathbf{p}) \rightarrow (\boldsymbol{\theta}^*, \mathbf{p}^*)$, acceptance probability in this algorithm is defined as

$$\min\{1, \exp\{-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + H(\boldsymbol{\theta}, \mathbf{p})\}\}.$$

Thus, HMC iterates between updating momentum variables, proposal for the parameter values, additional update to the momentum variables, and then an acceptance/rejection step. The Gibbs sampler provides a good understanding for the system evolution in this algorithm:

$$\mathbf{p}^{n+1} | \boldsymbol{\theta}^n \sim p(\mathbf{p}^{n+1} | \boldsymbol{\theta}^n) = p(\mathbf{p}^{n+1}) = N(\mathbf{p}^{n+1} | 0, \mathbf{M}), \quad (40)$$

$$\boldsymbol{\theta}^{n+1} | \mathbf{p}^{n+1} \sim p(\boldsymbol{\theta}^{n+1} | \mathbf{p}^{n+1}) \quad (41)$$

Similarly to MALA, the choice of matrix \mathbf{M} is crucial for good performance of HMC. While the choice of the step size and the leapfrog steps can be tuned relatively easily by considering acceptance rate, the choice of the matrix \mathbf{M} is challenging, and there is no principled way to define it. Leapfrog step and step size proposal are two variables that need to be calibrated when implementing HMC. Usually, different combinations of these

two variables are considered, and the combination leading to the highest acceptance rate is picked.

3.4. Riemann Manifold Hamiltonian Monte Carlo

The further improvement of HMC and MALA can be done by defining the algorithms on Riemann manifold instead of Euclidean space. Proposals guided by Riemann metric instead of Euclidean distance have the potential to explore parameter space more efficiently, especially in the cases when the target density is high-dimensional or exhibits strong correlation [5]. The method originally proposed in [5] and multiple algorithms were compared in the paper: MALA, MMALA, HMC, and RMHMC. For detailed comparison between these methods, we refer to [5], while in our simulation studies, we will focus on comparing RMHMC and particle Metropolis–Hastings for the estimation of parameters in stochastic volatility models.

Girolami and Calderhead [5] define HMC methods in the form of Riemann manifold, and this can be seen as generalization of HMC. The Hamiltonian on the Riemann manifold is defined as follows

$$H(\boldsymbol{\theta}, \mathbf{p}) = -\log p(\boldsymbol{\theta}) + \frac{1}{2} \log((2\pi)^n |G(\boldsymbol{\theta})|) + \frac{1}{2} \mathbf{p}^T G(\boldsymbol{\theta}) \mathbf{p} \quad (42)$$

with $\exp(-H(\boldsymbol{\theta}, \mathbf{p})) = p(\boldsymbol{\theta}, \mathbf{p}) = p(\boldsymbol{\theta})p(\mathbf{p} | \boldsymbol{\theta})$ and the marginal target density

$$\begin{aligned} p(\boldsymbol{\theta}) &\propto \int \exp(-H(\boldsymbol{\theta}, \mathbf{p})) d\mathbf{p} = \frac{\exp\{\log p(\boldsymbol{\theta})\}}{\sqrt{2\pi^n |G(\boldsymbol{\theta})|}} \int \exp\left\{-\frac{1}{2} \mathbf{p}^T G(\boldsymbol{\theta})^{-1} \mathbf{p}\right\} d\mathbf{p} \\ &= \exp(\log p(\boldsymbol{\theta})). \end{aligned} \quad (43)$$

The general idea behind the updates in RMHMC is similar to that of HMC, and the updates for the momentum variables and parameters of the model are defined in Equations (44)–(46).

$$\mathbf{p}\left(\tau + \frac{\epsilon}{2}\right) = \mathbf{p}(\tau) - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} H\left\{\boldsymbol{\theta}(\tau), \mathbf{p}\left(\tau + \frac{\epsilon}{2}\right)\right\}, \quad (44)$$

$$\boldsymbol{\theta}(\tau + \epsilon) = \boldsymbol{\theta}(\tau) + \epsilon/2 \left[\nabla_{\mathbf{p}} H\left\{\boldsymbol{\theta}(\tau), \mathbf{p}\left(\tau + \frac{\epsilon}{2}\right)\right\} + \nabla_{\mathbf{p}} H\left\{\boldsymbol{\theta}(\tau + \epsilon), \mathbf{p}\left(\tau + \frac{\epsilon}{2}\right)\right\} \right], \quad (45)$$

$$\mathbf{p}(\tau + \epsilon) = \mathbf{p}\left(\tau + \frac{\epsilon}{2}\right) - \frac{\epsilon}{2} \nabla_{\boldsymbol{\theta}} H\left\{\boldsymbol{\theta}(\tau + \epsilon), \mathbf{p}\left(\tau + \frac{\epsilon}{2}\right)\right\} \quad (46)$$

Therefore, as in standard HMC algorithm, we iterate between half-step update of the momentum variables, and then we update position variables, and we finish iteration with additional half-step update of the momentum variables and Metropolis acceptance/rejection step with the probability

$$\min\{1, \exp\{-H(\boldsymbol{\theta}^*, \mathbf{p}^*) + H(\boldsymbol{\theta}^n, \mathbf{p}^{n+1})\}\}.$$

Similarly to HMC, RMHMC can be viewed as a Gibbs sampling scheme

$$\mathbf{p}^{n+1} | \boldsymbol{\theta}^n \sim p(\mathbf{p}^{n+1} | \boldsymbol{\theta}^n) = N\{\mathbf{p}^{n+1} | 0, G(\boldsymbol{\theta}^n)\}, \quad (47)$$

$$\boldsymbol{\theta}^{n+1} | \mathbf{p}^{n+1} \sim p(\boldsymbol{\theta}^{n+1} | \mathbf{p}^{n+1}). \quad (48)$$

Recall that in the case of MALA and HMC, matrix \mathbf{M} has to be chosen manually and there is no principled way to choose it. In RMHMC, matrix $\mathbf{G}(\boldsymbol{\theta})$ is defined at each step by underlying geometry; see for more details [5]. Below we discuss quantities that need to be computed for the implementation of RMHMC in the case of stochastic volatility model and in particular $\mathbf{G}(\boldsymbol{\theta})$.

Recall stochastic volatility model parametrized through β

$$y_t = \beta \exp(h_t/2)\epsilon_t, \tag{49}$$

$$h_{t+1} = \phi h_t + \eta_{t+1}, \tag{50}$$

$\epsilon_t \sim N(0, 1), \eta_t \sim N(0, \sigma^2)$, with $h_1 \sim N(0, \sigma^2/(1 - \phi^2))$.

The joint likelihood of the model is

$$p(y, h, \beta, \phi, \sigma) = \prod_{t=1}^T p(y_t | h_t, \beta) \prod_{t=2}^T p(h_t | h_{t-1}, \phi, \sigma) \pi(\beta) \pi(\phi) \pi(\sigma) \tag{51}$$

The prior distributions are chosen as follows

$$\beta \propto \exp(\beta), \quad \sigma^2 \sim \text{Inv} - \chi^2(10, 0.05), \quad (\phi + 1)/2 \sim \text{Beta}(20, 1.5). \tag{52}$$

Further, following [5], we write the partial derivatives for $L = p(y, h | \beta, \phi, \sigma)$

$$\frac{\partial L}{\partial \beta} = -\frac{T}{\beta} + \sum_{t=1}^T \frac{y^2}{\beta^3 \exp(h_t)}, \tag{53}$$

$$\frac{\partial L}{\partial \phi} = -\frac{\phi}{(1 - \phi^2)} + \frac{\phi h_1^2}{\sigma^2} + \sum_{t=2}^T \frac{h_{t-1}(h_t - \phi h_{t-1})}{\sigma^2}, \tag{54}$$

$$\frac{\partial L}{\partial \sigma} = -\frac{T}{\sigma} + \frac{h_1^2(1 - \phi^2)}{\sigma^3} + \sum_{t=2}^T \frac{(h_t - \phi h_{t-1})^2}{\sigma^3}. \tag{55}$$

To implement the algorithms, we require the expressions for the individual components of the metric tensor for the likelihood. Following [5], the expressions are

$$\mathbb{E} \left\{ \frac{\partial L}{\partial \beta} \frac{\partial L}{\partial \beta} \right\} = \frac{2T}{\beta^2}, \quad \mathbb{E} \left\{ \frac{\partial L}{\partial \sigma} \frac{\partial L}{\partial \sigma} \right\} = \frac{2T}{\sigma^2}, \quad \mathbb{E} \left\{ \frac{\partial L}{\partial \beta} \frac{\partial L}{\partial \sigma} \right\} = \mathbb{E} \left\{ \frac{\partial L}{\partial \beta} \frac{\partial L}{\partial \phi} \right\} = 0, \tag{56}$$

$$\mathbb{E} \left\{ \frac{\partial L}{\partial \sigma} \frac{\partial L}{\partial \phi} \right\} = \frac{2\phi}{\sigma^3(1 - \phi^2)}, \quad \mathbb{E} \left\{ \frac{\partial L}{\partial \phi} \frac{\partial L}{\partial \phi} \right\} = \frac{2\phi^2}{(1 - \phi^2)^2} + \frac{T - 1}{1 - \phi^2}. \tag{57}$$

Furthermore, the expressions for the metric tensor for the likelihood and its partial derivatives follow

$$G(\phi, \sigma, \beta) = \begin{bmatrix} \frac{2T}{\beta^2} & 0 & 0 \\ 0 & \frac{2T}{\sigma^2} & \frac{2\phi}{\sigma^3(1 - \phi^2)} \\ 0 & \frac{2\phi}{\sigma^3(1 - \phi^2)} & \frac{2\phi^2}{(1 - \phi^2)^2} + \frac{T - 1}{1 - \phi^2} \end{bmatrix}, \tag{58}$$

$$\frac{\partial G}{\partial \beta} = \begin{bmatrix} -\frac{4T}{\beta^3} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \tag{59}$$

$$\frac{\partial G}{\partial \sigma} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & -\frac{4T}{\sigma^3} & -\frac{6\phi}{\sigma^4(1 - \phi^2)} \\ 0 & -\frac{6\phi}{\sigma^4(1 - \phi^2)} & 0 \end{bmatrix}, \tag{60}$$

$$\frac{\partial G}{\partial \phi} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{2}{\sigma^3(1 - \phi^2)} + \frac{4\phi^2}{\sigma^3(1 - \phi^2)^2} \\ 0 & \frac{2}{\sigma^3(1 - \phi^2)} + \frac{4\phi^2}{\sigma^3(1 - \phi^2)^2} & \frac{2\phi(1 + T)}{(1 - \phi^2)^2} + \frac{6\phi^3}{(1 - \phi^2)^3} \end{bmatrix}. \tag{61}$$

The proposal machinery in RMHMC provides advantages for exploring parameter space efficiently. However, it is not easily adaptable for different model specifications, especially when increasing the model's dimensionality, as we discussed in Section 1. In particular, although matrix G can be computed in the multivariate model specified in Equations (12) and (13) exactly, it scales quadratically with the number of parameters. This might be one of the reasons why the method has not been used on multivariate stochastic volatility models we introduced in Section 1. However, probabilistic programming languages [35,36] and automatic differentiation possibilities developed in recent years allow the efficient and adaptable implementation of these algorithms in practice.

3.5. Integrated Nested Laplace Approximation

Integrated Nested Laplace Approximation was introduced in [7]. The method is based on the nested version of the classical Laplace approximation and was introduced for latent Gaussian models (LGMs). It became a popular approach in Bayesian inference due to its good performance in the variety of models in the class of LGMs and its computational advantages over other methods in Bayesian literature. The computational appeal of this method comes from the possibility of exploiting sparse matrix computations when evaluating certain approximations. INLA has found its applications in many fields in the models where high-dimensional problems arise. Stochastic volatility models have been analyzed using INLA in [37,38]. Bivariate stochastic volatility model has been considered in [39], where the authors present and solve some issues that arise in using INLA in the multivariate case of the model. One of the conclusions of this study was that INLA loses its computational advantage with increased dimensionality of the stochastic volatility model. We further discuss the details of the method and the implementation shortcomings in a multivariate case and present the reader with a simulation study that illustrates the discussed approach's performance.

Stochastic volatility model can be written in the form of LGMs

$$y \mid h, \theta_1 \sim \prod_{i \in \mathcal{I}} \pi(y_i \mid h_i, \theta_1), \quad (62)$$

$$h \mid \theta_2 \sim N(\mu(\theta_2), Q^{-1}(\theta_2)). \quad (63)$$

As before, y_t is the data that we observe and h_t is the latent volatility process, and we are interested in the posterior distribution of the parameters of the model θ and the latent process given the data

$$p(h, \theta \mid y) \propto p(\theta) p(h \mid \theta) \prod_{t=1}^T p(y \mid h_t, \theta). \quad (64)$$

The outline of the INLA approach can be summarized in the following steps [7,37]

1. Build an approximation $p(\theta \mid y)$
2. Build an approximation to $p(h_t \mid \theta, y)$
3. Compute an approximation to $p(h_t \mid y)$ using the approximations from steps 1 and 2.

