INSTITUT FÜR HÖHERE STUDIEN
INSTITUTE FOR ADVANCED STUDIES
Vienna

CONTINUING EDUCATION CENTER

MSc Economics

Boosting MCMC Estimation of Stochastic Volatility Models

A Master's Thesis submitted for the degree of
"Master of Science"

supervised by

Leopold Sögner

Dorisz Albrecht

1527522

Vienna, 05/06/2017

INSTITUT FÜR HÖHERE STUDIEN
INSTITUTE FOR ADVANCED STUDIES
Vienna

TU WIEN CONTINUING EDUCATION CENTER

MSc Economics

Affidavit

I, Dorisz Albrecht

hereby declare

that I am the sole author of the present Master's Thesis,

Boosting MCMC Estimation of Stochastic Volatility Models

33 pages, bound, and that I have not used any source or tool other than those referenced or any other illicit aid or tool, and that I have not prior to this date submitted this Master's Thesis as an examination paper in any form in Austria or abroad.

Vienna, 05/06/2017

Signature

Contents

List of Figures

List of Tables

Abstract

This paper analyzes the arising problems of using MCMC sampling methods under different model parameterizations in a stochastic volatility model. It turns out that the performance of Bayesian inference is dependent on the true parameter values. The standard centered parameterization has shortcomings when the variability of its volatility is relatively small, while the non-centered parameterization presents with complications when the persistence parameter is close to one.

This paper uses the recently presented ancillarity-sufficiency interweaving strategy which overcomes the pitfalls of the parameterizations by using both of them in order to update the latent states and the parameters of interest jointly, this way maintaining the dependence between them.

1 Introduction

1.1 Motivation

By definition, stochastic volatility models are those in which the variance of a given stochastic process is itself randomly distributed. They are used for instance in the field of mathematical finance, as it is crucial to understand the underlying variability in markets in order to optimize risk management or investment decisions. The arising problem lies in the fact that volatility is a latent phenomenon. As volatility is usually a required statistics for the observed variable, it is inevitable to estimate its path somehow.

One way of estimation is to represent it as a random variable, emphasizing its unpredictability. The fluctuation of the value of a financial product imply a varying volatility process. We have a plethora of evidence suggesting time-varying volatility. To be convinced, it is enough to consider the bigger financial crises. If we look at Figure 1. which shows us the log returns of the S&P 500 between 2007 and 2016, we can clearly see that the volatility is time-varying.

Figure 1. displays the so called "volatility clustering", which refers to the observation of Mandelbrot (1963, p. 418.): "*large changes tend to be followed by large changes, of either sign, and small changes tend to be followed by small changes*."

Analyzing Figure 2. we see that the distribution of log returns is highly peaked and fat-tailed compared to the normal distribution (with kurtosis 9.7). In the literature, high central peak and fat-tails are usually characteristics of mixtures of distributions with different variances (Mandelbrot (1963)). This also serves as a reason why we need to consider volatility as a random variable. The aforementioned "volatility clustering" implies that volatility is correlated with past realizations of the volatility process, which is a consequence of mean-reversion.

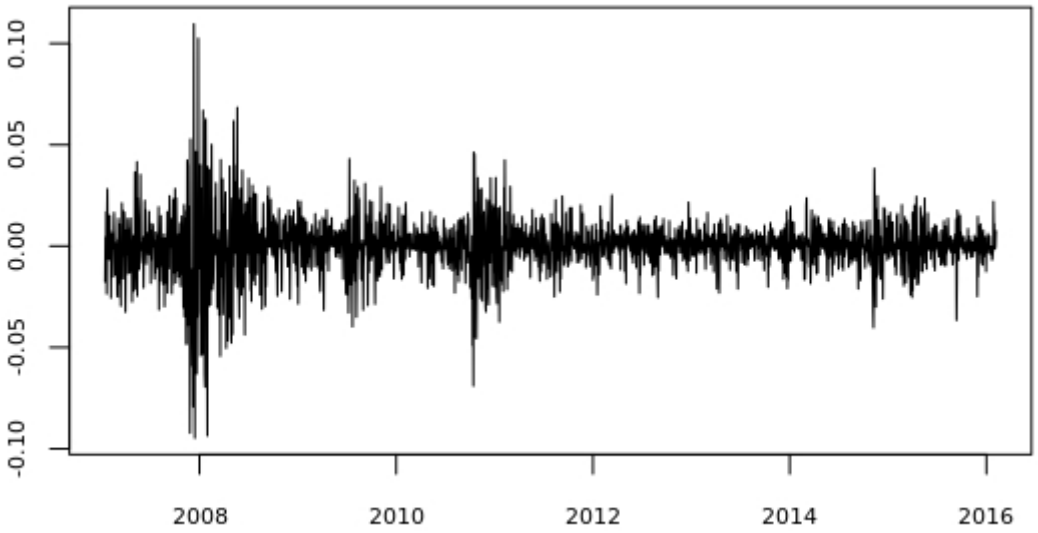


Figure 1: Log returns of S&P500

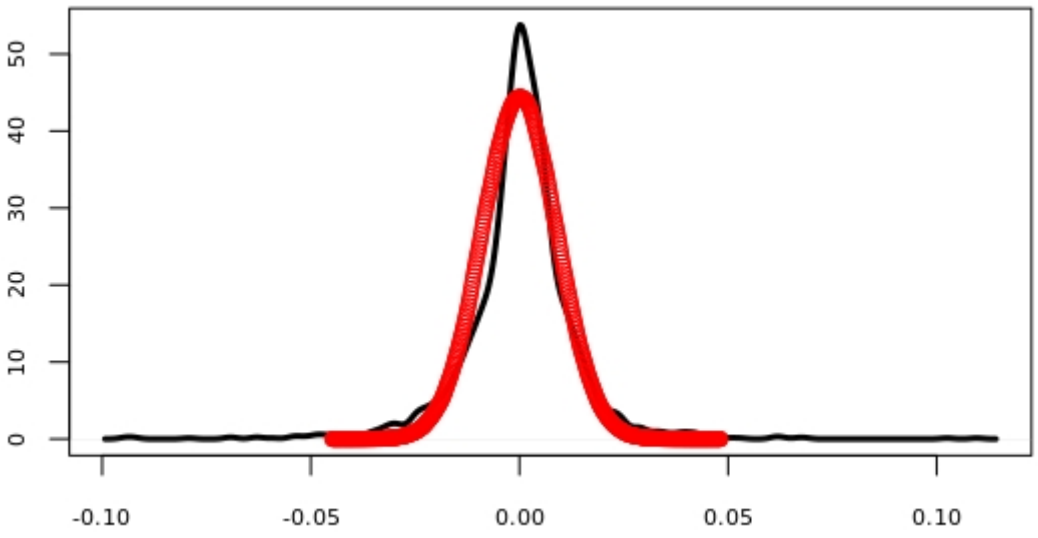


Figure 2: Normal distribution (red) and the frequency distribution of the S&P 500 log returns

1.2 Outline of the estimation process

Even after allowing volatility to vary over time, there is still no consensus over how to model volatility or how to best estimate the required parameters. My main focus will be on univariate stochastic volatility (SV) autoregressive models.
The SV model, which was first introduced by Taylor (1986), had the pioneering idea of allowing the variance of the returns to be a random process. The characteristics of the SV models explain better the modern financial innovations than the ARCH models.
First let's consider the underlying continuous process: the SV price dynamics comes from the movements of an equity index Y_t and its stochastic volatility V_t via a continuous diffusion by a Brownian motion (Hull and White (1987)):

$$\begin{aligned} d\log Y_t &= \nu dt + \sqrt{V_t} dB_t^P \\ d\log V_t &= \kappa(\gamma - \log V_t)dt + \tau dB_t^V. \end{aligned} \tag{1}$$

Where the driver parameters of volatility are $(\nu, \kappa, \gamma, \tau)$ and (B_t^P, B_t^V) are the Brownian motions. As we can access data only in discrete time, it is natural to take the Euler discretization of equation (1) in order to get the stochastic volatility autoregressive model :

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

$$h_t = \mu + \phi h_{t-1} + \tau\eta_t. \tag{3}$$

Where $h_t = \log V_t$, ϵ_t, η_t are iid. normal errors. We take $\nu = 0$ as a simplification, $\mu = \kappa\gamma, \phi = 1 - \kappa$. Here in this setting h_t is the underlying stochastic volatility which can be interpreted as the unobservable information on the markets, and $\theta = (\mu, \phi, \sigma_\epsilon, \tau)$ are the vector of parameters. The initial state is $h_1 \sim \mathcal{N}(\mu, \tau^2/(1-\phi^2))$.
Our interest is to estimate the parameters in the discrete model and preferably to extract the volatility sequence $\boldsymbol{h}_{1:T} = (h_1, ..., h_T)$ in order to evaluate the uncertainty surrounding the returns. The arising difficulty in estimating the model given by equation $(2), (3)$ is that y_t depends non-linearly on the states h_t. We are facing a non-linear state-space model, where equation (2) defines the measurement equation, equation (3) defines the transition equation. The measurement equation represents the relation between the observed data y_t and the unobserved state variable $\boldsymbol{h}_{1:T}$. The transition equation indicates the evolution of the state variable.
Due to the non-linear dependence in (2), we cannot use Kalman filter equations to obtain the likelihood function of the observations. One response to this obstacle is to approximate the

measurement equation by linearization:

$$\ln y_t^2 = h_t + \ln \epsilon_t^2, \quad \epsilon_t \sim \mathcal{N}(0,1) \tag{4}$$

The obvious choice would be the maximum likelihood estimation method (Harvey et al. 1994) by approximating the density of $\ln \epsilon_t^2$ by a normal density. The downfall of the resulting linear Gaussian state-space model is that it does not compensate for approximating by only one normal density, hence the resulting estimation is only a quasi maximum likelihood estimate (QML).
Another linearization of the measurement equation was introduced in Kim et al. (1998) and Omori et al. (2007) where the density of $\ln \epsilon_t^2$ is approximated by a mixture of normal densities, often called as auxiliary mixture sampling:

$$\ln \epsilon_t^2 \sim \sum_{i=1}^{N} p_i \rho(m_i, s_i^2) \tag{5}$$

$$(\ln \epsilon_t^2 | r_t = i) \sim \mathcal{N}(m_i, s_i^2) \quad r_t \in 1, 2, ..., N \tag{6}$$

Where p_i is the weight of the mixture element, m_i, s_i^2 are the mean and variance of the normal density denoted by ρ.
Kim et al. (1998) used 7 mixture components, Omori et al. (2007) uses 10 components and his approximation is promising on Figure 3. We have to emphasize that the values for p, m and s^2 are fixed and they do not depend on any parameters or variable.
The QML method would correspond to a 1-component "mixture" in this sampling.
In conclusion, using the auxiliary mixture approximation, the model given by equation (4) is now conditionally a linear Gaussian state-space model given the indicators $\mathbf{r} = (r_1, r_2, ..., r_T)$.
In this setting, the next step is a Bayesian Markov Chain Monte Carlo (MCMC) approach with data augmentation instead of maximizing the likelihood function. If one assumes that the element r_t of the mixture is used at time t then the measurement equation of (4) can be rewritten as:

$$\ln y_t^2 = h_t + m_{r_t} + \xi_t, \quad \xi_t \sim \mathcal{N}(0, s_{r_t}^2) \tag{7}$$

From then on, the forward filtering backward sampling (FFBS) method can be used: conditional on the data Y, the indicator vector **r** and parameters θ the FFBS offers a sample from the states **h**. Then, given **h**, Y and the indicators **r** the parameter vector θ can be evaluated. Afterwards, given Y, **h** and θ each mixture node has a certain probability which can be computed.