The first approximation $p(\theta \mid y)$ relies on the Gaussian approximation of the form

$$p(x \mid y, \theta) \propto \exp \left\{ -\frac{1}{2} x^T Q x + \sum g_t(h_t) \right\}, \quad (65)$$

where $x = (\mu, h)$, $g_t(h_t) = \log p(y_t \mid h_t, \theta)$. By matching the mode and curvature in the mode, we obtain the Gaussian approximation for our model

$$\tilde{p}_G(x \mid y, \theta) = K_1 \exp \left\{ -\frac{1}{2} (x - \mu)^T (Q + \text{diag}(c)) (x - m) \right\}, \quad (66)$$

where K_1 is a normalizing constant, m is the modal value of $p(x | y, \theta)$, the vector c contains the second order terms in the Taylor expansion of $\sum g_t(h_t)$ at the modal value m , and Q is the precision matrix that has the form

$$Q = \begin{bmatrix} 1 & -\phi & & & & \\ -\phi & 1 + \phi^2 & -\phi & & & \\ & \ddots & \ddots & \ddots & & \\ & & -\phi & 1 + \phi^2 & -\phi & \\ & & & -\phi & 1 & \end{bmatrix}. \quad (67)$$

The sparsity of the precision matrix above allows one to exploit efficient sparse matrix computational methods and thus gain computational speed. Note that in the multivariate case, this advantage disappears since the matrix Q is not sparse anymore.

When it comes to the estimation of stochastic volatility models, approximation of the marginals $p(h_t | \theta, y)$ is always the most challenging task. The solution that is proposed in [7] is (simplified) Laplace approximation of the form

$$\log \tilde{p}_{SLA}(x_t | \theta, y) = const - \frac{1}{2}x_t^2 + \gamma_t^{(1)}(\theta)x_t + \frac{1}{6}x_t^3\gamma_t^{(3)}(\theta) + \dots, \quad (68)$$

where $\gamma_t^{(1)}$ and $\gamma_t^{(3)}$ are the terms in the Taylor expansion. The final step of the method is to approximate $p(x_t | y)$ with the numerical integration scheme

$$\tilde{p}(x_t | y) = \sum_k \tilde{p}(x_t | \theta^k, y) \tilde{p}(\theta^k | y) \Delta_k, \quad (69)$$

for some θ^k of θ , where θ^k is selected by creating a grid of points that covers the area of high density for $\tilde{p}(\theta | y)$. For more details on implementation of the simplified Laplace approximation and the selection of grid of points for θ^k , see [7,37].

3.6. Fixed-Form Variational Bayes

In this section, we discuss how the posterior distribution can be approximated using the fixed-form variational Bayes method proposed in [6]. The general idea of fixed-form variational inference consists in assuming a certain factorization of the prior distribution, which naturally leads to the factorized structure of the posterior. The factorizing distributions of the posterior are then assumed to come from a certain parametric family of distributions (for example, exponential) and instead of a sampling task, as in the previous section, we would perform the optimization task of minimizing the distance between the approximating distribution and the unknown posterior distribution.

As before, assume we observe a process $\{y_t\}_{t=1}^T$ that is driven by an unobservable or latent process $\{h_t\}_{t=1}^T$. Recall that Bayes' rule gives us the posterior distribution of the parameters of the system

$$p(h | y) \propto g(y | h)\pi(h). \quad (70)$$

In the Bayesian framework, we formulate our prior beliefs, which we update once we acquire more data. In general, Variational Bayes methods focus on approximating the posterior distribution $p(h | y)$ with some distribution $q(h | y)$. Further, it is common to choose blocks of the parameters and impose independence for these blocks

$$p(h | y) \approx q(h | y) = q(h_1 | y)q(h_2 | y). \quad (71)$$

By construction, the posterior of the blocks of the parameters is independent. In the literature, this is referred to as *the mean-field assumption*. To find the optimal approximation, we minimize the Kullback–Leibler (KL) divergence from $q(h | y)$ to $p(h | y)$

$$\tilde{p}(h | y) = \arg \min_{q(h_1|\cdot)q(h_2|\cdot)} KL(q(h_1 | y)q(h_2 | y) || p(h | y)). \quad (72)$$

Distributional approximation can be viewed as an optimization problem; i.e., an optimal distribution has to be chosen from the space of all possible distributions, and the KL divergence is chosen as a loss function [40]. Salimans et al. [6] propose a specific approach to the minimization problem of KL divergence, which is based on the similarities between the optimal solution to the problem and linear regression. The general idea of the approach is summarized as comprising the following steps:

- initialize all the posterior approximations $q(\theta)$;
- iterate over the parameters updating every one of them given the others;
- repeat until convergence.

Consider the stochastic volatility model

$$y_t = \beta \exp(h_t/2) \epsilon_t \quad (73)$$

$$h_{t+1} = \phi h_t + \eta_{t+1}, \quad (74)$$

with $h_1 \sim N(0, \sigma^2/(1 - \phi^2))$ and $\epsilon_t \sim N(0, 1)$, $\eta_t \sim N(0, \sigma_\eta^2)$. We specify our a priori beliefs in the following manner

$$p(\beta) \propto \beta^{-1} \quad (\phi + 1)/2 \sim \text{Beta}(20, 1.5), \quad \sigma^2 \sim \text{IG}(5, 0.25). \quad (75)$$

To apply the Variational Bayes method, we need to specify the posterior approximations $q(\theta)$. It is convenient to assume a hierarchical structure of the prior, in which case it factorizes to

$$p(\phi, \sigma^2, \beta, f) = p(\phi) p(\sigma^2) p(f | \phi, \sigma^2) p(y | f), \quad (76)$$

where $f = (\log(\beta), h')$. The hierarchical structure of the prior leads to the following factorization of the posterior approximation

$$q_\xi(\sigma_\eta^2, f | f) = q_\xi(\sigma_\eta^2) q_\xi(f | \phi, \sigma^2) = \frac{q_\xi(\sigma_\eta^2 | \phi) p(f | \phi, \sigma^2) q_\xi(y | f)}{q_\xi(y | \phi, \sigma^2)}. \quad (77)$$

Thus, the posterior approximations can be chosen as follows

$$q_\xi((\phi + 1)/2) = \text{Beta}(\xi_1, \xi_2), \quad (78)$$

$$q_\xi(\sigma^2 | \phi) \sim \text{IG}(\xi_3, \xi_4 + \xi_5 \phi^2), \quad (79)$$

$$q(\log(\beta), h | \phi, \sigma^2) = N(m, V), \quad (80)$$

where

$$V^{-1} = P(\phi, \sigma^2) + \xi_6, \quad m = V^{-1} \xi_7,$$

with $P(\phi, \sigma^2)$ precision matrix of $p(\log(\beta), h | \phi, \sigma^2)$.

Once the posterior approximations are initialized, we proceed with the next step and iterate over the parameters. The parameters are updated in blocks that correspond to the factorization of the posterior approximations. First, we update the block for the persistence parameter in the latent process $q_\xi(\phi)$

$$\phi^* = s_1(\xi, z_1^*), \quad \text{with } s_1() \text{ and } z_1^* \text{ such that } \sigma^{2*} \sim q_\xi(\sigma^2 | \phi^*), \quad (81)$$

$$\sigma^{2*} = s_2(\xi, z_2^*, \phi^*), \quad \text{with } s_2() \text{ and } z_2^* \text{ such that } \sigma^{2*} \sim q_\xi(\sigma^2 | \phi^*), \quad (82)$$

$$\hat{C}_1 = \nabla_\xi [s_1(\xi, z_1^*)] \nabla_\phi [T_1(\phi^*)], \quad (83)$$

$$\hat{g}_1 \approx \nabla_\xi [s_1(\xi, z_1^*)] \{ \nabla_\phi [\log p(\phi^*) + \log q_\xi(y | \phi^*, \sigma^{2*}) - \log q_\xi(\sigma^{2*} | \phi^*)] \}. \quad (84)$$

Second, we update the block for the variance of the latent process $q_\xi(\sigma^2 | \phi)$

$$\hat{C}_2 = \nabla_\xi [s_2(\xi, z_2^*, \phi^*)] \nabla_{\sigma^2} [T_2(\sigma^{2*})] \quad (85)$$

$$\xi_2 \approx \nabla_{\xi} [s_2(\xi, z_2^*, \phi^*)] \nabla_{\sigma^2} [\log p(\sigma^{2*}) + \log q_{\xi}(y | \phi^*, \sigma^{2*})], \quad (86)$$

where $T_2(\sigma^{2*})$ are the sufficient statistics of $q_{\xi}(\sigma^2 | \phi)$. The last update is the update of the likelihood approximation

$$a_{t+1} = (1 - \omega)a_t + \omega \mathbb{E}_{q_{\xi}(f|\phi^*, \sigma^{2*})} [\nabla_f \log p(y | f)], \quad (87)$$

$$z_{t+1} = (1 - \omega)z_t + \omega \mathbb{E}_{q_{\xi}(f|\phi^*, \sigma^{2*})} [f], \quad (88)$$

$$\tilde{\xi}_{6,t+1} = (1 - \omega)\tilde{\xi}_{6,t} - \omega \mathbb{E}_{q_{\xi}(f|\phi^*, \sigma^{2*})} [\nabla_f \nabla_f \log p(y | f)], \quad (89)$$

$$\tilde{\xi}_{7,t+1} = a_{t+1} + \tilde{\xi}_{6,t+1} z_{t+1}. \quad (90)$$

For more extensive derivations of the updates, we refer the reader to [6]. Further, one might wonder how the latent process is estimated in this procedure. Salimans et al. [6] propose using the Kalman filter to estimate the filtering distribution. Even though it is a valid approach that is also undertaken in quasi-maximum likelihood method [41], its weakness lies in the linearization of the observation equation which implies that the distribution of the noise process is not longer Gaussian.

4. Results

In this section, we present results for the comparison of the discussed methods. We compare two particle filters (bootstrap and auxiliary particle filters) on the basis of bias, variance and on the estimated effective number of particles. We choose the better performing procedure of the two for using in the particle Metropolis–Hastings algorithm. We compare particle Metropolis–Hastings (PMH), Riemann Manifold Hamiltonian Monte Carlo (RMHMC), integrated nested Laplace approximation (INLA), and fixed-form variational Bayes (VB) on the basis of how well the posterior distributions obtained with these methods capture the ground truth (e.g., true parameter values). The ability of the methods to recover ground truth is assessed based on five simulated data sets with different underlying parameters. We additionally provide effective sample sizes for the comparison of the sampling methods (PMH and RMHMC). For illustration purposes, we also provide comparison on two real-world data sets.

4.1. Variance of the Estimated Likelihood

As we mentioned before, the marginal likelihood can be approximated sequentially through particle filtering. The marginal likelihood approximation of the parameters θ reads

$$p(y_{1:T} | \theta) \approx \prod_t \hat{p}(y_t | y_{1:t-1}, \theta), \quad (91)$$

where the right hand side is obtained by running particle filter presented in Algorithm A1. In practice, usually the log-likelihood

$$\log p_{\theta}(y_{1:T}) = \log p_{\theta}(y_1) + \sum_{t=2}^T \log p_{\theta}(y_t | y_{1:t-1}) \quad (92)$$

is estimated for the purpose of numerical stability (as the product of small weights would lead to unstable results). The estimate of the log-likelihood is the by-product of the particle filtering, as it is the average over log-weights that are assigned to the particles at every time step. In this section, we compare the bootstrap (BPF) and auxiliary particle filters (APF) in terms of bias, variance, and number of effective particles. Both of them can be used for obtaining simulated likelihood estimates, which can be further used in the particle

Metropolis–Hastings algorithm. We denote by \hat{L} the estimate of the likelihood obtained with a particle filter. Then, the bias and the variance can be estimated as follows

$$Bias = 5000^{-1} \sum_{i=1}^K \sum_{j=1}^M (\log \hat{L}^j - \log \bar{L}(y^i)), \quad (93)$$

$$Variance = 5000^{-1} \sum_{i=1}^K \sum_{j=1}^M (\log \hat{L}^j - \log \bar{L}(y^i))^2, \quad (94)$$

where y_i is the i -th time series, and $\log L$ is the “true” log-likelihood value. For the comparison, we use $K = 50$ different time series generated from the stochastic volatility model and $M = 100$ Monte Carlo iterations. We use $N = 100$, $N = 1000$, and $N = 10,000$ number of particles for this study. As the true value of the likelihood is not available, we substitute it with an estimate that is obtained with $N = 1,000,000$ number of particles. First, we conduct the analysis of the variance of the estimated likelihood in true parameter values as discussed in [24]. The authors of, ref. [27] showed theoretically that the asymptotic variance is not always smaller for the APF in comparison to the BPF. We run additional simulation studies to examine whether the variance of the estimated likelihood varies in the parameter space. Since we are interested in using the estimated likelihood in the Markov Chain Monte Carlo setting, it is relevant how the variance behaves in different points of the parameter space. If we start far away from the true value and the variance of the estimated likelihood is larger in that part of the parameter space, it can affect the convergence and calibration of the algorithm. Table 1 shows variance of the estimated likelihood for bootstrap and auxiliary particle filters. N indicates the number of the particles that we used for the estimation of the likelihood. It is clear that, on average, APF performs better in terms of the variance of the estimated likelihood. Table 2 indicates results for a similar experiment, but on the level of individual times series. We consider different data-generating processes and find that, in particular, higher variance of the latent volatility process is associated with higher variance of the estimated likelihood. Finally, in Figures 2–4, we illustrate that the variance of the estimated likelihood changes depending on the location in the parameter space, and these changes can be specific to a data-generating process. These figures correspond to the experiments with time series 2, 3, and 4 from Table 2. The likelihood was estimated with $N = 1000$ particles. We observe that the variance of the latent process has a strong effect on the landscape of the variance of the estimated likelihood in the parameter space. From Figure 4c,d, we see that the variance of the estimated likelihood obtained with the bootstrap particle filter appears to be more strongly affected by the location in the parameter space than the variance of the estimated likelihood obtained with the auxiliary particle filter. In Figure 3, we observe that the variance of the estimated likelihood is affected by the scale parameter β in the case of auxiliary particle filter, but not so much in the case of the bootstrap particle filter. Thus, initialization of PMCMC and the choice of number of particles should be considered with care for the optimal performance of the algorithm as the variance of the estimated likelihood can differ in the parameter space, and these changes can vary across different time series.

Table 1. Variance, bias, and number of effective particles N_{eff} for the estimated likelihood with bootstrap particle filter (BPF) and auxiliary particle filter (APF) averaged over 50 time series. N_{eff} is computed as in Equation (22). Variance and bias are computed as in Equations (93) and (94).

	Variance		Bias		N_{eff}	
	BPF	APF	BPF	APF	BPF	APF
$N = 100$	778.46	3.36	19.95	−0.68	64.07	23.52
$N = 1000$	805.79	0.20	20.32	−0.07	640.23	224.65
$N = 10,000$	808.40	0.02	20.36	−0.01	6402.11	2233.59

Table 2. Variance of the estimated likelihood for 10 different data-generating processes (TS). We consider different settings for the number of particles N .

TS	Bootstrap Particle Filter			Auxiliary Particle Filter			True Parameters		
	$N = 100$	$N = 1000$	$N = 10,000$	$N = 100$	$N = 1000$	$N = 10,000$	β	ϕ	σ_η
1	511.26	556.01	559.85	2.436	0.251	0.017	0.36	0.96	0.13
2	128.50	138.05	139.19	0.688	0.040	0.005	1.18	0.95	0.08
3	18,143.0	18,883.0	18,945.0	259.3	9.669	0.993	1.93	0.89	0.43
4	79.085	84.045	83.510	0.517	0.045	0.005	0.9	0.96	0.07
5	51.753	52.112	52.078	0.066	0.005	0.001	0.93	0.76	0.08
6	4.2531	4.2704	4.2961	0.026	0.003	0.000	1.43	0.79	0.04
7	2684.2	2780.2	2790.5	5.076	0.339	0.032	1.58	0.91	0.27
8	137.22	140.65	140.48	0.210	0.018	0.002	0.09	0.85	0.11
9	8.5151	13.541	13.419	0.808	0.067	0.008	0.94	0.98	0.10
10	3601.2	3665.6	3670.4	25.32	2.614	0.291	0.55	0.84	0.21

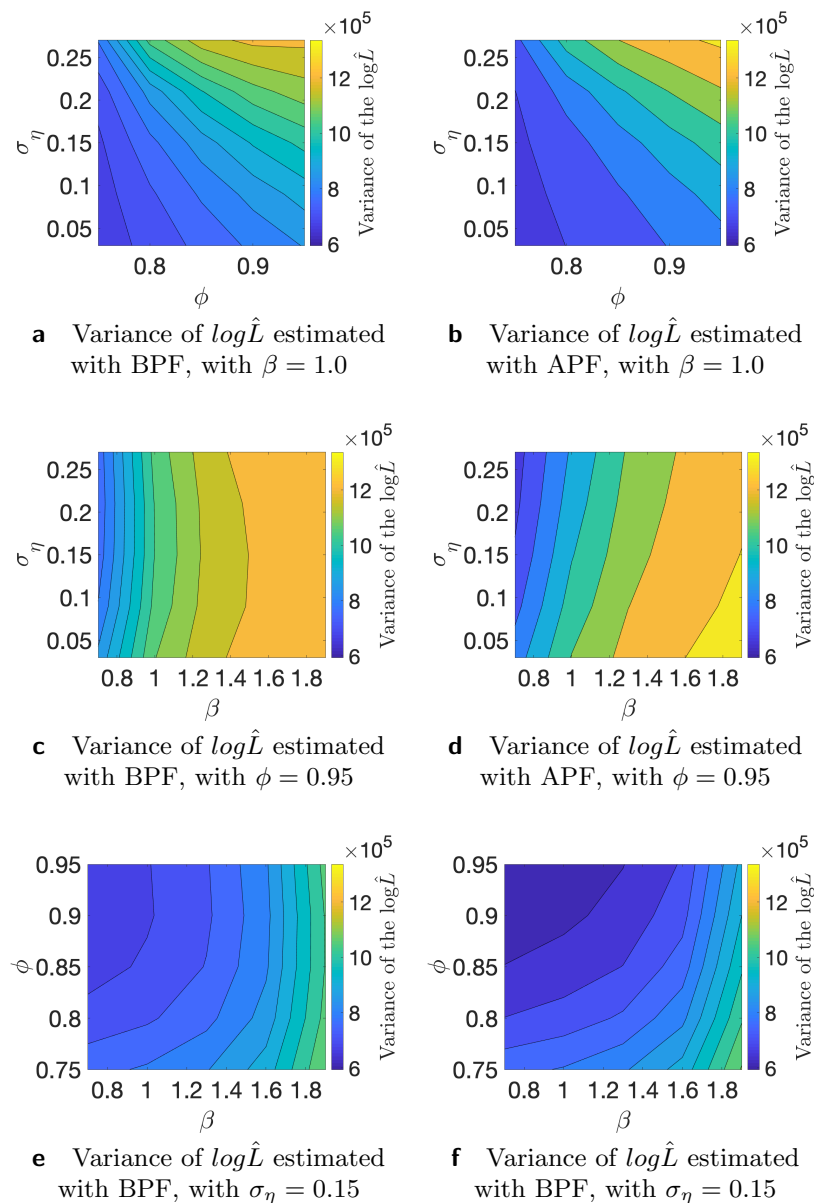


Figure 2. Variance of the estimated likelihood in different points of the parameter space for $TS = 2$ from Table 2.

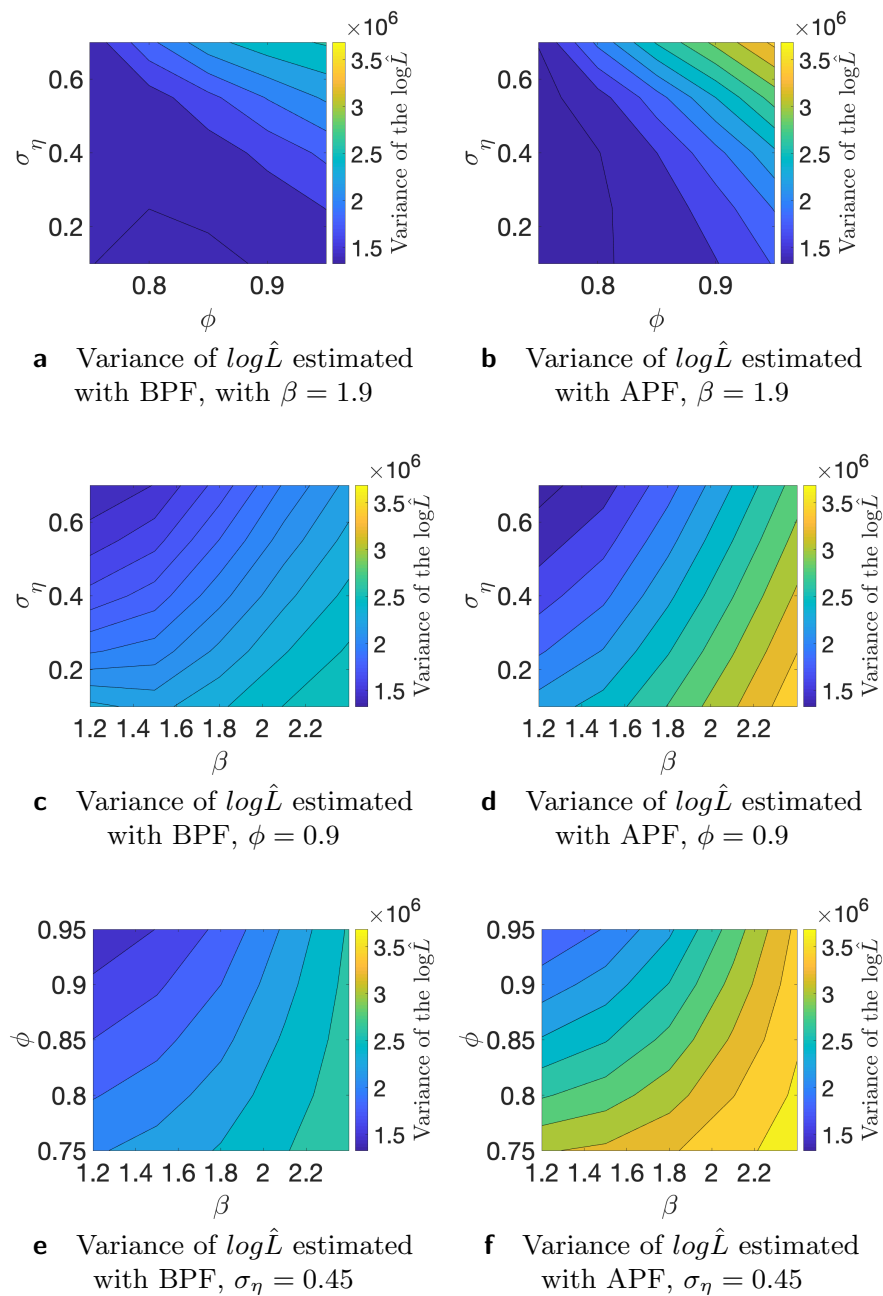


Figure 3. Variance of the estimated likelihood in different points of the parameter space for $TS = 3$ from Table 2.

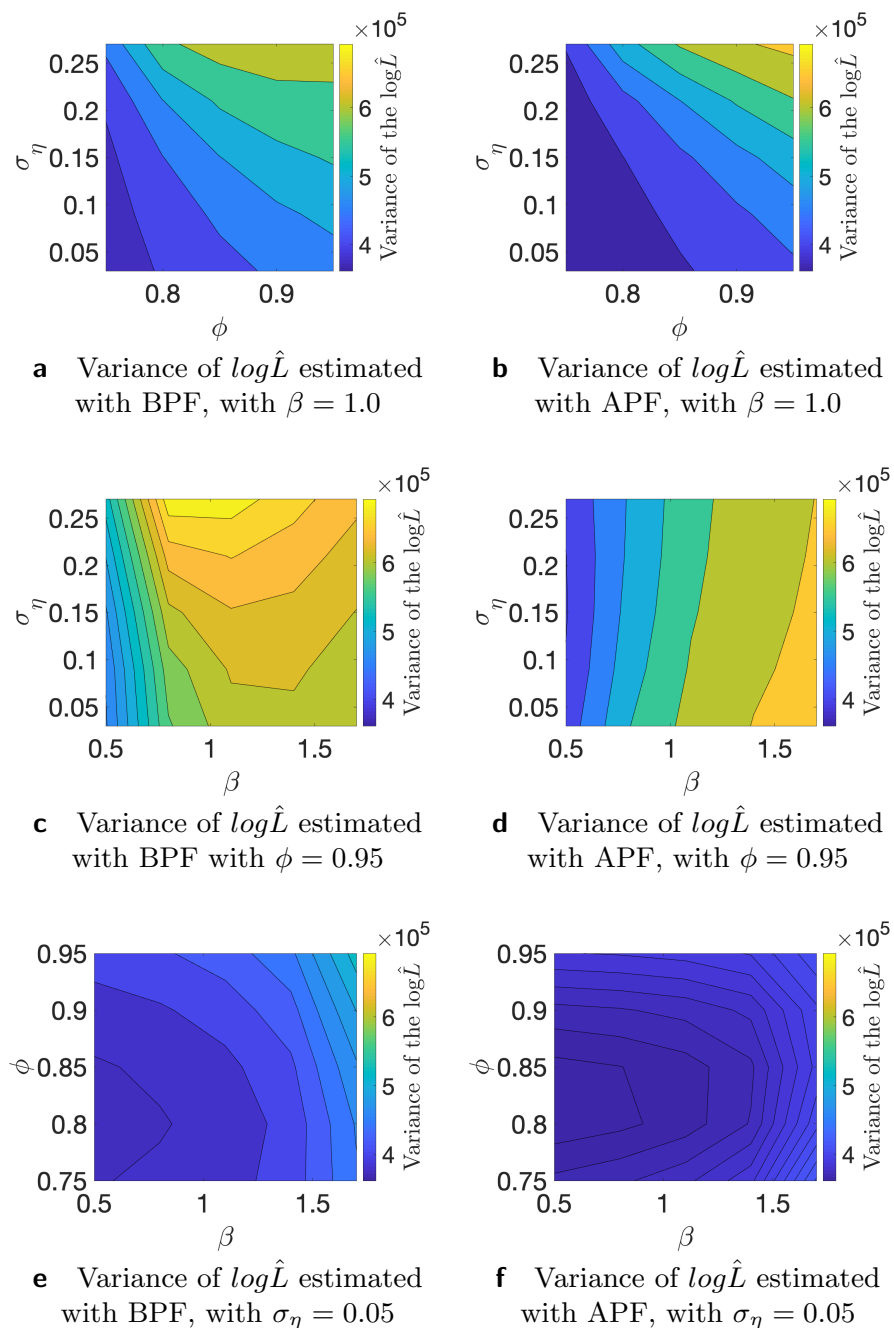


Figure 4. Variance of the estimated likelihood in different points of the parameter space for $TS = 4$ from Table 2.