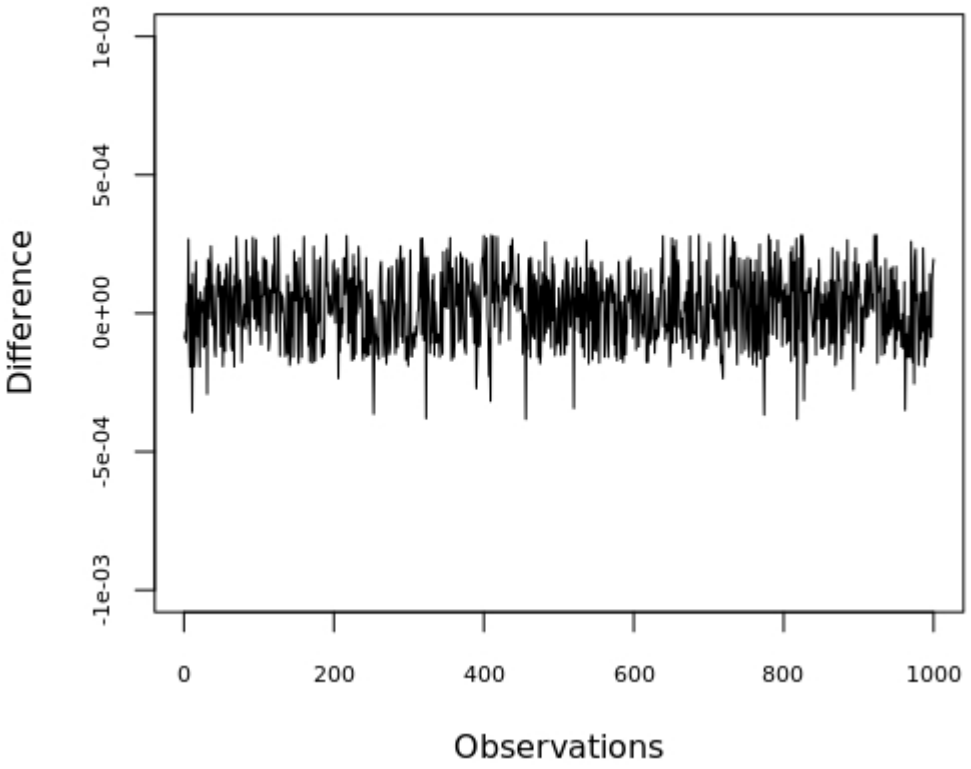


Figure 3: Difference between $\mathcal{X}^2$ distribution and the 10-component mixture distributions

2 MCMC

One of the main limitations of the SV model is that the distribution of the return y_t, conditional on the Information set $\mathcal{I}_{t-1}$ is often analytically intractable. To overcome this problem, we use Bayesian inference in this paper. This section reviews the concept of Bayesian statistics based on Petris et al. (2007) and Platanioti et al. (2005).

2.1 Bayesian Econometrics

The essence of Bayesian statistics is that specific variables might contain some uncertainty and because of this, they should be described by probabilistic tools. Denoting the variable of interest as a random variable with a certain probability distribution is supported by the fact that we seldom have perfect information about for example financial markets and we might be missing latent processes or dealing with measurement errors.
Bayesian statistics account for the missing information with the usage of conditional probability. Given the known information we can update our beliefs about the event of interest. The main result used in Bayesian econometrics is by Thomas Bayes that says:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}.$$

The expression says that the probability of A given B is the fraction of the probability of B given A times the marginal probability of A and the marginal probability of B.
We can also interpret the expression as the updating scheme of our "learning process" about our event of interest. If we think that event B contains information about event A, but event A is unobservable, then knowing that event B occurred must change our prior beliefs about event A. In most cases in practice, event B is usually something that can be observed or sampled, event A is usually the vector of parameters that describes the model. The observable process is usually denoted by a random variable y, and the parameters of interest are usually denoted by a vector θ. The Bayesian inference of θ is given by the Bayes formula:

$$\pi(\theta|y) = \frac{f(y|\theta)\pi(\theta)}{\int_\theta f(y|\theta)\pi(\theta)d\theta} \propto f(y|\theta)\pi(\theta)$$

Where $f(y|\theta)$ is called the likelihood function or the model specification, $\pi(\theta)$ is called the prior and m(y) is the marginal density of y. The prior, $\pi(\theta)$ should be chosen as to properly mirror our beliefs about θ.
The striking difference from the frequentist viewpoint in this procedure is that with the Bayes

rule, the parameters of interest θ have a probability distribution which can account for the uncertainty about the "perfectness of the collected information". We already treat θ as a random variable with its prior, then we collect the data y and update our beliefs to obtain the posterior distribution of the parameters $\pi(\theta|y)$.
Using the posterior of the parameters, we can draw inferences from the model that we assume to have generated the given data. By treating the parameters as random variables we can drop the notion of confidence intervals and simply express the probability of the parameters being in a certain interval.
One of the main drawbacks of Bayesian statistics is that the posterior distribution of the parameters of interest $\pi(\theta|y)$ is often analytically intractable, which means that is not of standard form. One can face computational problems in the posterior distribution, or it can happen that the posterior distribution does not even exist. One way to overcome these problems is to use specific sampling methods such as Markov Chain Monte Carlo (MCMC).

2.2 Markov Chain Monte Carlo methods

This section reviews two well-known Markov Chain Monte Carlo (MCMC) samplers. MCMC is a useful device to generate samples from arbitrary posterior distributions. Its main idea is to draw a sample from approximated distributions and then use them in order to update and improve the approximation by correcting them. Each iteration step of an MCMC sampler depend on the latest draw and try to make it better in order to converge to the target distribution of the parameters of interest.

2.2.1 Sampling methods

In my thesis I am using two particular MCMC sampling methods: the Gibbs sampler and the Metropolis-Hastings algorithm. I will now introduce the two procedures using the notation and the concept from Gelman et al. (2014).

- **Gibbs sampler**
  This specific Markov chain method has the striking advantage of usability when facing a multidimensional problem. To sample from the parameter of interest θ, we first partition it into d parts, namely $\theta = (\theta_1, \theta_2, ..., \theta_d)$. With each iteration, the Gibbs sampler goes through the whole subvector, drawing a new θ_j conditional on all the other $\theta_k, k \neq j$. At the end of each iteration there are exactly d (the number of partitions of θ) draws.
  At each iteration t, each θ_j^t is drawn - assuming all the other values of θ are given - from the following conditional distribution function:

$$p(\theta_j^t|\theta_{-j}^{t-1}, y)$$

Hence we update every component of θ conditional on the previous values of the other elements of θ, that is why this algorithm is also called the alternating conditional sampling method.

- **Metropolis-Hastings algorithm**
  The Metropolis-Hastings algorithm is a special case of the Metropolis algorithm. The method is the following:

  1. Draw a starting point θ_0 from a starting distribution $p_0(\theta)$. We can get this starting distribution from an approximation or it can also be a simple crude estimation.
  2. For $t = 1, 2, ...$:
     The generation of θ_t is a two-stage procedure in this algorithm for all t.
     First, we sample $\theta^\star$ from a proposal distribution at time t. This $\theta^\star$ is a candidate to be θ_t.
     In the second step, we have to calculate the following ratio of densities:

$$\mathrm{R} = \frac{p(\theta^\star|y)/J(\theta^\star|\theta_{t-1})}{p(\theta_{t-1}|y)/J(\theta_{t-1}|\theta^\star)}$$

     And then we can either accept or reject the proposed candidate given the following rule:

$$\theta_t = \begin{cases} \theta^\star & \text{with probability } \min(R, 1) \\ \theta_{t-1} & \text{otherwise} \end{cases}$$

     The ratio of the proposal distributions $J(\theta_{t-1}|\theta^\star)/J(\theta^\star|\theta_{t-1})$ correct for any arising bias.
     The ratio R is required to be calculable for all $(\theta, \theta^\star)$. Even if the proposed parameter is not accepted, it still counts as an iteration step.

2.2.2 Problems with the convergence rate

Using the notations of Brooks (1998), I briefly summarize the possible problem with the convergence rate.
For a given target distribution π, MCMC methods construct a chain $\{X_n\}$ which has π as its invariant distribution, no matter what the initial X_0 is. Although the used MCMC algorithm usually satisfies convergence, the arising issue is the speed at which the specific MCMC algorithm converges to the stationary distribution. This feature determines how many simulation steps we should run the chain before treating the simulated values as draws from the invariant π.

A related issue is the dependence of the draws. High dependence among the samples can result in very slow convergence of the average estimates to the expectations under π. Papaspiliopoulos et al. (2003) investigated the latter problem carefully and computed exact convergence rates for particular classes of MCMC algorithms.