4.2. Particle Metropolis–Hastings and Riemann Manifold Hamiltonian Monte Carlo

In this section, we compare particle Metropolis–Hastings (PMH) and Riemann Manifold Hamiltonian Monte Carlo. We evaluate the algorithms based on how well they recover the true parameters of the model β , ϕ , and σ_η and on the basis of the effective sample size. We obtained 20,000 samples and discarded the first 1000 as burn-in. Further, Figure 5 and Figures A1–A4 show results for both samplers: trace plots, histograms, and autocorrelation function are depicted. Table 3 presents the moments and highest posterior density intervals for the parameters of the model. The marginal likelihood in PMH was estimated with auxiliary particle filter as discussed in [23]. The Metropolis–Hastings part of the algorithm was calibrated to achieve 20–40% acceptance rate. RMHMC was implemented as in [5] with openly available implementation of the method by the authors. Both PMH and RMHMC

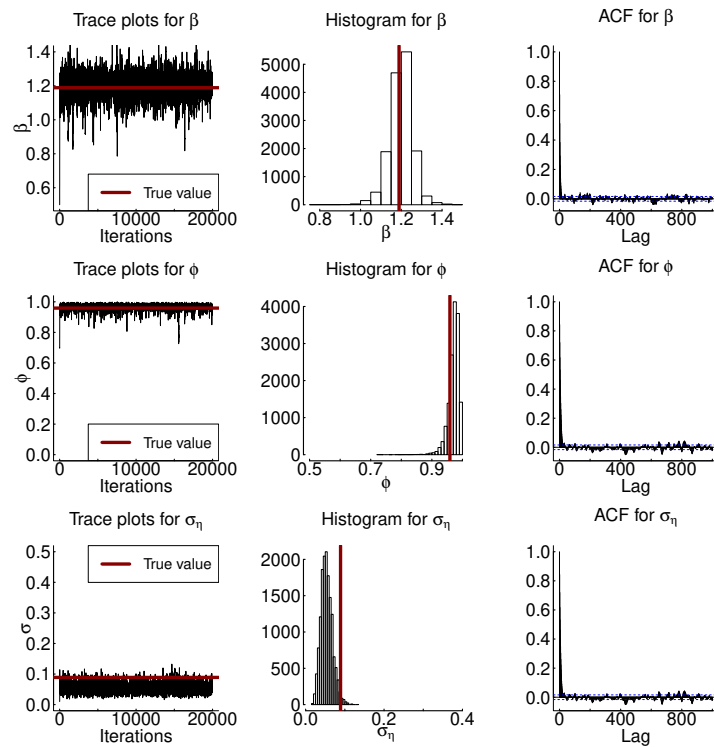
require careful calibration of the step-size, and RMHMC additionally needs calibration of the number of the leapfrog steps; thus in Table A1, we additionally present results for the no-u-turn sampler (NUTS). NUTS is an extension of HMC algorithm that allows one to tune the algorithm automatically. From Figure 5 and Figures A1–A4, we observe that autocorrelation of the samples indicated in the third column of the plots decreases faster for PMH than for RMHMC, in particular for the parameters ϕ and σ_η . Effective sample size is similarly high for parameter β for both samplers. Effective sample size for the parameters ϕ and σ_η is lower in the case of RMHMC. However, if we compute the ESS per second as presented in the last column of Table 3, this advantage disappears. This result is not surprising since PMH is the most computationally intensive procedure we are considering. Both the likelihood and the posterior use sequential sampling methods, which makes computations very demanding. Nevertheless, PMH allows us to recover the underlying parameters more accurately. In particular, in most of the presented examples, true variance of the latent volatility process lies inside the 95% highest posterior density interval for PMH. RMHMC tends to overestimate this parameter. As Table A1 indicates, the highest posterior density intervals obtained with NUTS are larger than those obtained with PMH and RMHMC. Number of gradient evaluations for RMHMC are 69718, 70042, 69802, 69801, and 70041 for Experiments 1–5, respectively.

Table 3. Posterior moments for the samples obtained with particle Metropolis–Hastings (PMH) and Riemann Manifold Hamiltonian Monte Carlo (RMHMC) for the parameters β , ϕ and σ_η of the stochastic volatility model. Experiments 1, 2, 3, 4, and 5 correspond to TS 2, 4, 5, 9, and 10 from Table 2.

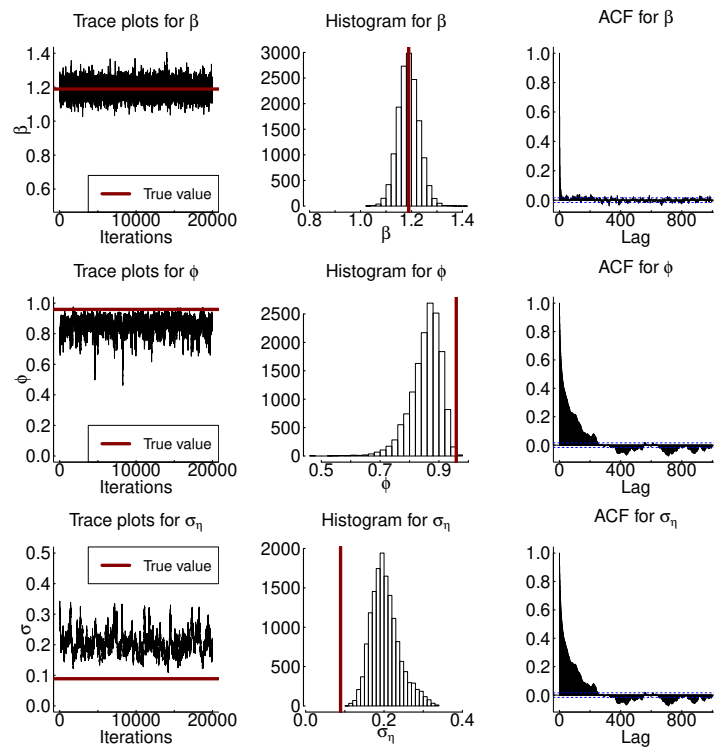
Experiment 1: Posterior Moments Obtained with PMH							
	Mean	Mode	95% HPD _l	95% HPD _u	True	ESS	ESS/s
β	1.1985	1.2140	1.0771	1.3034	1.189	2783.2	0.054
ϕ	0.9713	0.9732	0.9392	0.9987	0.959	1516.8	0.029
σ_η	0.0537	0.0514	0.0260	0.0828	0.089	2280.7	0.044
Experiment 1: Posterior Moments Obtained with RMHMC							
	Mean	Mode	95% HPD _l	95% HPD _u	True	ESS	ESS/s
β	1.18628	1.16608	1.10609	1.26526	1.189	3580.7	50.57
ϕ	0.83824	0.78390	0.71272	0.93643	0.959	188.6244	2.66
σ_η	0.21889	0.17203	0.14846	0.30072	0.0828	93.5265	1.32
Experiment 2: Posterior Moments Obtained with PMH							
	Mean	Mode	95% HPD _l	95% HPD _u	True	ESS	ESS/s
β	0.8629	0.8780	0.7870	0.9385	0.903	2909.6	0.045
ϕ	0.9644	0.9843	0.9077	0.9987	0.967	1157.2	0.017
σ_η	0.0534	0.0504	0.0233	0.0912	0.076	1694.3	0.026
Experiment 2: Posterior Moments Obtained with RMHMC							
	Mean	Mode	95% HPD _l	95% HPD _u	True	ESS	ESS/s
β	0.85071	0.82888	0.79927	0.90418	0.903	4108.8	76.26
ϕ	0.79293	0.73992	0.62454	0.92638	0.967	186.8240	3.46
σ_η	0.22291	0.23163	0.14751	0.30825	0.076	72.6236	1.34

Table 3. Cont.

Experiment 3: Posterior Moments Obtained with PMH							
	Mean	Mode	95% HPD_l	95% HPD_u	True	ESS	ESS/s
β	0.9579	0.9623	0.9090	1.0043	0.938	3330.7	0.071
ϕ	0.9079	0.9583	0.7974	0.9885	0.764	2178.9	0.046
σ_η	0.0429	0.0317	0.0184	0.0725	0.08	2690.1	0.057
Experiment 3: Posterior Moments Obtained with RMHMC							
	Mean	Mode	95% HPD_l	95% HPD_u	True	ESS	ESS/s
β	0.94395	0.90574	0.89572	0.99246	0.938	8112.2	164.83
ϕ	0.67333	0.43245	0.42649	0.88097	0.764	116.9159	2.37
σ_η	0.19759	0.18142	0.13385	0.27098	0.08	98.9095	2.0
Experiment 4: Posterior Moments Obtained with PMH							
	Mean	Mode	95% HPD_l	95% HPD_u	True	ESS	ESS/s
β	0.8865	0.8932	0.7297	1.0485	0.942	3046.7	0.057
ϕ	0.9826	0.9892	0.9672	0.9961	0.981	1781.9	0.033
σ_η	0.1100	0.1011	0.0702	0.1503	0.1	1609.9	0.031
Experiment 4: Posterior Moments Obtained with RMHMC							
	Mean	Mode	95% HPD_l	95% HPD_u	True	ESS	ESS/s
β	0.90560	0.80476	0.74840	1.07499	0.942	445.7	9.22
ϕ	0.96391	0.96149	0.93865	0.98672	0.981	336.5	6.96
σ_η	0.17588	0.15766	0.13452	0.22005	0.1	130.0	2.69
Experiment 5: Posterior Moments Obtained with PMH							
	Mean	Mode	95% HPD_l	95% HPD_u	True	ESS	ESS/s
β	0.5836	0.5893	0.5359	0.6293	0.553	2733.8	0.0552
ϕ	0.9497	0.9970	0.8890	0.9970	0.840	2733.8	0.0552
σ_η	0.0756	0.0525	0.0294	0.1331	0.215	1589.1	0.032
Experiment 5: Posterior Moments Obtained with RMHMC							
	Mean	Mode	95% HPD_l	95% HPD_u	True	ESS	ESS/s
β	0.56805	0.57215	0.53372	0.60258	0.553	3276.6	75.41
ϕ	0.77895	0.79364	0.60134	0.92411	0.840	130.7	3.0
σ_η	0.22905	0.21974	0.14799	0.31755	0.215	64.9	1.49



a Trace plots (left), histograms (middle) and ACF plots (right) obtained with Particle Metropolis-Hastings



b Trace plots (left), histograms (middle) and ACF plots (right) obtained with Riemann manifold Hamiltonian Monte Carlo

Figure 5. Results of the sampling from the posterior distribution with PMH and RMHMC for $TS = 2$ from Table 2. The first column corresponds to the trace plots, the middle column to histograms obtained with the samples from the posterior distribution, and the last column corresponds to autocorrelation function for the samples.

4.3. Integrated Nested Laplace Approximation

We provide two simulation studies for the integrated nested Laplace approximation. First, we replicate and extend the simulation study provided in [38] by analyzing data-generating processes with different values of μ and σ_η since the estimation of the variance parameter appears to be a challenge for existing methods. Table 4 replicates results from [38] with the parametrization of the model with the scale parameter μ , and Table 5 presents results for the parametrization with the scale parameter β and different data-generating processes. Both Monte Carlo studies are conducted with 1000 iterations. Our findings are comparable to those of [38]: the mean of the volatility process and the persistence parameter are estimated quite accurately, while the variance of the latent volatility process estimated with INLA is biased—usually, it is overestimated. Second, we provide the posterior moments for INLA similarly to Table 3 for PMH and RMHMC. These results are presented in Table 6. These results also suggest that the variance of the latent volatility process tends to be overestimated with INLA to a larger degree than with RMHMC, which also overestimates this parameter, as can be seen from the Table 3. Moreover, highest posterior density intervals for the parameter ϕ obtained with INLA are larger than those obtained with PMH and RMHMC.

Table 4. Bias and square root of the mean squared error for integrated nested Laplace approximation (INLA) parametrized with scale parameter μ .