2.2.3 Understanding the problem through an example

The two-component Gibbs sampler will be of particular importance in this subsection.
Let's try to understand why the methods work badly under some true parameters. I will focus on a linear state space model:
Non-centered parameterization:

$$\begin{aligned} y_t &= \mu + a_t + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2) \\ a_t &= \phi a_{t-1} + \eta_t \quad \eta_t \sim \mathcal{N}(0, \sigma_\eta^2) \\ a_1 &\sim \mathcal{N}(0, \sigma_\eta^2/(1-\phi^2)). \end{aligned} \tag{8}$$

Centered parameterization:

$$\begin{aligned} y_t &= \omega_t + \epsilon_t \quad \epsilon_t \sim \mathcal{N}(0, \sigma_\epsilon^2) \\ \omega_t &= \mu + \phi(\omega_{t-1} - \mu) + \eta_t \quad \eta_t \sim \mathcal{N}(0, \sigma_\eta^2) \\ \omega_1 &\sim \mathcal{N}(\mu, \sigma_\eta^2/(1-\phi^2)). \end{aligned} \tag{9}$$

Given these two parameterizations, let's focus on a simple case, when only one parameter is unknown. I will follow the approach of Gelfand et al. (1995) in order to consider the effect of the Gibbs-sampling in both cases. Consider $\sigma_\epsilon^2, \sigma_\eta^2$ and ϕ as the fixed parameters, and think about what happens to μ in the two parameterizations. As we do not have to sample the other parameters, the updating problem simplifies to sampling from μ|y,a and $\mu|\omega$,y. Clearly, the interesting feature is the correlation in the Gibbs sampling. Pitt and Shephard (1999) defined the autocorrelation at lag 1 for the non-centered parameterization as $\rho_\mu(1; a)$ and similarly for the centered one by $\rho_\mu(1; \omega)$.
After a rather long computation, they get the following important feature:

$$\begin{aligned} \rho_\mu(1; a) &\to 1 \quad as \quad \phi \to 1 \\ \rho_\mu(1; \omega) &\to 0 \quad as \quad \phi \to 1 \end{aligned}$$

Papaspiliopoulos et al. (2003) computed the convergence rate for these parameterizations using the results of Roberts and Sahu (1997):

$$\begin{aligned} \rho_c &= 1 - \kappa \\ \rho_{nc} &= \kappa, \quad \text{where} \\ \kappa &= \frac{\sigma_\eta^2}{\sigma_\eta^2 + \sigma_\epsilon^2}. \end{aligned} \tag{10}$$

$1 - \kappa$ is called the Bayesian fraction of missing information. Equation (10) means that the centered method will face problems when the observed data is not informative, while the non-centered method will perform badly, when the data is informative.
Papaspiliopoulos et al. (2003) also calculated the asymptotic relative performance of the two parameterizations for Gaussian state-space models:

$$\frac{1 - \rho_{nc}}{1 - \rho_c} = \frac{1 - \kappa}{\kappa} \frac{1 - \phi}{1 + \phi} \tag{11}$$

So the relative performance of the two parameterizations depends on the "signal/noise ratio" and the persistence parameter. Analyzing equation (11), we can say that given the error terms, the higher the dependence (parameter ϕ) among the latent volatilities, the more preferable the centered model becomes over the non-centered one.
Given these features, we can expect to have highly correlated (slowly converging) samples of μ when ϕ is close to one under the non-centered model, and rapidly converging samples under the centered model.
I followed Papaspiliopoulos (2003) estimating μ with both models using the two-component Gibbs sampling method as follows:

1. a|y,μ then μ|a,y
2. ω|y,μ then $\mu|\omega$,y

So basically sampling the states given all parameters then all states given μ.
The conditional densities are the following:

$$
\begin{aligned}
a|y,\mu &\sim \mathcal{N}(Vb - \mathbf{1}\mu, V),\\
\mu|a,y &\sim \mathcal{N}(\overline{y} - \overline{a}, \sigma_\epsilon^2/n),\\
\omega|\mu,y &\sim \mathcal{N}(Vb, V),\\
\mu|\omega,y &\sim \mathcal{N}(p/q, \sigma_\eta^2/p)
\end{aligned} \tag{12}
$$

where

$$
\begin{aligned}
V &= (\sigma_\epsilon^{-2} I_n + D^{-1})^{-1},\\
b &= \sigma_\epsilon^{-2} y + D^{-1}\mathbf{1}\mu,\\
p &= (n-1)(1-\phi)^2 + (1-\phi^2),\\
q &= \omega_1(1-\phi^2) + (1-\phi)\sum_{i=2}^{n}(\omega_t - \phi\omega_{t-1})
\end{aligned} \tag{13}
$$

and

$$
D^{-1} = 1/\sigma_\eta^2 \begin{pmatrix}
1 & -\phi & 0 & \dots & 0\\
-\phi & 1+\phi^2 & -\phi & \dots & 0\\
0 & \ddots & \ddots & \ddots & 0\\
\vdots & \ddots & -\phi & 1+\phi^2 & -\phi\\
0 & \dots & 0 & -\phi & 1
\end{pmatrix}. \tag{14}
$$

With both parameterizations, I simulated 200 observations, sampled 20000 times and looked at the path of μ and its autocorrelation. In the following figures we can see that our prediction was correct: when ϕ is close to 1 and the ratio of the errors is small ($\phi = 0.98$, $\sigma_\eta^2 = 0.02$, $\sigma_\epsilon^2 = 0.1$), the speed of convergence of the centered model is almost zero (near independence), while the non-centered parameterization has autocorrelation beyond lags of 200. Both samplers of μ vary around its true value.

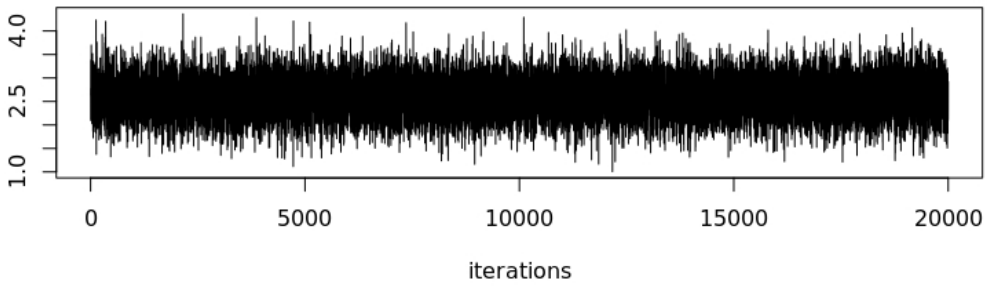


Figure 4: Centered method varying around true $\mu = 3$

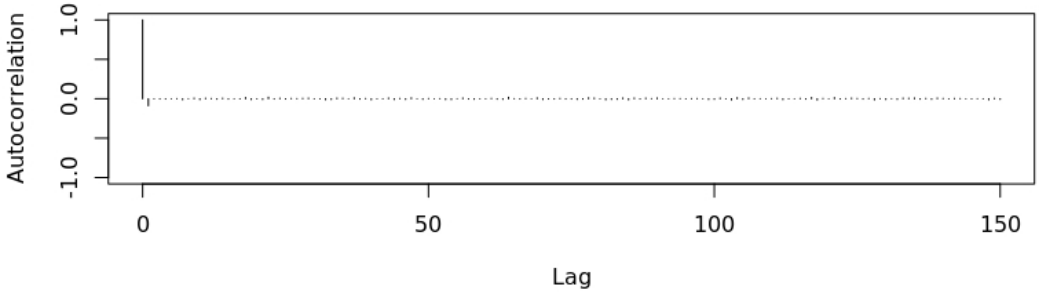


Figure 5: Uncentered method, varying around true $\mu = 3$

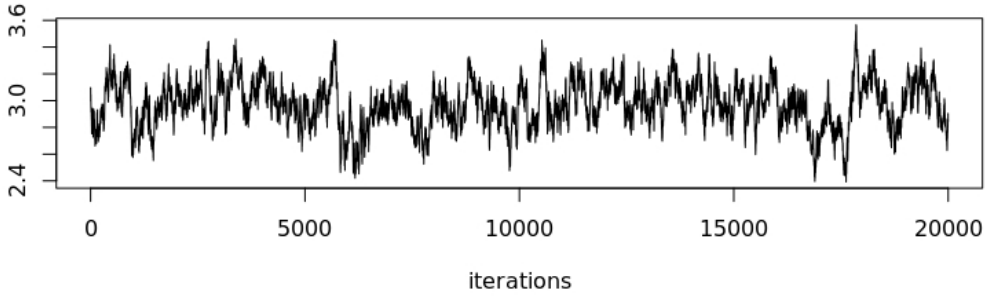


Figure 6: Autocorrelation for centered method

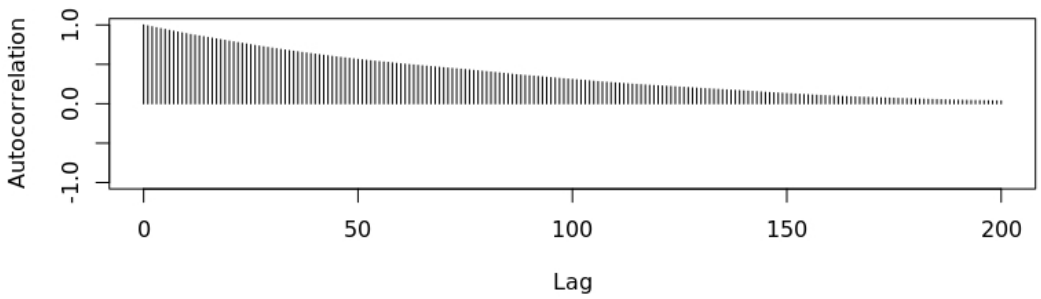


Figure 7: Autocorrelation for uncentered method

When ϕ is close to zero, then the centered model has problems with convergence (with notable autocorrealtion at lag 100) and the non-centered model converges rapidly:

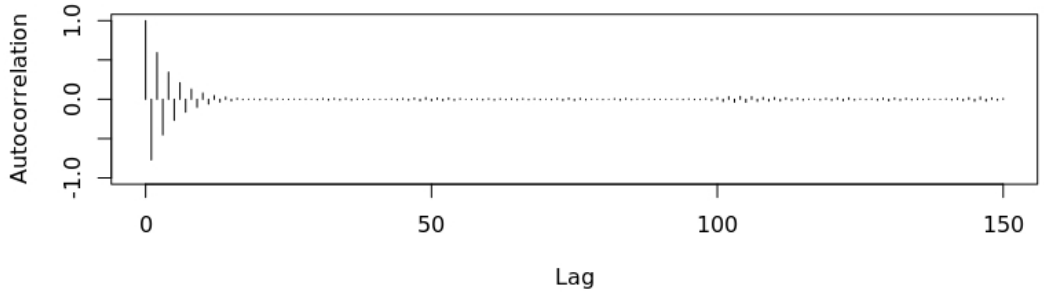


Figure 8: Autocorrelation for centered method, $\phi = 0.05$

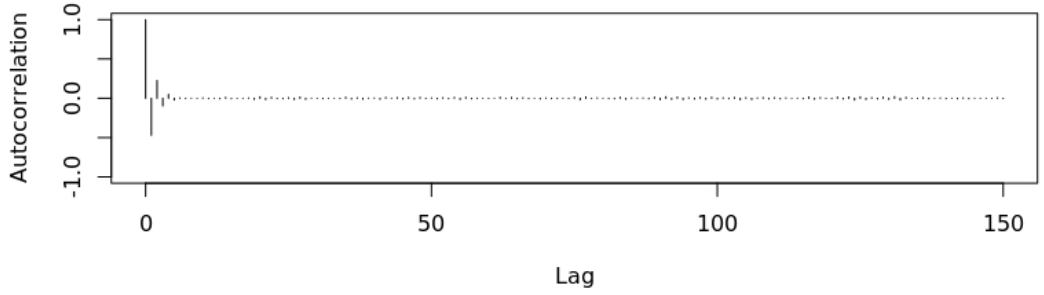


Figure 9: Autocorrelation for centered method,$\phi = 0.5$

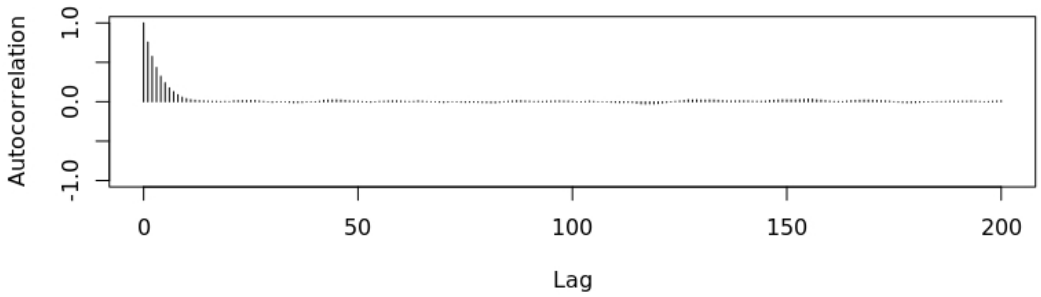


Figure 10: Autocorrelation for uncentered method, $\phi = 0.75$

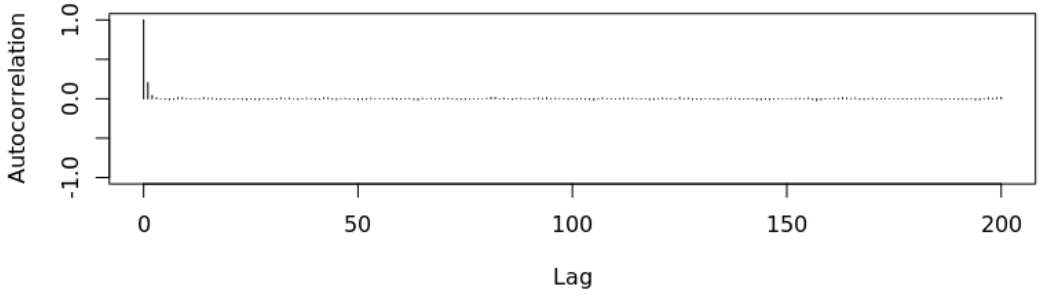


Figure 11: Autocorrelation for uncentered method, $\phi = 0.1$

3 Ancillarity Sufficiency Interweaving Strategy

The parameterization of MCMC algorithms, such as data augmentation (DA) has been of particular interest in statistics for many years. I will now summarize the paper by Yu and Meng (2011) which provides a remedy to the problem of the possible inefficiency of estimating centered and non-centered models.

3.1 Introduction of the method

The ancillarity-sufficiency interweaving strategy (ASIS) uses two special data-augmentation schemes, the ancillarity-augmentation (AA) and the sufficient-augmentation (SA) and bounds them together by interweaving them in each iteration step of the sampler. ASIS exploits the fact that the missing data is an ancillary statistic in the AA scheme, meanwhile in the SA scheme, the missing data is a sufficient statistic. In a Bayesian setting, being an ancillary statistic means that the parameter of interest and the missing data are a priori independent, while being a sufficient statistic means that the parameter of interest depends only on the missing data.
Although both of the parameterizations have the same target distribution, it is well established (Meng and van Dyk (1997)) that there are cases when one parameterization has a fast convergence rate while the other has a rather slow convergence rate (or can even fail to converge).
Yu and Meng (2011) explain their method with a simple two-level normal hierarchical model:

$$\begin{aligned} Y_{obs}|(\theta, Y_{mis}) &\sim \mathcal{N}(Y_{mis}, 1) \\ Y_{mis}|\theta &\sim \mathcal{N}(\theta, V), \end{aligned} \tag{15}$$

where the posterior:

$$\theta|Y_{obs} \sim \mathcal{N}(Y_{obs}, 1+V).$$

Using Y_{mis} as the unobservable data, the DA method iterates between drawing $Y_{mis}|(\theta, Y_{obs})$ and sampling $\theta|Y_{obs}, Y_{mis}$:

$$\begin{aligned} Y_{mis}|(\theta, Y_{obs}) &\sim \mathcal{N}\left(\frac{\theta + VY_{obs}}{1+V}, \frac{V}{1+V}\right) \\ \theta|(Y_{obs}, Y_{mis}) &\sim \mathcal{N}(Y_{mis}, V) \end{aligned} \tag{16}$$

In this setting Y_{mis} is a sufficient statistic for θ. And the convergence rate is $\frac{1}{1+V}$

Moreover, if we let: $\widetilde{Y}_{mis} = Y_{mis} - \theta$ and use $\widetilde{Y}_{mis}$ as the missing data, then the model changes to:

$$\begin{aligned} Y_{obs}|(\theta, \widetilde{Y}_{mis}) &\sim \mathcal{N}(\widetilde{Y}_{mis} + \theta, 1) \\ \widetilde{Y}_{mis}|\theta &\sim \mathcal{N}(0, V). \end{aligned} \tag{17}$$

This results in the same target distribution:

$$\theta|Y_{obs} \sim \mathcal{N}(Y_{obs}, 1 + V).$$

This gives another algorithm:

$$\begin{aligned} \widetilde{Y}_{mis}|(\theta, Y_{obs}) &\sim \mathcal{N}\left(\frac{V(Y_{obs} - \theta)}{1+V}, \frac{V}{1+V}\right) \\ \theta|(Y_{obs}, \widetilde{Y}_{mis}) &\sim \mathcal{N}(Y_{obs} - \widetilde{Y}_{mis}, 1). \end{aligned} \tag{18}$$

In this setting, $\widetilde{Y}_{mis}$ is an ancillary statistic for θ. And the convergence rate is $\frac{V}{1+V}$.
Concerning the convergence rates, if V is small, then the first parameterization has a rate close to one, while the rate of the second one is close to zero. If the value of V is big, then the rates are the other way around.

3.2 How ASIS works

Yu and Meng (2011) continue with explaining their method as follows.
To use the interweaving strategy, we need two DA algorithms say Y_{mis} and $\widetilde{Y}_{mis}$ such that their joint distribution $p(Y_{mis}, \widetilde{Y}_{mis}|Y_{obs}, \theta)$ conditional on θ and Y_{obs} is well defined. This joint distribution is often degenerate in the sense that $\widetilde{Y}_{mis} = M(Y_{mis}; \theta)$ is a one-to-one mapping for given θ.

The two DA schemes lead to two algorithms, where one iterates between:
Algorithm I:
Step 1. Draw $Y_{mis} \sim p(Y_{mis}|\theta)$
Step 2. Draw $\theta \sim p(\theta|Y_{mis})$
Algorithm II:
Step $\widetilde{1}$. Draw $Y_{mis} \sim p(\widetilde{Y}_{mis}|\theta)$
Step $\widetilde{2}$. Draw $\theta \sim p(\theta|\widetilde{Y}_{mis})$

The interweaving scheme replaces Step 2. and Step $\widetilde{1}$. with a single step where we do not condition on θ when drawing $\widetilde{Y}_{mis}$

$$[Y_{mis}|\theta^t] \rightarrow [\widetilde{Y}_{mis}|Y_{mis}] \rightarrow [\theta^{t+1}|\widetilde{Y}_{mis}]$$

Yu and Meng (2011) claim that it is often beneficial to first draw $\theta \sim p(\theta|Y_{mis})$ and then sample $Y_{mis} \sim p(\widetilde{Y}_{mis}|Y_{mis}, \theta)$. Subsequently, the previous scheme expands to:

$$[Y_{mis}|\theta^t] \rightarrow [\theta|Y_{mis}] \rightarrow [\widetilde{Y}_{mis}|Y_{mis}, \theta] \rightarrow [\theta^{t+1}|\widetilde{Y}_{mis}]$$

It is worth mentioning that the only difference between the interweaving and the alternating scheme is that here we draw the next sample from the conditional $\widetilde{Y}_{mis} \sim p(\widetilde{Y}_{mis}|Y_{mis}, \theta)$ instead of only conditioning on θ. While this change introduces more dependence in the sampling of $\widetilde{Y}_{mis}$, as it is a function of Y_{mis} and θ, it also accelerates convergence as it causes less dependence between θ^t and θ^{t+1}.