μ_{true}	ϕ_{true}	$\sigma_{\eta true}$	bias (μ)	smse (μ)	bias (ϕ)	smse (ϕ)	bias (σ_η)	smse (σ_η)
0.1366	0.9	0.0186	−0.0672	0.01804	−0.6138	0.5403	0.29183	0.0912
−0.2143	0.9	0.0366	−0.0565	0.0092	−0.5803	0.4953	0.2739	0.0815
−0.0658	0.9	0.0636	−0.0664	0.0151	−0.5817	0.5176	0.2494	0.0704
−0.0289	0.95	0.0186	−0.0675	0.0151	−0.6354	0.5550	0.2883	0.0900
0.0203	0.95	0.0366	−0.0705	0.0207	−0.5796	0.4843	0.2677	0.0795
−0.0630	0.95	0.0636	−0.0931	0.0333	−0.4142	0.3250	0.2129	0.0592
−0.0174	0.98	0.0186	−0.0763	0.0222	−0.6173	0.5458	0.2788	0.0874
0.1343	0.98	0.0366	−0.1534	0.0797	−0.4329	0.3650	0.2228	0.0654
0.0584	0.98	0.0636	−0.2596	0.1379	−0.2095	0.1710	0.1280	0.034

Table 5. Bias and square root of the mean squared error for INLA parametrized with scale parameter β .

β_{true}	ϕ_{true}	$\sigma_{\eta true}$	bias (μ)	smse (μ)	bias (ϕ)	smse (ϕ)	bias (σ_η)	smse (σ_η)
0.367	0.965	0.134	−0.2295	0.0526	−0.0083	0.0001	0.4725	0.2233
1.188	0.959	0.088	0.2000	0.0400	−0.1909	0.0364	0.3466	0.1201
1.937	0.897	0.433	1.6162	2.6122	−0.0021	0.0000	0.5926	0.3512
0.902	0.966	0.075	−0.1888	0.0356	−0.3049	0.0930	0.3153	0.0994
0.938	0.764	0.080	−0.0515	0.0026	−0.4921	0.2421	0.1468	0.0215
1.435	0.793	0.048	0.6014	0.3616	−0.3872	0.1499	0.2155	0.0464
1.588	0.919	0.275	0.2539	0.0644	0.0202	0.0004	0.4786	0.2291
0.092	0.857	0.109	−0.0841	0.0070	−0.7490	0.5610	0.1341	0.0179
0.942	0.980	0.100	−0.3246	0.1054	0.0192	0.0003	95.8384	9185. 0
0.553	0.840	0.214	−0.2384	0.0568	−0.3904	0.1524	0.1918	0.0368

Table 6. Posterior results for estimation of the stochastic volatility (SV) model with INLA. Experiments 1, 2, 3, 4, and 5 correspond to TS 2, 4, 5, 9, and 10 from Table 2.

Experiment 1					
	mean	mode	95% HPD_l	95% HPD_u	true
β	1.1786	1.1753	1.0937	1.2797	1.189
ϕ	0.7717	0.9986	0.2029	0.9945	0.959
σ_η	0.4368	0.4918	0.3048	0.6665	0.089
Experiment 2					
	mean	mode	95% HPD_l	95% HPD_u	true
β	0.8457	0.8426	0.7807	0.9367	0.903
ϕ	0.7480	0.9993	−0.0844	0.9996	0.967
σ_η	0.4155	0.9111	0.2631	0.7726	0.076
Experiment 3					
	mean	mode	95% HPD_l	95% HPD_u	true
β	0.9446	0.9403	0.8672	1.0798	0.938
ϕ	0.3827	0.9999	−0.7491	0.9999	0.764
σ_η	0.2328	4.5772	0.1327	0.5202	0.080
Experiment 4					
	mean	mode	95% HPD_l	95% HPD_u	true
β	0.0892	0.0893	0.0847	0.0939	0.942
ϕ	0.1266	1.0000	−0.9470	0.9994	0.981
σ_η	0.2405	0.3196	0.1407	0.4447	0.100
Experiment 5					
	mean	mode	95% HPD_l	95% HPD_u	true
β	0.5614	0.5604	0.5286	0.5993	0.553
ϕ	0.4567	0.9980	−0.3055	0.9726	0.840
σ_η	0.4067	0.4538	0.2888	0.5680	0.215

4.4. Fixed-Form Variational Bayes

In this section, we discuss results for the simulation study with fixed-form variational Bayes. We consider the same time series as in the case of comparison between PMH and RMHMC. In Table 7, we present estimated variational parameters and in Figure 6, comparison of the posterior with fixed-form variational Bayes (in blue), RMHMC (histograms from the posterior samples), and INLA (green). It is clear that in some cases, the variational Bayes method performs quite well; in particular, parameter β is very well estimated in most of the cases. Only in Figure 6j is the approximate posterior for β far from the truth. The variance parameter is underestimated in all cases with VB; this is less severe in the cases when the true variance is relatively small. However, when the true variance is relatively large, the discrepancy between VB estimate and the true value increases, as can be seen from Figure 6o. We observe the opposite picture with INLA: it tends to overestimate the variance of the latent volatility process. Overestimation of the variance of the latent volatility process for stochastic volatility models with INLA has been previously reported in [38]. Additionally, it is reported in [38] that this effect decreases with larger values of σ_η . The source of this has to be investigated further. RMHMC overestimates the variance to a lesser degree than INLA, and as can be seen from Figure 6o, this is also connected to the value of the ground truth for σ_η : with larger true value of σ_η , RMHMC provides more accurate results.

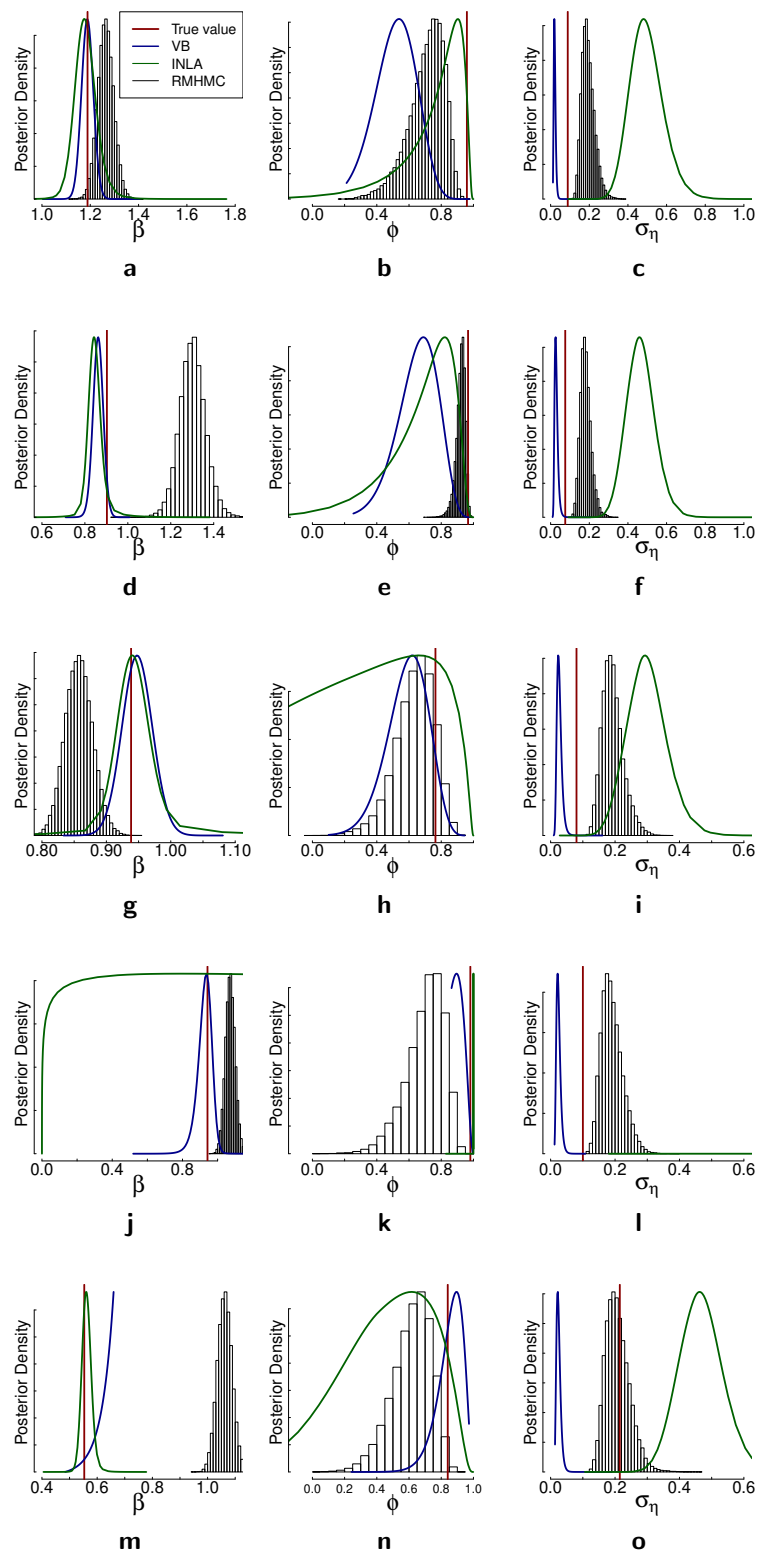


Figure 6. Illustration of the Fixed-form Variational Bayes in comparison to RMHMC and INLA. Subfigures illustrate the posterior distributions estimated with different methods for the different data-generating processes. (a–c) correspond to Experiment 1 from Tables 2 and 6, (d–f) correspond to Experiment 2, (g–i) correspond to Experiment 3, (j–l) correspond to experiment 4, and (m–o) correspond to Experiment 5. Red vertical lines indicate true parameter values.

Table 7. Parameters of the posterior distribution obtained with fixed-form Variational Bayes. Exp. 1–5 correspond to the Experiments 1–5 in Tables 3 and 6.

Exp.	ζ_1	ζ_2	ζ_3	ζ_4	ζ_5
1	31.1347	9.3573	19.1852	0.4658	−0.2051
2	28.2720	5.2039	12.6658	0.4510	−0.1812
3	30.3353	7.1038	13.9647	0.4563	−0.2039
4	38.8151	2.1501	17.9035	0.7568	−0.4296
5	38.8151	2.1501	17.9035	0.7568	−0.4296

4.5. Comparison of the Methods on the Real Data

In this section, we present posterior distributions of the parameters estimated with different Bayesian inference methods on two real-world time series. First, we consider the mean corrected log-returns of the Australian dollar against the US dollar. The data range from January 1994 to December 2003 with a total of 519 weekly observations. Resulting posterior distributions obtained with different inference methods are presented in Figure 7. Second, we consider daily log-returns for the DAX index from 3 January 2000 until 17 May 2001, which in total constitute 1000 observations. We provide descriptive statistics for both time series in Table A2. Resulting posterior distributions for this time series are presented in Figure 8. The main discrepancies between the methods are largest in the estimation of the parameter σ_η for both time series, and the results are consistent with the simulation studies in terms of the difference of these discrepancies. As we can see from Figures 7c and 8d, the posterior distribution of σ_η obtained with variational Bayes is concentrated in smaller values in comparison to the other methods. INLA suggests the higher values for σ_η in comparison to the other methods. The posterior samples obtained with RMHMC are concentrated in values higher than the ones obtained with PMH. Both sampling methods appear to give results larger than VB but smaller than INLA for the parameter σ_η .

In Table 8, we present results for efficient sample size (ESS) for both empirical applications and both samplers. Similarly to what is found in simulation studies, ESS is higher in the case of the PMH algorithm. However, if the computational time were taken into account, this advantage would have disappeared, similarly to the results in Table 2.

Table 8. Efficient sample size (ESS) for PMH and RMHMC in real-world time series applications: weekly log-returns for the exchange rate of Australian/US dollars and daily log-returns of DAX index.

Australian/US Dollars Exchange Rate		
	ESS PMH	ESS RMHMC
β	3373.9	906.2
ϕ	1546.4	481
σ_η	2439	208.3
DAX Index		
	ESS PMH	ESS RMHMC
β	3868.4	3962.3
ϕ	915.3	134.6
σ_η	2439	79.6

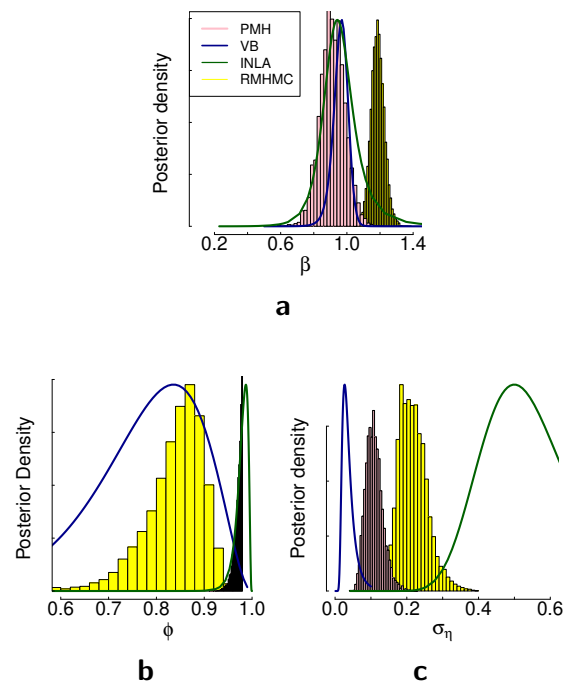


Figure 7. Comparison of PMH (pink), VB (blue), INLA (green), and RMHMC (yellow) on the weekly log-returns of the Australian dollar against the US dollar. Subfigures illustrate the posterior distributions for different parameters of the model obtained with different methods. (a) Corresponds to the posterior distribution for the parameter β . (b) Corresponds to the posterior distribution of the parameter ϕ . (c) Corresponds to the posterior distribution of the parameter σ_η .