In my computation I use the Global Interweaving Strategy which updates the vector of parameters jointly and does not partition it into its components. That means that first we need to sample the missing data (Y^t_{mis}), then conditional on them we can draw a "temporary" vector of parameters ($\theta^{t+0.5}$). This will help us sample the missing data with the other parameterization ($\widetilde{Y}^{t+1}_{mis}$) using the sampled pair of $(Y_{mis}, \theta^{t+0.5})$, finally, we can sample the new vector of parameters (θ^{t+1}). Formally, the method is the following:

Global Interweaving Strategy (GIS):

Non-integer superscripts index intermediate draws.

- Step 1. Draw Y_{mis} given θ : $Y^t_{mis}|\theta^t$
- Step 2. Draw θ given Y_{mis} : $\theta^{t+0.5}|Y^t_{mis}$
- Step 3. Redraw θ given $\widetilde{Y}_{mis}$: $\theta^{t+1}|\widetilde{Y}^{t+1}_{mis}$
  where $\widetilde{Y}^{t+1}_{mis} \sim p(\widetilde{Y}_{mis}|Y^t_{mis}, \theta^{t+0.5})$

4 Parameter estimation

In the following, I will describe the method introduced by Kastner and Frühwirth-Schnatter (2014), which I will use in my computations.
Having motivated the importance of stochastic volatility models in the first section, it is not a surprise that the SV model has been extensively studied. The method of Kastner and Frühwirth-Schnatter (2014) can be partitioned into two main parts: first an update of the latent variable (stochastic volatility) and the second part is a joint update of the latent variable and the vector of parameters.

4.1 Outline of the estimation

Kastner and Frühwirth-Schnatter (2014) introduce the "centered" SV model which is very similar to equation (2) and (3):

$$\begin{aligned} y_t &= \exp(h_t/2)\epsilon_t \\ h_t &= \mu + \phi(h_{t-1} - \mu) + \sigma\eta_t. \end{aligned} \tag{19}$$

They assume ϵ_t, η_s to be iid. standard normal innovations independent for all t,s $\in 1, ...T$. Throughout the analysis we consider $|\phi| < 1$ to get a stationary autoregressive volatility process. As before, **h**$= (h_1, ..., h_T)$ is the time-varying latent volatility process. The initial state variable is distributed as: $h_0|\mu, \phi, \sigma \sim \mathcal{N}(\mu, \sigma^2/(1-\phi^2))$.
The basic properties of this model are the following:

- The unconditional mean of h_t is: $\mathcal{E}(h_t) = \frac{\mu}{1-\phi}$
- The unconditional variance of h_t is: $\mathcal{V}(h_t) = \frac{\sigma^2}{1-\phi^2}$
- As y_t is the product of two kind of processes, $\exp(h_t/2)$ and iid. normal innovations, y_t is stationary if and only if h_t is stationary. As h_t is assumed to be stationary, it follows that y_t is also stationary (Platanioti et al. (2005)).

When simulating state-space models, an efficiency improvement is often achieved by reparameterization. Kim et al. (1998) and many others consider a "non-centered" version of equation (2) and (3):

$$\begin{aligned} y_t &\sim \mathcal{N}(0, \omega \exp(\sigma a_t)) \\ a_t &= \phi a_{t-1} + \eta_t \quad \eta_t \sim \mathcal{N}(0, 1) \end{aligned} \tag{20}$$

In the non-centered parameterization μ is shifted from the transition equation to the measurement equation by the following transformation: $a_t = (h_t - \mu)/\sigma$. The parameter $\omega = e^{\mu}$.
They linearize the measurement equations and use the aforementioned 10-component mixture of Gaussians given by Omori (2007). This method enables them to draw h_t directly from $p(\boldsymbol{h}|\sigma, \phi, \mu, \boldsymbol{y})$. This is possible via several methods i.e. the forward filtering backward sampling (FFBS, Carter and Kohn (1994)) or the Cholesky-factorization of the precision matrix (McCausland 2011).
In the following subchapters, I will expound the steps needed for estimating **h** and the parameters of interest.

1. Sampling the latent volatilities via Forward Filtering Backward Sampling.

2. Sampling μ, σ^2, ϕ via Bayesian regression: it can be either a one-block Metropolis-Hastings sampler for all parameters in the centered model and a two-block Gibbs sampler for μ and σ^2 and Metropolis-Hastings sampler for ϕ in the non-centered model.

3. Updating the indicator vector **r**.

4.2 Forward filtering backward sampling (FFBS)

In order to make inference about the parameters of interest, first we have to sample the latent volatility process **h** and treat it as known. As mentioned before, the linearization of the measurement equation and the 10 component mixture of Gaussian innovations make it possible to sample the latent states directly from $p(\boldsymbol{h}|\sigma, \phi, \mu, \boldsymbol{y})$. I am going to use the forward filtering backward sampling method for my computation and not the Cholesky-factorization as in Kastner and Frühwirth-Schnatter (2014). In the following paragraph I will briefly summarize the FFBS method based on West and Harrison (1997).

The univariate Gaussian Dynamic Linear Model (GDLM) with intercepts is represented by:

$$y_t|h_t, \Psi \sim \mathcal{N}(\lambda_t + F_t h_t, \upsilon_t^2)$$
$$h_t|h_{t-1}, \Psi \sim \mathcal{N}(\mu + G_t h_{t-1}, \sigma_t^2),$$

where we condition on $\Psi = (\lambda_{1:T}, \mu, F_{1:T}, G_{1:T}, \upsilon_{1:T}^2, \sigma_{1:T}^2)$, and treat the initial value as known $h_0 \sim \mathcal{N}(m_0, C_0)$. Moreover, the observational and the transition errors are not auto- nor cross-correlated with each other.

Based on *Theorem 4.1.* of West and Harrison (1997), the one-step forecast and posterior distributions are as follows:

$$\begin{aligned} h_t|y_{1:t} &\sim \mathcal{N}(m_t, C_t), \\ h_t|y_{1:t-1} &\sim \mathcal{N}(b_t, R_t) \quad \text{and} \\ y_t|y_{1:t-1} &\sim \mathcal{N}(f_t, Q_t) \quad \text{is the one-step forecast.} \end{aligned}$$

With the subsequent recursions we get:

$$\begin{aligned} m_t = b_t + A_t e_t \quad &\text{and} \quad C_t = R_t - A_t^2 Q_t, \\ A_t = R_t F_t / Q_t \quad &\text{and} \quad e_t = y_t - f_t \quad \text{are the Kalman gain and the prediction error,} \\ b_t = \mu + G_t m_{t-1} \quad &\text{and} \quad R_t = G_t^2 C_{t-1} + \sigma_t^2, \\ f_t = \lambda_t + F_t b_t \quad &\text{and} \quad Q_t = F_t^2 R_t + v_t^2. \end{aligned} \tag{21}$$

In this sampling method we also need the conditional and smoothed densities which is given in Lopes and Tsay (2011):

$$\begin{aligned} h_t|h_{t+1}, y_t &\sim \mathcal{N}(z_t, Z_t) \\ h_t|y_{1:T} &\sim \mathcal{N}(m_t^T, C_t^T) \end{aligned}$$

where

$$\begin{aligned} z_t = m_t + B_t(h_{t+1} - b_{t+1}) \quad &\text{and} \quad Z_t = C_t - B_t^2 R_{t+1} \\ m_t^T = m_t + B_t(m_{t+1}^T - b_{t+1}) \quad &\text{and} \quad C_t^T = C_t + B_t^2(R_{t+1} - C_{t+1}^T) \end{aligned}$$

After applying this sampling method we get the approximated latent volatility process.
To use the 10-component mixture of normal distribution approximation, we have to do the following in each iteration step:
Denote with $\epsilon_t^\star = y_t - h_t$, where h_t is the approximated latent state sampled in the previous iteration step. Then, we compute the probabilities $\Pr(r_t = k)$ for $k \in \{1, 2, ..., 10\}$ and $t \in \{1, 2, ..., T\}$ according to :

$$\Pr(r_t = k) = \frac{p_k \mathcal{N}(\epsilon_t^\star | m_{r_k}, s_{r_k}^2)}{\sum_{j=1}^{10} p_j \mathcal{N}(\epsilon_t^\star | m_{r_j}, s_{r_j}^2)},$$

where p_k is the weight of the mixture element, $\mathcal{N}(\epsilon_t^\star | m_{r_k}, s_{r_k}^2)$ is the density function of the normal distribution with mean and variance $m_{r_k}, s_{r_k}^2$. Table 1. contains the values of $m_{r_k}, s_{r_k}^2, p_{r_k}$ given in Omori et al. (2007). Then we sample the indicator vector **r**, given the calculated probabilities.

Each element of the sampled indicator vector **r** determines which normal distribution we should use in the linearized equation (7). This approximation with the 10-component mixture of normal distributions will give us a better approximation of the latent volatility process. All we have to do now is to run the FFBS method, now with $\lambda_t = m_{r_t}$ and $v_t^2 = s_{r_t}^2$ in equation (21), where the pair of $(m_{r_t}, s_{r_t}^2)$ is determined by the sampled indicator component r_t for all t (Lopes (2009)).

k	m_{r_k}	$s_{r_k}^2$	p_{r_k}
1	1.92677	0.11265	0.00609
2	1.34744	0.17788	0.04775
3	0.73504	0.26786	0.13057
4	0.02266	0.40611	0.20674
5	-0.85173	0.62699	0.22715
6	-1.97278	0.98583	0.18842
7	-3.46788	1.57469	0.12047
8	-5.55246	2.54498	0.05591
9	-8.68384	4.16591	0.01575
10	-14.65000	7.33342	0.00115

Table 1: Values of the moments for the mixture normal distributions

After drawing $\mathbf{h} = (h_1, h_2, ..., h_T)$, the initial value of the latent process can be sampled from $h_0|h_1, \mu, \phi, \sigma^2 \sim \mathcal{N}(\mu + \phi(h_1 - \mu), \sigma^2)$ in the centered method and from $a_0|a_1, \phi \sim \mathcal{N}(a_1\phi^2, 1)$ in the non-centered method.

4.3 Sampling μ, ϕ and σ^2 in the centered model

Given the latent log-volatility process **h**, we now want to sample the vector of parameters $\theta = (\mu, \phi, \sigma)$. Kastner and Frühwirth-Schnatter (2014) propose to use the conditional regression equation of the volatility process in order to have a tractable likelihood function ($p_{reg}(\mathbf{h}|\theta)$) for the Bayesian estimation process. The estimation procedure is as follows:

$$h_t = \gamma + \phi h_{t-1} + \sigma\eta_t \quad \text{where} \quad \eta_t \sim \mathcal{N}(0, 1). \tag{22}$$

They specify the following prior distributions for both parameterizatons, level μ is normally distributed with mean b_μ and variance B_μ, $\frac{\phi+1}{2}$ is beta-distributed and σ^2 is gamma-distributed. The beta distribution makes sure that ϕ is inside the unit ball $(-1, 1)$, thus we do not have to worry about stationarity.
In the regression equation (22), $\gamma = (1-\phi)\mu$ and its conditional prior follows from the prior of μ, that is $\gamma \sim \mathcal{N}((1-\phi)b_\mu, B_\mu(1-\phi)^2)$. With this transformation, we can interpret the regression coefficients as follows, γ is the expected value of the log-volatility at time t given that at time $t-1$ the log-volatility was zero. ϕ is the expected change in the log-volatility at time t for every additional change in the log-volatility at time $t-1$.
In the next paragraphs I will follow Kastner and Frühwirth-Schnatter (2014) and elaborate the Bayesian estimation process of the two parameterizations.
With treating the latent volatility process as an auxiliary regression model, we can express the conditional distribution of h_t given the parameters and the lagged volatility with a normal distribution: $h_t|\theta, h_{t-1} \sim \mathcal{N}(\gamma + \phi h_{t-1}, \sigma^2)$.
I will now regroup the parameters and regressors in order to have a more understandable notation. Let's denote $\beta = (\gamma, \phi)$ and $\boldsymbol{H} = (\mathbf{1}, \boldsymbol{h}_{0:(T-1)})$. Obviously, β contains the coefficients of the regression and $\boldsymbol{H}$ is a $T \times 2$ matrix which contains ones in the first column and the latent log-volatility variables in the second. Then we can now write down the formula of the likelihood function:

$$p_{reg}(\boldsymbol{h}|\theta) = \left(\frac{1}{2\pi\sigma^2}\right)^{T/2} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{h} - \boldsymbol{H}\beta)'(\boldsymbol{h} - \boldsymbol{H}\beta)\right\}. \tag{23}$$

We are interested in finding the posterior distribution $p(\theta|\boldsymbol{h})$. We can expand the posterior into the product of two conditional distributions as follows, $p(\theta|\mathbf{h}) = p(\gamma, \phi|\sigma^2, \mathbf{h})p(\sigma^2|\mathbf{h})$. Following the Bayes formula and the auxiliary regression model, we know that $p(\theta|\boldsymbol{h}) \propto p_{reg}(\boldsymbol{h}|\theta)p_{reg\_prior}(\theta)$. After analyzing the likelihood function (23), a pair of appropriate conjugate prior distributions is $p_{reg\_prior}(\theta) = p_{reg\_prior}(\beta|\sigma^2)p_{reg\_prior}(\sigma^2)$. Kastner and Frühwirth-Schnatter (2014) propose the following priors: $p_{reg\_prior}(\sigma^2) \propto \sigma^{-1}$ and $p_{reg\_prior}(\beta|\sigma^2) = \mathcal{N}_2(\mathbf{0}, \sigma^2\boldsymbol{B_0})$, where $\boldsymbol{B_0} = \text{diag}(B_0^{11}, B_0^{22})$, so γ and ϕ are a priori independent given σ^2.
After specifying the conjugate prior distributions and the likelihood function, we can derive the

posterior of the parameters:

$$\begin{aligned} p_{proposal}(\beta, \sigma^2|\mathbf{h}) \propto \quad & p_{reg}(\boldsymbol{h}|\beta, \sigma^2) p_{reg\_prior}(\beta|\sigma^2) p_{reg\_prior}(\sigma^2) \\ & \propto \left(\frac{1}{\sigma^2}\right)^{T/2} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{h'h} - 2\beta'\boldsymbol{H'h} + \beta'\boldsymbol{H'H}\beta)\right\} * \\ & \left(\frac{1}{\sigma^4 B_0^{11} B_0^{22}}\right)^{1/2} \exp\left\{-\frac{1}{2\sigma^2}\beta'\boldsymbol{B_0^{-1}}\beta\right\} * \frac{1}{\sigma} \end{aligned} \tag{24}$$

We have to compare this expression to the pdf of a multivariate normal distribution with homoskedastic errors in order to extract the conditional posterior of β. Suppose $\boldsymbol{w} \sim \mathcal{N}_n(\xi, \Sigma\sigma^2)$, then

$$f(\boldsymbol{w}) \propto \exp\left\{-\frac{1}{2}(\boldsymbol{w}'\Sigma^{-1}\boldsymbol{w} - 2\xi'\Sigma^{-1}\boldsymbol{w} + \xi'\Sigma^{-1}\xi)\right\}$$

Now we can identify the mean and the variance-covariance matrix in equation (24):
$\Sigma^{-1}\sigma^{-2} = (\boldsymbol{H'H} + \boldsymbol{B}_0^{-1})/\sigma^2$ so the variance-covariance matrix is $\Sigma = (\boldsymbol{H'H} + \boldsymbol{B}_0^{-1})^{-1}$.
While $\xi'\Sigma^{-1} = \boldsymbol{h'H}$. Multiplying from the right with Σ yields: $\xi' = \boldsymbol{h'H}\Sigma$. After transposing the terms, we get the final form of the mean, which is $\xi = \Sigma\boldsymbol{H'h}$.
Hence,

$$\gamma, \phi|\sigma^2, \boldsymbol{h} \sim \mathcal{N}_2\left((\boldsymbol{H'H} + \boldsymbol{B}_0^{-1})^{-1}\boldsymbol{H'h}, (\boldsymbol{H'H} + \boldsymbol{B}_0^{-1})^{-1}\sigma^2\right). \tag{25}$$

Then the conditional posterior of σ^2 can be computed after collecting the terms containing σ^2 from equation (24):

$$\left(\frac{1}{\sigma^2}\right)^{\frac{T}{2}+\frac{3}{2}} \exp\left\{-\frac{1}{2\sigma^2}(\boldsymbol{h'h} - 2\beta'\boldsymbol{H'h} + \beta'\boldsymbol{H'H}\beta + \beta'\boldsymbol{B_0^{-1}}\beta)\right\}$$

This looks like the kernel of an inverse gamma distribution. It can be easily read off, that the first parameter (shape) of the inverse gamma distribution is $(T-1)/2$, for the second parameter

(rate) a facilitating calculation is the following:

$$
\begin{aligned}
&\exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{h'h}-2\beta'\boldsymbol{H'h}+\beta'\boldsymbol{H'H}\beta+\beta'\boldsymbol{B_0^{-1}}\beta\right)\right\}\\
=&\exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{h'h}-2\beta'\boldsymbol{H'h}+\beta'\left(\boldsymbol{H'H}+\boldsymbol{B_0^{-1}}\right)\beta\right)\right\}\\
=&\exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{h'h}-2\beta'\boldsymbol{H'h}+\underbrace{(\boldsymbol{H'h})'(\boldsymbol{H'H}+\boldsymbol{B}_0^{-1})^{-1}}_{\beta'}\left(\boldsymbol{H'H}+\boldsymbol{B_0^{-1}}\right)\beta\right)\right\}\\
=&\exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{h'h}-2\beta'\boldsymbol{H'h}+(\boldsymbol{H'h})'\mathbb{I}_2\beta\right)\right\}\\
=&\exp\left\{-\frac{1}{2\sigma^2}\left(\boldsymbol{h'h}-\beta'\boldsymbol{H'h}\right)\right\}
\end{aligned}
$$

Hence, the posterior of σ^2 conditional on **h** is:

$$
\begin{aligned}
\sigma^2|\boldsymbol{h} &\sim \mathcal{IG}(c_T, C_T) \\
where \quad c_T &= (T-1)/2 \\
and \quad C_T &= \frac{1}{2}\left(\boldsymbol{h'h}-((\boldsymbol{H'H}+\boldsymbol{B}_0^{-1})^{-1}\boldsymbol{H'h})'\boldsymbol{H'h}\right)
\end{aligned}
\tag{26}
$$

We sample the parameters with the Metropolis-Hastings algorithm introduced in Chapter 2. Kastner and Frühwirth-Schnatter (2014) propose to use the posterior of the regression equation (24) as the proposal density for the Metropolis-Hastings acceptance propability computation. The required ratio of densities $R = \frac{p(\theta^\star|y)/J_t(\theta^\star|\theta_{t-1})}{p(\theta_{t-1}|y)/J_t(\theta_{t-1}|\theta^\star)}$ is the following:

$$
R = \frac{p(h_0|\theta_{new})p(\gamma_{new}|\phi_{new})p(\phi_{new})p(\sigma^2_{new})}{p(h_0|\theta_{old})p(\gamma_{old}|\phi_{old})p(\phi_{old})p(\sigma^2_{old})} * \frac{p_{proposal}(\beta_{old}|\sigma^2_{old})p_{proposal}(\sigma^2_{old})}{p_{proposal}(\beta_{new}|\sigma^2_{new})p_{proposal}(\sigma^2_{new})}
$$

Then, the acceptance probability of the proposed new parameters is $\min(R, 1)$.

4.4 Sampling μ, ϕ and σ^2 in the non-centered model

In the non-centered parameterization, the latent volatility process in equation (20) simplifies to a regression equation without a constant. Sampling the only parameter in that equation is rather simple with the well-known OLS formula:
The regression model equation is $a_t = \phi a_{t-1} + \eta_t \quad \eta_t \sim \mathcal{N}(0, 1)$.

Using the OLS formula and a flat prior, we know that

$$\phi|\mathbf{a} \sim \mathcal{N}\left(\frac{\Sigma_{i=0}^{T-1} a_t a_{t-1}}{\Sigma_{i=0}^{T-1} a_t^2}, \frac{1}{\Sigma_{i=0}^{T-1} a_t^2}\right). \tag{27}$$

Moreover, if we use the Metropolis-Hastings sampling algorithm, then we need a ratio of densities in order to calculate the acceptance probability. Again, we use the derived posterior at equation (27) as the proposal density function.

$$R = \frac{p(a_0|\phi_{new})p(\phi_{new})/p_{proposal}(\phi_{new})}{p(a_0|\phi_{old})p(\phi_{old})/p_{proposal}(\phi_{old})}$$

Hence, the acceptance probability is $\min(1, R)$.