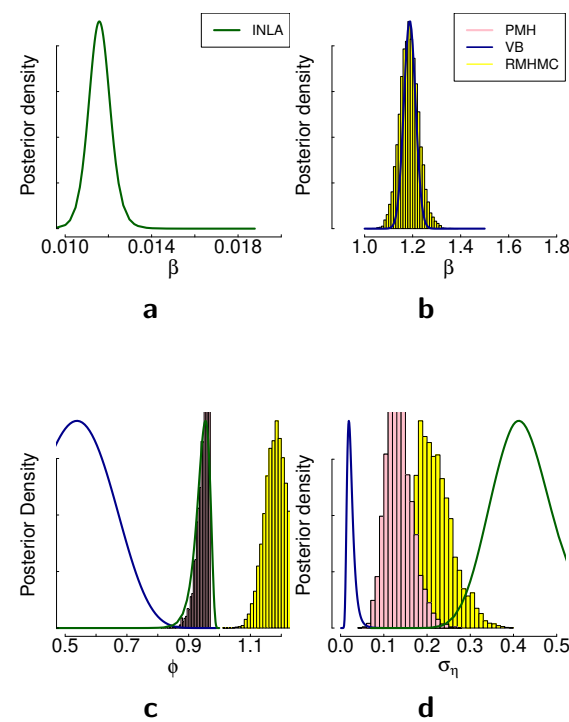


Figure 8. Comparison of PMH (pink), VB (blue), INLA (green), and RMHMC (yellow) on the daily log-returns of DAX index. Subfigures illustrate the posterior distributions for different parameters of the model obtained with different methods. (a,b) correspond to the posterior distribution of the parameter β . (c) corresponds to the posterior distribution of the parameter ϕ . (d) corresponds to the posterior distribution of the parameter σ_η .

5. Discussion

This paper reviewed multiple methods for the estimation of nonlinear state-space modes and stochastic volatility modes in particular that appear in Bayesian statistics and machine learning. We in particular focused on representative inference methods from different classes: methods that can recover the posterior distribution ‘exactly’ and the ones that build an approximation. We discussed which methods have the potential to be applied in a multivariate or high-dimensional situation and why they have this potential. Finally, we discovered that while stochastic volatility models are common for use in simulation studies for demonstrating the performance of the methods, usually not enough possible data-generating processes are considered to make a fair comparison. In particular, the performance of the methods is heavily connected to the variance of the latent volatility process.

State-space models can be powerful tools for modeling latent variables in different scientific fields. However, already for univariate time series, they are challenging to estimate. This paper’s main aim was to review and understand the existing classes of methods of estimation (targeting exact posterior or approximating it) and define the direction one can undertake for the estimation of *multivariate* nonlinear state-space models. The challenge arises from both statistical and computational perspectives. By this, we mean it is hard to develop methods that both provide sufficiently good results from the estimation point of view and are computationally feasible. We have reviewed a number of methods that allow a trade-off between these two aspects. In particular, we have considered particle Markov Chain Monte Carlo and reviewed multiple particle filtering approaches for this method, Riemann Manifold Langevin Hamiltonian Monte Carlo, Integrated Nested Laplace Approximation, and Variational Bayes. All these methods are equipped with the ability to estimate models with intractable likelihoods.

5.1. Sequential Monte Carlo

We compared the auxiliary particle filter with the bootstrap particle filter in terms of the variance of the estimated likelihood. We found that the auxiliary particle filter outperformed the bootstrap particle filter for most of the data-generating processes. As discussed in [27], auxiliary particle filter does not always have a smaller variance of the estimated likelihood. Additionally, we looked into how the variance of the estimated likelihood changes in the parameter space. We found that, in particular, the variance of the latent process affects the variance of the estimated likelihood. This implies that one has to find the balance for the number of particles used in Sequential Monte Carlo and a clever way of finding initial parameter values for the sampling from the posterior, especially when considering multivariate models. The advantage of the auxiliary particle filter from the methodological point of view is that it takes into account current observation y_t when constructing the proposal for the particles $q(h_t | h_{t-1}, y_t)$. The method that we did not include in our simulation study, but that possibly can solve the problem with the variance of the estimated likelihood, is the iterated auxiliary particle filter (iAPF): for the proposal of the particles, it uses not only current observation y_t , but all observations $q(h_t | h_{t-1}, y_{1:T})$. A backward sequential procedure with an optimization step is used in this proposal mechanism for the particles, which makes the algorithm computationally intensive. The multivariate application of the stochastic volatility model in [26] considers only diagonal case of the matrix Φ , and the proposed procedure for the particle proposals does not incorporate such dependence. While this method does introduce an additional computational burden on already computationally intensive method (particle Metropolis–Hastings), it is promising for getting state-of-the-art results for the task of parameter estimation.

5.2. Particle Metropolis-Hastings

Metropolis–Hastings is a general MCMC method that is easy to implement and works well for the univariate model. The estimation results are satisfying when it is properly calibrated, and good mixing of the chains is achieved. It works well in low-dimensional

problems but is unlikely to be successful in the case of multivariate stochastic volatility models. Considering the non-diagonal matrix Φ in a five-dimensional case, we would have 45 parameters to be estimated. The random walk proposal would be very inefficient even with a reasonable sparseness assumption on Φ . Nevertheless, in the low-dimensional model, we get the best estimation results with particle Metropolis–Hastings, where the particle filtering scheme is chosen to be an auxiliary particle filter. From the methods considered in this paper, particle Markov Chain Monte Carlo methods are easiest to adapt to different specifications of the model and are easiest to implement.

5.3. Riemann Manifold Hamiltonian Monte Carlo Methods

Hamiltonian Monte Carlo is a very attractive method for high-dimensional problems as it allows us to explore the parameter space efficiently. In particular, the gain in efficiency comes from avoiding random walk behavior in the proposals. The disadvantage comes from the need of careful calibration since there is no principled way of choosing matrix M . RMHMC avoids this problem by exploiting underlying geometry in the proposal mechanism. In our study, we notice that RMHMC results in good mixing of the Markov chains, and the method is generally easy to calibrate, but the estimation of the parameters is not very good. In particular, it appears that the variance of the latent volatility process is challenging for the method. It is not surprising that the PMH algorithm performs better in terms of parameter estimation since we use an auxiliary particle filter for the volatility process estimation and thus take current observation y_t for the particle proposals. RMHMC does not benefit from similar information when estimating model parameters. Therefore, improved estimation of the volatility process can be one of the directions for improving the performance of RMHMC for the parameter estimation of stochastic volatility models.

5.4. Variational Bayes

As one can see from the illustrative example, in some cases, variational Bayes performs quite well; however, there are also situations when it is far off from the underlying truth. The challenge with stochastic volatility models remains the same: it is difficult to estimate the latent states. In the approach of [6], this is done via Kalman filtering. Therefore, the drawback of linearization of the model will remain and will show in the final results. In this respect, the possible combination of VB and SMC can be of interest. Some advances in this direction have already been made [42].

5.5. Integrated Nested Laplace Approximation

Integrated Nested Laplace Approximation is another approach that works well considering how fast the method is, but it clearly overestimates the variance of the latent volatility process. Additionally, the sparse matrix computation that is used in univariate models is not applicable to the multivariate case. In the multivariate case, the precision matrix in Equation (67) is not sparse, and thus, the method does not benefit from fast sparse matrix computation. An approach that we have not considered in this paper is the Expectation Propagation algorithm. In particular, the authors of [43] propose a way to improve approximate marginals $p(x_i | \theta, y)$ in latent Gaussian fields by using EP. The motivation of the approach builds on the fact that EP can give better approximations than the Laplace approximation in this case. The improvements, however, would come at computational costs. In the univariate case, the extra computational costs do not play a significant role as the algorithm can be parallelized. However, it is hard to say how big the difference would be in the multivariate model, both in terms of improvement in the estimation and loss in computational speed.

6. Conclusions

We reviewed multiple Bayesian inference methods, which both target the exact posterior distribution and approximate it. By comparing methods on various data-generating processes, we notice that variational Bayes tends to underestimate the latent volatility

process variance, while INLA and RMHMC, in the cases considered, overestimated this parameter. We also get similar disposition of the results on two real-world data sets. We achieved the best performance with PMH in terms of recovering ground truth and uncertainty quantification. In PMH, the particle filtering step was performed with an auxiliary particle filter. This indicates that filtering with look-ahead approaches, which include current (or future) observations into proposal machinery can improve the performance of the inference method. It is important to note that different data-generating processes for simulation studies would indicate different performance results. Thus, we stress that when using stochastic volatility models, more than one data-generating process should be considered for methods comparison. This practice would allow indicating in which situation a method can fail or perform differently. Our results indicate that fixed-form variational Bayes tends to underestimate the variance of the latent process, while RMHMC and INLA overestimated this parameter. To estimate the stochastic volatility model in the multivariate case, the combination of different strategies appears to be necessary. In a high-dimensional case, the random-walk proposal would become extremely inefficient. At the same time, approximate methods lose their outstanding computational advantage (for example, INLA), and the implementation of these methods in the multivariate case is not straightforward.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: The author sincerely thanks anonymous referees for constructive and useful comments. Their suggestions helped improve and clarify this manuscript.

Conflicts of Interest: The author declares no conflict of interest.

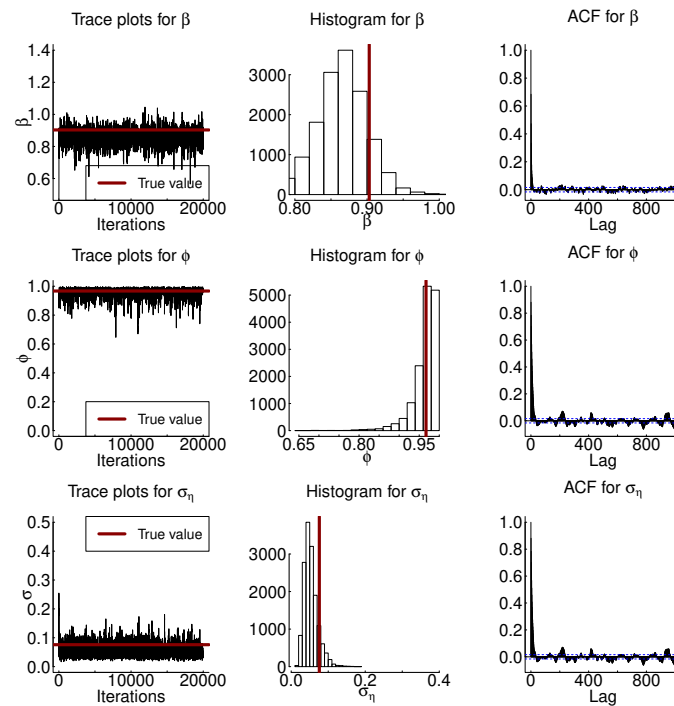
Abbreviations

The following abbreviations are used in this manuscript:

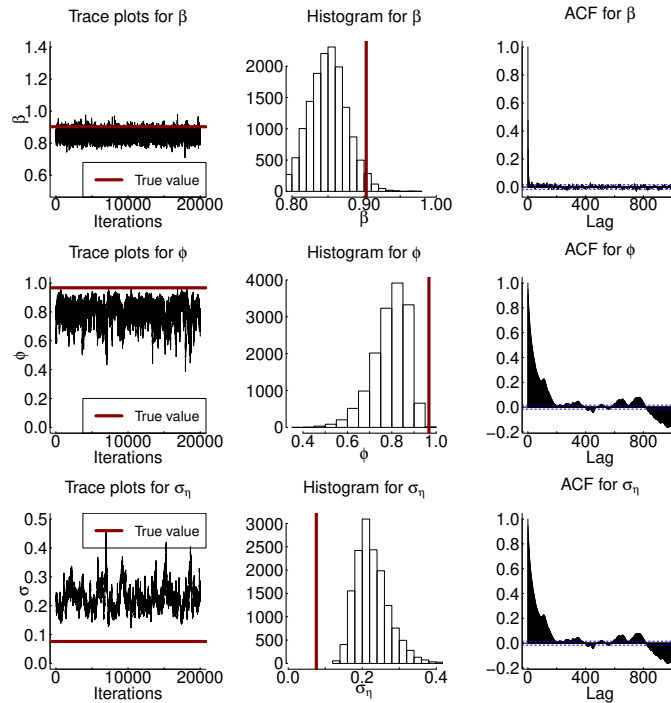
SV	Stochastic volatility
SMC	Sequential Monte Carlo
iAPF	Iterated Auxiliary Particle Filter
PMCMC	Particle Markov Chain Monte Carlo
PMH	Particle Metropolis-Hastings
HMC	Hamiltonian Monte Carlo
RMHMC	Riemann Manifold Hamiltonian Monte Carlo
VB	Variational Bayes
INLA	Integrated Nested Laplace Approximation

Appendix A

Appendix A.1.