As for sampling μ and σ^2, we can rewrite equation (8) by transforming h_t into a_t by non-centering:

$$\begin{aligned} \widetilde{y}_t &= m_{r_t} + \sigma a_t + \mu + \xi_t \\ \text{where} \quad \widetilde{y}_t &= \log y_t^2 \end{aligned} \tag{28}$$

We can transform equation (28) into a more familiar regression model,

$$\widehat{\boldsymbol{y}} = \boldsymbol{X} \begin{bmatrix} \mu \\ \sigma \end{bmatrix} + \boldsymbol{\epsilon} \tag{29}$$

where $\boldsymbol{\epsilon} \sim \mathcal{N}_T(\mathbf{0}, \boldsymbol{I_T})$ and

$$\widehat{\boldsymbol{y}} = \begin{bmatrix} \frac{\widetilde{y}_1 - m_{r_1}}{s_{r_1}} \\ \vdots \\ \frac{\widetilde{y}_T - m_{r_t}}{s_{r_T}} \end{bmatrix} \quad , \quad \boldsymbol{X} = \begin{bmatrix} 1/s_{r_1} & a_1/s_{r_1} \\ \vdots & \vdots \\ 1/s_{r_T} & a_T/s_{r_T} \end{bmatrix}$$

Now, we can follow the same routine as in the centered method, that is to find the posterior of $p(\mu, \sigma|\boldsymbol{a})$ we have to use the Bayes formula $p_{proposal}(\mu, \sigma|\boldsymbol{a}) \propto p(\boldsymbol{a}|\mu, \sigma)p_{prior}(\mu, \sigma)$. The joint prior of the parameters of interest is $p_{prior}(\mu, \sigma) = \mathcal{N}_2(\boldsymbol{b_0}, \boldsymbol{B_0})$ where $\boldsymbol{b_0} =$

$(b_\mu, 0)'$ and $\boldsymbol{B_0} = \text{diag}(B_\mu, B_\sigma)$.
Then the posterior is as follows:

$$\begin{aligned} p(\mu, \sigma|\boldsymbol{a}) \propto \quad & p_{reg}(\boldsymbol{a}|\mu, \sigma)p(\mu, \sigma) \\ & \propto \exp\left\{-\frac{1}{2}(\widehat{\boldsymbol{y}}'\widehat{\boldsymbol{y}} - 2(\mu, \sigma)'\boldsymbol{X}'\widehat{\boldsymbol{y}} + (\mu, \sigma)'\boldsymbol{X}'\boldsymbol{X}(\mu, \sigma))\right\} * \\ & \left(\frac{1}{B_\mu B_\sigma}\right)^{1/2} \exp\left\{-\frac{1}{2}((\mu, \sigma)' - \boldsymbol{b_0})'\boldsymbol{B_0^{-1}}((\mu, \sigma)' - \boldsymbol{b_0})\right\} \end{aligned} \tag{30}$$

Comparing this formula to the multivariate normal pdf, we can easily read off the moments of the posterior distribution:
$\Sigma^{-1}/1 = (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{B}_0^{-1})/1$ so the variance-covariance matrix is $\Sigma = (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{B}_0^{-1})^{-1}$.
While $\xi'\Sigma^{-1} = \widehat{\boldsymbol{y}}'\boldsymbol{X} + \boldsymbol{b_0}'\boldsymbol{B_0^{-1}}$. Multiplying from the right with Σ yields: $\xi' = (\widehat{\boldsymbol{y}}'\boldsymbol{X} + \boldsymbol{b_0}'\boldsymbol{B_0^{-1}})\Sigma$.
After transposing the terms, we get the final form of the mean, which is $\xi = \Sigma(\boldsymbol{X}'\widehat{\boldsymbol{y}} + \boldsymbol{B_0^{-1}b_0})$.
Hence,

$$\mu, \sigma|\boldsymbol{a} \sim \mathcal{N}_2\left((\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{B}_0^{-1})^{-1}(\boldsymbol{X}'\widehat{\boldsymbol{y}} + \boldsymbol{B_0^{-1}b_0}), (\boldsymbol{X}'\boldsymbol{X} + \boldsymbol{B}_0^{-1})^{-1}\right). \tag{31}$$

4.5 Interweaving part

The second part of the method of Kastner and Frühwirth-Schnatter (2014), as mentioned before, is a joint update of the latent variable and the vector of parameters. The Ancillarity-Sufficiency Interweaving Strategy (ASIS) method developed by Yu and Meng (2011) is based on the idea of interweaving two parameterizations (in this case the centered (C) and non-centered (NC)) in order to increase the sampling efficiency. ASIS exploits the fact that **h** in C is a sufficient statistic for μ and σ while **a** from NC is an ancillary statistic for these parameters. Hence the ASIS method can converge even if C or NC fails to do so.
The ASIS method proceeds as the following (using C as the baseline method):

1. Draw **h**, following the method introduced in Chapter 4.2
2. Draw θ using Metropolis-Hastings algorithm, explained in Chapter 4.3
3. Move to NC by setting $\mathbf{a} = \frac{(\mathbf{h} - \mu)}{\sigma}$
4. Draw $\theta = (\mu, \sigma, \phi)$ with the sampling method introduced in Chapter 4.3

5. Move back to C by setting $\mathbf{h} = \mu + \sigma\mathbf{a}$

6. Redraw the mixture component indicators $\mathbf{r}$, see more details in Chapter 4.2

These steps should be repeated in order to estimate the parameters of interest and the latent volatility process.

5 ASIS through an example

In Chapter 4. I have introduced and expounded the method to sample the latent volatility process and the Bayesian estimation of the vector of parameters. I will now present the boosting power of ASIS in terms of autocorrelation of the samplers on a simulated time series. I set the true parameters to be $\mu_{true} = -10, \sigma_{true} = 0.1$ and $\phi_{true} = 0.98$. Then I followed Kastner and Frühwirth-Schnatter (2014) with setting the priors to be the following:

$$\mu \sim \mathcal{N}(-10, 10), \quad \frac{\phi+1}{2} \sim \mathcal{B}\left(40, \frac{80}{1.98} - 40\right) \quad \text{and} \quad \sigma^2 \sim \mathcal{G}\left(0.5, \frac{1}{0.02}\right)$$

I also used the same parameters for the auxiliary regression priors.
For the centered parameterization:

$$\sigma^2 \sim \mathcal{G}^{-1}(0.5, 0), \quad \gamma, \phi|\sigma^2 \sim \mathcal{N}_2(\mathbf{0}, \text{diag}(10^{12}, 10^8))$$

For the non-centered parameterization:

$$\mu, \sigma \sim \mathcal{N}_2((-10, 0)', \text{diag}(10, 0.01))$$

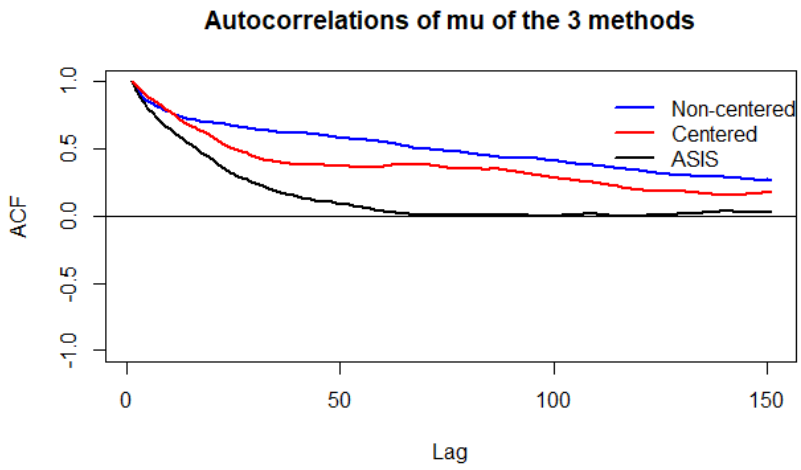
I followed Lopes (2009) to implement the FFBS method.
In Table 2., I summarize and compare the results from my code with the results from using the R-package 'stochvol'(Kastner (2013)). Both estimations ran 5000 iterations with a burnin period of 100. For each parameter, the first row contains the mean, the second row contains the standard deviation of the samplers.

Parameters	**Centered method**		**Non-centered method**		**ASIS method**	
	My code	'stochvol'	My code	'stochvol'	My code	'stochvol'
μ	-8.633	-9.5084	-11.52	-9.533	-8.95	-9.5574
Std. dev.	0.8869	0.5191	1.847	0.6170	1.256	0.7167
ϕ	0.985	0.9706	0.9963	0.970	0.9774	0.9726
Std. dev.	0.0125	0.0331	0.0036	0.0390	0.0174	0.0340
σ	0.0709	0.0604	0.107	0.0527	0.1335	0.0516
Std. dev.	0.0046	0.0476	0.0389	0.0562	0.0437	0.0508

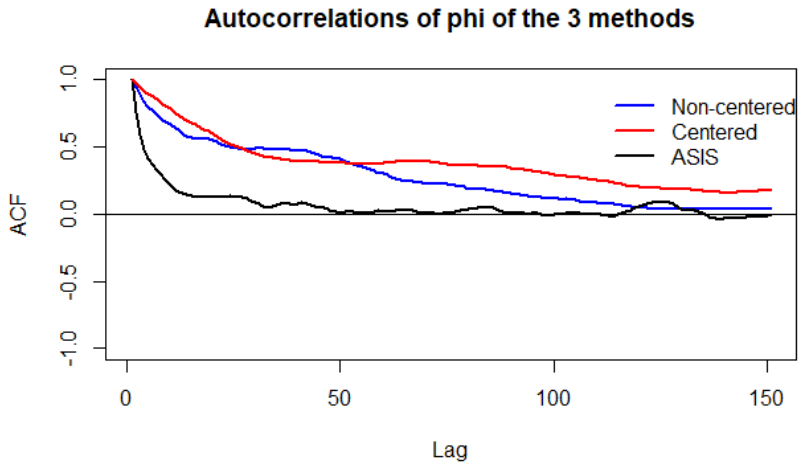
Table 2: Estimation results

It can be seen that the R-package 'stochvol'performs better estimating the parameters. It is also visible, that my code cannot estimate well the parameter μ while it gives a good enough estimation of ϕ and σ.
Concerning the methodology, the centered (non-centered) method first samples the latent volatility process given by Chapter 4.2, then proceeds to draw the parameters of interest using the posterior distributions derived in Chapter 4.3 (4.4 for the non-centered method). Then, treating the parameters as know, we can resample the latent states and the indicator vector $\mathbf{r}$.
The ASIS method proceeds as given in Chapter 4.5.