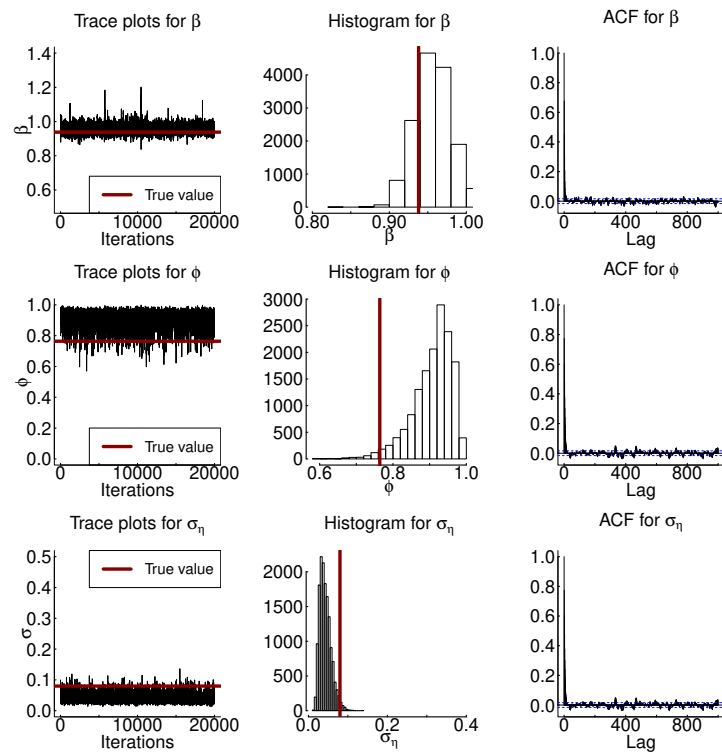


a Trace plots (left), histograms (middle) and ACF plots (right) obtained with Particle Metropolis-Hastings

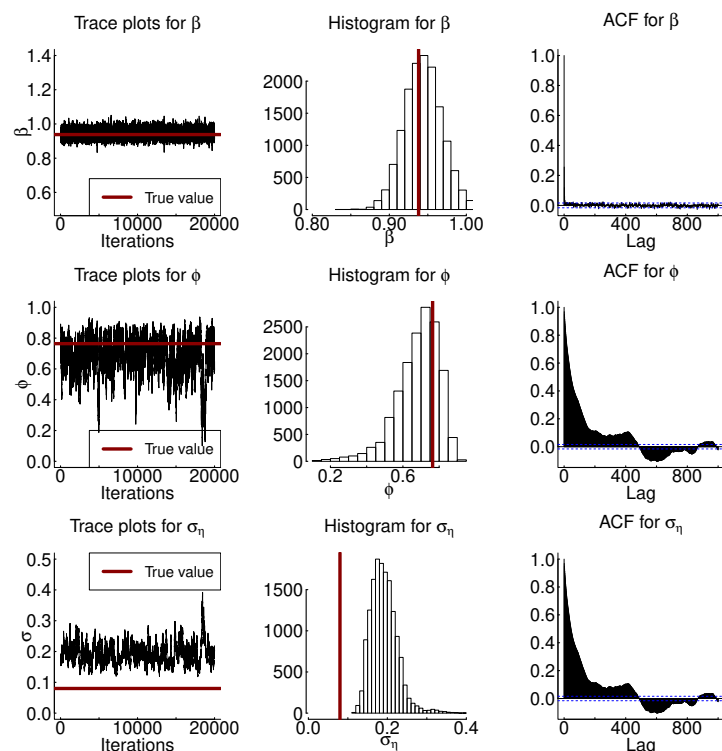


b Trace plots (left), histograms (middle) and ACF plots (right) obtained with Riemann manifold Hamiltonian Monte Carlo

Figure A1. Results of the sampling from the posterior distribution with PMH and RMHMC for $TS = 4$ from Table 2. The first column corresponds to the trace plots, the middle column to histograms obtained with the samples from the posterior distribution, and the last column to autocorrelation function for the samples.

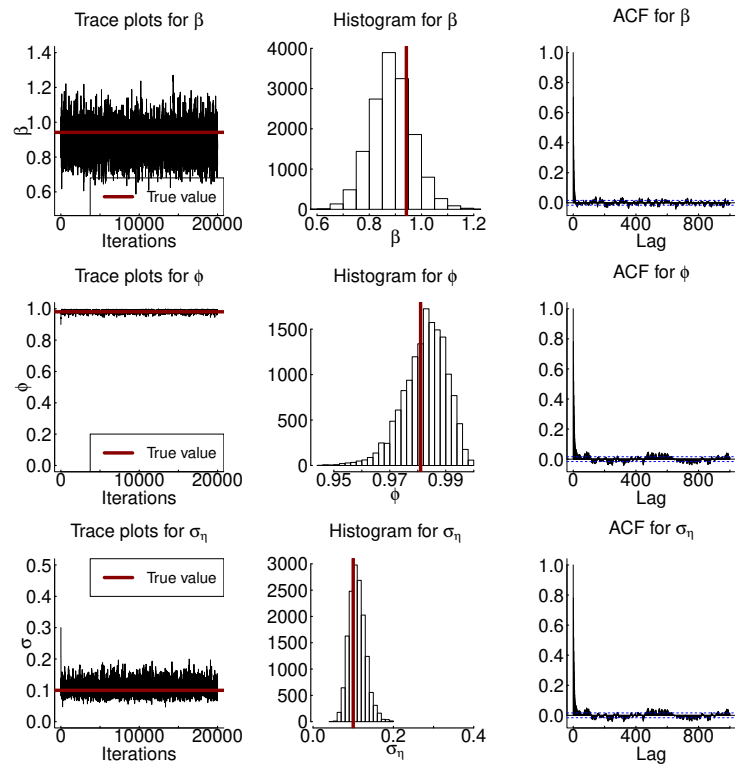


a Trace plots (left), histograms (middle) and ACF plots (right) obtained with Particle Metropolis-Hastings

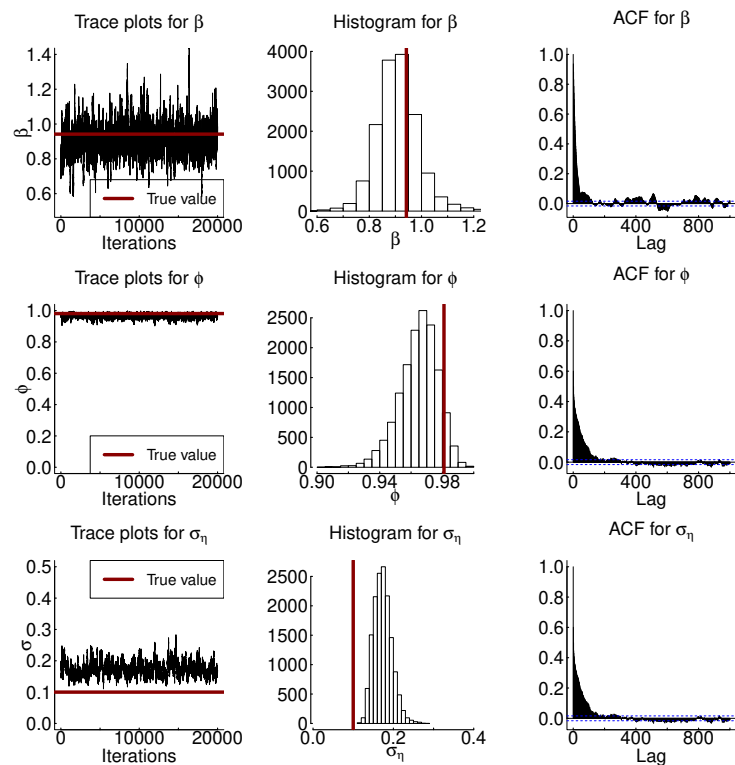


b Trace plots (left), histograms (middle) and ACF plots (right) obtained with Riemann manifold Hamiltonian Monte Carlo

Figure A2. Results of the sampling from the posterior distribution with PMH and RMHMC for $TS = 5$ from Table 2. The first column corresponds to the trace plots, the middle column to histograms obtained with the samples from the posterior distribution, and the last column to autocorrelation function for the samples.

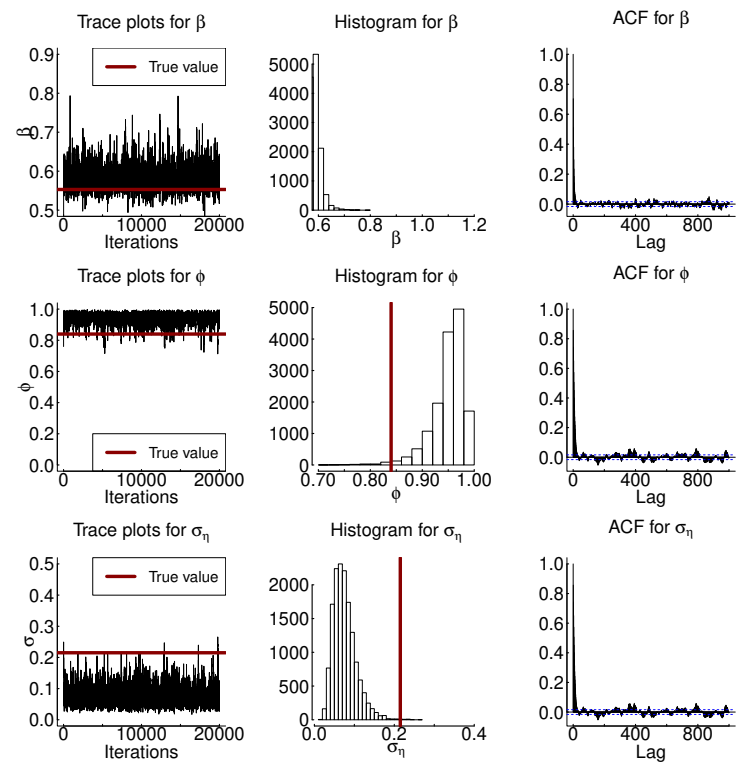


a Trace plots (left), histograms (middle) and ACF plots (right) obtained with Particle Metropolis-Hastings

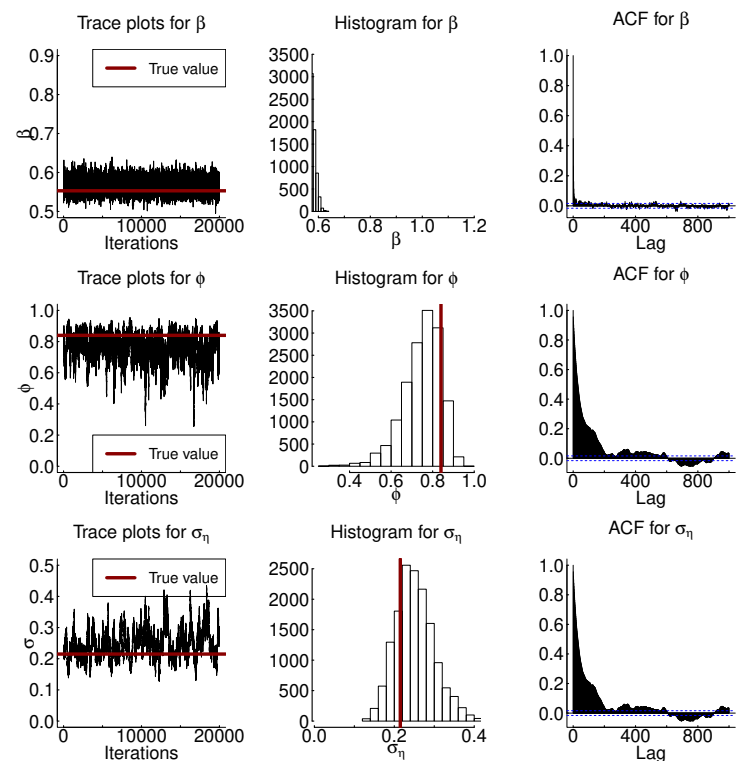


b Trace plots (left), histograms (middle) and ACF plots (right) obtained with Riemann manifold Hamiltonian Monte Carlo

Figure A3. Results of the sampling from the posterior distribution with PMH and RMHMC for $TS = 9$ from Table 2. The first column corresponds to the trace plots, the middle column to histograms obtained with the samples from the posterior distribution, and the last column to autocorrelation function for the samples.



a Trace plots, histograms and ACF plots obtained with Particle Metropolis-Hastings



b Trace plots, histograms and ACF plots obtained with Riemann manifold Hamiltonian Monte Carlo

Figure A4. Results of the sampling from the posterior distribution with PMH and RMHMC for $TS = 10$ from Table 2. The first column corresponds to the trace plots, the middle column to histograms obtained with the samples from the posterior distribution, and the last column to autocorrelation function for the samples.

Appendix A.2.

Although we do not give details for the NUTS sampler [44] in the main text, we present here experiments for this sampler using the same experiments as in the main text. The results in Table A1 are obtained with the sampler implemented in RStan [45]. The method provides large confidence intervals for the parameters ϕ and σ_η . Similarly to RMHMC, the variance of the latent volatility process tends to be overestimated based on the mean and the mode of the posterior distribution. The confidence intervals obtained with NUTS appear to be quite large, especially for the parameters ϕ and σ_η . The multivariate version of the stochastic volatility model can provide additional challenges since different parameters might require different step sizes and the sampler can get stuck in the regions of space where a small step size is needed to achieve target acceptance rate. In the univariate case, NUTS appears to be more efficient than both PMH and RMHMC as can be seen from the last two columns of Table A1. The applicability of this particular implementation can be limited due to the large 95% highest posterior intervals as uncertainty about parameters ϕ and σ_η is very large in most cases.

Table A1. Posterior results for estimation of the SV model with INLA. Experiments 1, 2, 3, 4, and 5 correspond to TS 2, 4, 5, 9, and 10 from Table 2.

Experiment 1							
	mean	mode	95% HPD _l	95% HPD _u	true	ESS	ESS/s
β	1.1785	1.1779	1.1053	1.2567	1.189	18794	87.31
ϕ	0.6266	0.8366	0.0886	0.9773	0.959	18704	86.89
σ_η	0.3117	0.2985	0.0961	0.5397	0.089	18534	86.10
Experiment 2							
	mean	mode	95% HPD _l	95% HPD _u	true	ESS	ESS/s
β	0.8498	0.8509	0.7978	0.9004	0.903	18694	42.08
ϕ	0.4033	0.7162	-0.3309	0.9912	0.967	17888	40.26
σ_η	0.2944	0.3064	0.0547	0.5127	0.076	18789	42.29
Experiment 3							
	mean	mode	95% HPD _l	95% HPD _u	true	ESS	ESS/s
β	0.9510	0.9510	0.9074	0.9960	0.938	19001	42.31
ϕ	0.0188	-0.2263	-0.8129	0.8727	0.764	18168	40.46
σ_η	0.1259	0.0276	0.0000	0.2906	0.080	19001	42.31
Experiment 4							
	mean	mode	95% HPD _l	95% HPD _u	true	ESS	ESS/s
β	0.9096	0.9042	0.7178	1.1550	0.942	18587	39.88
ϕ	0.9753	0.9784	0.9525	0.9965	0.981	18529	39.75
σ_η	0.1398	0.1317	0.0876	0.1923	0.100	18397	39.47
Experiment 5							
	mean	mode	95% HPD _l	95% HPD _u	true	ESS	ESS/s
β	0.5631	0.5632	0.5329	0.5958	0.553	18819	42.40
ϕ	0.2804	0.3784	-0.3789	0.8847	0.840	18703	42.14
σ_η	0.3648	0.3910	0.1657	0.5445	0.215	18248	41.11

Appendix A.3.

Algorithm A1 is a generic particle filter. We use auxiliary version of it proposed in [23].

Algorithm A1 Approximation of marginal likelihood with ASIR algorithm.

- 1: Draw N samples $h_0^{(i)}$ from the prior

$$h_0^{(i)} \sim \pi(h_0 | \theta), \quad i = 1, \dots, N \quad (\text{A1})$$

and set $w_0^{(i)} = 1/N$, for all $i = 1, \dots, N$.

- 2: For each $t = 1, \dots, T$ do the following
- 3: Draw samples $h_t^{(i)}$ from the importance distribution

$$h_t^{(i)} \sim q(h_t | h_{t-1}^{(i)}, y_{1:t}, \theta), \quad i = 1, \dots, N. \quad (\text{A2})$$

- 4: Compute the following weights

$$w_t^{(i)} = \frac{g(y_t | h_t^{(i)}, \theta) f(h_t^{(i)} | h_{t-1}^{(i)}, \theta)}{q(h_t^{(i)} | h_{t-1}^{(i)}, y_{1:t}, \theta)} \quad (\text{A3})$$

and compute the estimate of $p(y_t | y_{1:t-1}, \theta)$ as

$$\hat{p}(y_t | y_{1:t-1}, \theta) = \sum_i W_{t-1}^{(i)} w_t^{(i)}. \quad (\text{A4})$$

- 5: Compute normalized weights as

$$W_t^{(i)} \propto W_{t-1}^{(i)} w_t^{(i)}. \quad (\text{A5})$$

- 6: If the effective number of particles is too low, perform resampling.

Appendix A.4.

Table A2. Descriptive statistics for time series from the empirical example in Section 3.5: daily log-returns for DAX index and weekly log-returns for Australia/US dollar exchange rate.

	DAX	Australia/US
Mean	−0.001	−0.04
Std.dev.	0.013	1.00
Skewness	0.202	0.06
Kurtosis	3.253	3.37

References

1. Shephard, N.; Torben, G. Stochastic volatility: Origins and overview. In *Handbook of Financial Time Series*; Springer: Berlin/Heidelberg, Germany, 2009; pp. 233–254.
2. Platanioti, K.; McCoy, E.; Stephens, D. *A Review of Stochastic Volatility: Univariate and Multivariate Models*; Technical Report, Working Paper; Imperial College London: London, UK, 2005.
3. Asai, M.; McAleer, M.; Yu, J. Multivariate stochastic volatility: A review. *Econom. Rev.* **2006**, *25*, 145–175. [[CrossRef](#)]
4. Andrieu, C.; Doucet, A.; Holenstein, R. Particle Markov chain Monte Carlo methods. *J. R. Stat. Soc. Ser. B* **2010**, *72*, 269–342. [[CrossRef](#)]
5. Girolami, M.; Calderhead, B. Riemann Manifold Langevin and Hamiltonian Monte Carlo methods. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2011**, *73*, 123–214. [[CrossRef](#)]
6. Salimans, T.; Knowles, D.A. Fixed-form variational posterior approximation through stochastic linear regression. *Bayesian Anal.* **2013**, *8*, 837–882. [[CrossRef](#)]
7. Rue, H.; Martino, S.; Chopin, N. Approximate Bayesian inference for latent Gaussian models by using Integrated Nested Laplace Approximations. *J. R. Stat. Soc. Ser. B* **2009**, *71*, 319–392. [[CrossRef](#)]
8. Mandelbrot, B.B. The variation of certain speculative prices. In *Fractals and Scaling in Finance*; Springer: New York, NY, USA, 1997; pp. 371–418.

9. Black, F. Studies of stock price volatility changes. In *Proceedings of the 1976 Meeting of the Business and Economic Statistics Section*; American Statistical Association: Washington, DC, USA, 1976; pp. 177–181.
10. Black, F.; Scholes, M. The pricing of options and corporate liabilities. *J. Political Econ.* **1973**, *81*, 637–654. [[CrossRef](#)]
11. Hull, J.; White, A. The pricing of options on assets with stochastic volatilities. *J. Financ.* **1987**, *42*, 281–300. [[CrossRef](#)]
12. Johnson, H.; Shanno, D. Option pricing when the variance is changing. *J. Financ. Quant. Anal.* **1987**, *22*, 143–151. [[CrossRef](#)]
13. Wiggins, J.B. Option values under stochastic volatility: Theory and empirical estimates. *J. Financ. Econ.* **1987**, *19*, 351–372. [[CrossRef](#)]
14. Heston, S.L. A closed-form solution for options with stochastic volatility with applications to bond and currency options. *Rev. Financ. Stud.* **1993**, *6*, 327–343. [[CrossRef](#)]
15. Campbell, J.Y.; Champbell, J.J.; Campbell, J.W.; Lo, A.W.; Lo, A.W.; MacKinlay, A.C. *The Econometrics of Financial Markets*; Princeton University Press: Princeton, NJ, USA, 1997.
16. Harvey, A.C.; Shephard, N. Estimation of an asymmetric stochastic volatility model for asset returns. *J. Bus. Econ. Stat.* **1996**, *14*, 429–434.
17. Jacquier, E.; Polson, N.G.; Rossi, P.E. Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. *J. Econom.* **2004**, *122*, 185–212. [[CrossRef](#)]
18. Yu, J. On leverage in a stochastic volatility model. *J. Econom.* **2005**, *127*, 165–178. [[CrossRef](#)]
19. Kim, S.; Shephard, N.; Chib, S. Stochastic volatility: Likelihood inference and comparison with ARCH models. *Rev. Econ. Stud.* **1998**, *65*, 361–393. [[CrossRef](#)]
20. Gordon, N.J.; Salmond, D.J.; Smith, A.F. Novel approach to nonlinear/non-Gaussian Bayesian state estimation. *IEE Proc. F (Radar Signal Process.)* **1993**, *140*, 107–113. [[CrossRef](#)]
21. Doucet, A.; de Freitas, N.; Gordon, N. *Sequential Monte Carlo Methods in Practice*; Springer: New York, NY, USA, 2001.
22. Chopin, N. Central limit theorem for sequential Monte Carlo methods and its application to Bayesian inference. *Ann. Stat.* **2004**, *32*, 2385–2411. [[CrossRef](#)]
23. Pitt, M.K.; Shephard, N. Filtering via simulation: Auxiliary particle filters. *J. Am. Stat. Assoc.* **1999**, *94*, 590–599. [[CrossRef](#)]
24. Scharth, M.; Kohn, R. Particle efficient importance sampling. *J. Econom.* **2016**, *190*, 133–147. [[CrossRef](#)]
25. Richard, J.F.; Zhang, W. Efficient high-dimensional importance sampling. *J. Econom.* **2007**, *141*, 1385–1411. [[CrossRef](#)]
26. Guarniero, P.; Johansen, A.M.; Lee, A. The iterated auxiliary particle filter. *J. Am. Stat. Assoc.* **2017**, *112*, 1636–1647. [[CrossRef](#)]
27. Johansen, A.M.; Doucet, A. A note on auxiliary particle filters. *Stat. Probab. Lett.* **2008**, *78*, 1498–1504. [[CrossRef](#)]
28. Särkkä, S. *Bayesian Filtering and Smoothing*; Cambridge University Press: Cambridge, UK, 2013; Volume 3.
29. Roberts, G.O.; Gelman, A.; Gilks, W.R. Weak convergence and optimal scaling of random walk Metropolis algorithms. *Ann. Appl. Probab.* **1997**, *7*, 110–120. [[CrossRef](#)]
30. Robert, C.P.; Casella, G. *Monte Carlo Statistical Methods*; Springer Texts in Statistics; Springer: New York, NY, USA, 2005.
31. Duane, S.; Kennedy, A.D.; Pendleton, B.J.; Roweth, D. Hybrid Monte Carlo. *Phys. Lett. B* **1987**, *195*, 216–222. [[CrossRef](#)]
32. Neal, R.M. MCMC Using Hamiltonian Dynamics. In *Handbook of Markov Chain Monte Carlo*; CRC Press: Boca Raton, FL, USA, 2011; Volume 2.
33. Betancourt, M. A Conceptual Introduction to Hamiltonian Monte Carlo. *arXiv* **2017**, arXiv:1701.02434.
34. Roberts, G.O.; Rosenthal, J.S. Optimal scaling of discrete approximations to Langevin diffusions. *J. R. Stat. Soc. Ser. B* **1998**, *60*, 255–268. [[CrossRef](#)]
35. Gelman, A.; Lee, D.; Guo, J. Stan: A probabilistic programming language for Bayesian inference and optimization. *J. Educ. Behav. Stat.* **2015**, *40*, 530–543. [[CrossRef](#)]
36. Salvatier, J.; Wiecki, T.V.; Fonnesbeck, C. Probabilistic programming in Python using PyMC3. *PeerJ Comput. Sci.* **2016**, *2*, e55. [[CrossRef](#)]
37. Martino, S.; Aas, K.; Lindqvist, O.; Neef, L.R.; Rue, H. Estimating stochastic volatility models using integrated nested Laplace approximations. *Eur. J. Financ.* **2011**, *17*, 487–503. [[CrossRef](#)]
38. Ehlers, R.; Zevallos, M. Bayesian Estimation and Prediction of Stochastic Volatility Models via INLA. *Commun. Stat.-Simul. Comput.* **2015**, *44*, 683–693. [[CrossRef](#)]
39. Martino, S. *Approximate Bayesian Inference for Multivariate Stochastic Volatility Models*; Technical Report; Department of Mathematical Sciences, Norwegian University of Science and Technology: Trondheim, Norway, 2007.
40. Šmídl, V.; Quinn, A. *The Variational Bayes Method in Signal Processing*; Springer Science & Business Media: Dordrecht, The Netherlands, 2006.
41. Ruiz, E. Quasi-maximum likelihood estimation of stochastic volatility models. *J. Econom.* **1994**, *63*, 289–306. [[CrossRef](#)]
42. Naesseth, C.; Linderman, S.; Ranganath, R.; Blei, D. Variational Sequential Monte Carlo. In *Proceedings of the Twenty-First International Conference on Artificial Intelligence and Statistics*, Playa Blanca, Lanzarote, Canary Islands, Spain, 9–11 April 2018; pp. 968–977.
43. Cseke, B.; Heskes, T. Approximate marginals in latent Gaussian models. *J. Mach. Learn. Res.* **2011**, *12*, 417–454.
44. Hoffman, M.D.; Gelman, A. The No-U-Turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.* **2014**, *15*, 1593–1623.
45. Stan Development Team. RStan: The R Interface to Stan; R Package Version 2.21.2. Available online: <http://mc-stan.org/> (accessed on 6 April 2021).