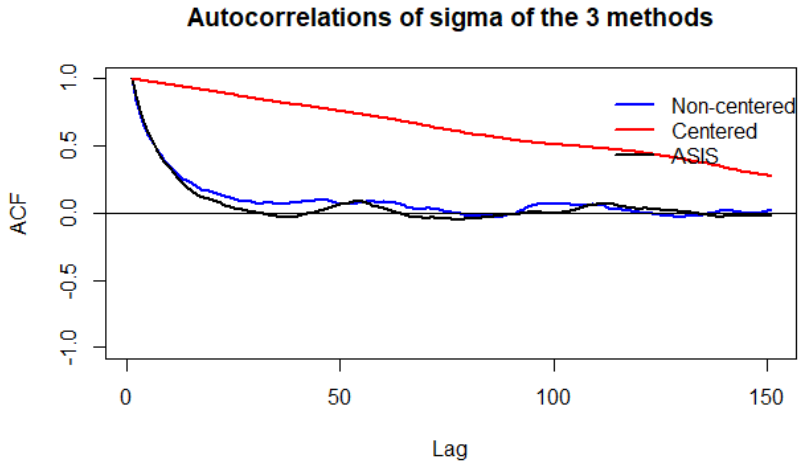
Regarding the convergence rates, Figure 12. shows the autocorrelations of the three parameters with the three estimation methods. ASIS always outperforms the centered and the non-centered methods.
Figure 12. shows us the pitfall of using only one parameterization. The centered method has a relatively fast convergence rate when estimating μ compared to the non-centered method. At the same time, the centered method falls short when sampling the persistence parameter ϕ and the standard deviation σ. We can see that the centered method has an extremely slow convergence rate for σ.
We would most probably opt for using the non-centered method if we would have to choose between them, but that would also mean that we would have to compromise with a slower convergence rate for the parameter μ in order to have faster rates for the other parameters.
With the ASIS method we do not have to make any compromises as it outperforms the other two methods for all parameters.



(a) Autocorrelation of μ



(b) Autocorrelation of ϕ



(c) Autocorrelation of σ

Figure 12: Difference in autocorrelations under different methods

6 Conclusion

In this thesis, the arising efficiency problems of two possible parameterizations of stochastic volatility models are presented. The goal of shifting from one parameterization to the other one is to improve any MCMC algorithms.
The ASIS method offers the possibility of using both parameterizations and avoid choosing one and face the consequences of possible efficiency problems.
ASIS interweaves the two parameterizations and updates jointly the latent states and the vector of parameters. This joint update preserves the link between them. This dependence often increase the likelihood of the proposed updates to land in a high posterior density area.
In Section 2. I presented an example adapted from Papaspiliopoulos et al. (2003) which presents how much the choice of the parameterization can affect the sampling efficiency and how easily can certain parameter values cause serious autocorrelation among sampled parameters.
In Section 5. I presented the power of ASIS through a simulation example where using the ASIS method outperformed both the centered and the non-centered parameterizations in terms of convergence rate.

References

[1] Bos, C. S.,2012, Relating stochastic volatility estimation methods. In L. Bauwens, C. Hafner and S. Laurent, eds., Handbook of Volatility Models and Their Applications, Wiley, 147-174.

[2] Brooks, S. P., 1998, Morkov Chain Monte Carlo Method and Its Application, Journal of Royal Statistical Society 47, 69-100.

[3] Carter, C. K. and Kohn, R., 1994, On Gibbs sampling for state space models, Biometrika 81, 541-553.

[4] Gelfand, A., Sahu, S. and Carlin, B., 1995, Efficient parametrisations for normal linear mixed models, Biometrika 82, 479-488.

[5] Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A. and Rubin D. B., 2014, Bayesian Data Analysis, Third Edition, Chapman & Hall.

[6] Harvey, A. C., Ruiz, E. and Shephard, N., 1994, Multivariate stochastic models, Review of Economic Studies 61, 247-264.

[7] Hull, J. and White, A., 1987, The pricing of options on assets with stochastic volatilities. The Journal of Finance 42, 281-300.

[8] Kastner, G.,2013,. stochvol: Efficient Bayesian inference for stochastic volatility (SV) models, R package version 0.5-1.

[9] Kastner, G. and Frühwirth-Schnatter, S.,2014, Ancillarity-Sufficiency Interweaving Strategy (ASIS) for boosting MCMC estimation of stochastic volatility models, Computational Statistics and Data Analysis 76, 408-423

[10] Kim, S., Shephard, N. and Chib, S.,1998, Stochastic volatility: Likelihood inference and comparison with ARCH models, Review of Economic Studies 65, 361-393.

[11] Lopes, H. F. and Tsay, R. S., 2011, Particle Filters and Bayesian Inference in Financial Econometrics, Journal of Forecasting 30, 168-209.

[12] Lopes, H. F., 2009, Monte Carlo Methods and Stochastic Volatility URL: http://hedibert.org/monte-carlo-methods-and-stochastic-volatility, Accessed: 2017-05-12.

[13] Mandelbrot, B., 1963, The Variation of Certain Speculative Prices, The Journal of Business 36, 394-419

[14] McCausland, W. J., Miller, S. and Pelletier, D. ,2011, Simulation smoothing for statespace models: A computational efficiency analysis, Computational Statistics and Data Analysis 55, 199212.

[15] Meng, X. L. and van Dyk, David, 1997, The EM Algorithm - an Old Folk-song Sung to a Fast New Tune, Journal of Royal Statistical Society 59, 551-567.

[16] Omori, Y., Chib, S., Shephard, N. and Nakajima, J., 2007, Stochastic volatility with leverage: Fast and efficient likelihood inference, Journal of Econometrics 140., 425-449.

[17] Papaspiliopoulos, O., 2003, Non-centered parameterizations for hierarchical models and data augmentation , Ph.D. dissertation, Department of Mathematics and Statistics, Lancaster University.

[18] Papaspiliopoulos, O., Roberts, G. and Sköld, M., 2003, Non-centered Parameterisations for hierarchical models and data augmentation, Bayes Statistics 7, 307-326.

[19] Petris, G., Petrone, S., and Campagnoli, P., 2007, Dynamic Linear Models with R, Springer, Berlin

[20] Platanioti, K., McCoy, E. j. and Stephens, D. A., 2005, A review of Stochastic Volatility: univariate and multivariate models, Imperial College London *working paper* .

[21] Pitt, M. K. and Shephard, N., 1999, Analytic convergence rates and parameterization issues for the Gibbs sampler applied to state space models, Journal of Time Series Analysis 20, 63-85.

[22] Roberts, G. O. and Sahu, S. K., 1997, Updating schemes, correlation structure, blocking and parameterization for the Gibbs sampler, Journal of the Royal statistical Society 59, 291-317.

[23] Taylor, S. J., 1986, Modelling Financial Time Series, John Wiley

[24] West, M. and Harrison, J., 1997, Bayesian Forecasting and Dynamic Models, Second Edition, Springer, New York

[25] Yu, Y. and Meng, X.-L., 2011, To center or not to center: that is not the questionan ancillarity suffiency interweaving strategy (ASIS) for boosting MCMC efficiency, Journal of Computational and Graphical Statistics 20, 531-570.

A Code

```
log_volatilities <- function(y,FF,G,mu,mr,a_1,R_1,sigma_sq,sr){
  # defining the needed variables
  t=length(y) # number of periods
  a=numeric(length=t)
  R=numeric(length=t)
  m=numeric(length=t)
  f=numeric(length=t)
  Q=numeric(length=t)
  C=numeric(length=t)
  A=numeric(length=t)
  x=numeric(length=(t-1))
  H=numeric(length=(t-1))
  mn=numeric(length=(t-1))
  Cm=numeric(length=(t-1))
  B=numeric(length=(t-1))
  # filling up the first elements
  a[1]=a_1
  R[1]=R_1
  f[1]=mr[1]+FF*a[1]
  Q[1]=FF*R[1]*FF + sr[1]
  A[1]=R[1]*FF/Q[1]
  m[1]=a[1]+A[1]*(y[1]-f[1])
  C[1]=R[1]-A[1]*Q[1]*A[1]
  # filling up the rest with a loop
  for (i in 2:t){
    a[i]=mu+G*m[i-1]
    R[i]=G*C[i-1]*G + sigma_sq
    f[i]=mr[i]+FF*a[i]
    Q[i]=FF*R[i]*FF + sr[i]
    A[i]=R[i]*FF/Q[i]
    m[i]=a[i]+A[i]*(y[i]-f[i])
    C[i]=R[i]-A[i]*Q[i]*A[i]
    B[i-1]=C[i-1]*G/R[i]
    H[i-1]=C[i-1]-B[i-1]*R[i]*B[i-1]
  }
  # now extracting the log volatilities with backward sampling
  h=numeric(length=t)
  h[t]=rnorm(1,m[t],sqrt(C[t]))
  x = numeric(length=(t-1))
  for (ii in (t-1):1){
    x[ii]=m[ii]+B[ii]*(h[ii+1]-a[ii+1])
    h[ii]=rnorm(1,x[ii],sqrt(H[ii]))
  }
  # returning the log volatilites
  return(h)
}


## Following the code of Hedibert Feitas Lopes
## URL: http://hedibert.org/monte-carlo-methods-and-stochastic-volatility/
## now accounting for the mixture of normal errors:
approx_log_vol = function(y,vol,FF,G,mu,a_1,R_1,sigma_sq){
  #Omori (2007) 10 piece approximation of a log chi^2:
```

```
  mr= c
      (1.92677,1.34744,0.73504,0.02266,-0.85173,-1.97278,-3.46788,-5.55246,-8.68384,-14.65000)
       #mean
  sr = c(0.11265,0.17788,0.26768,0.40611,0.62699,0.98583,1.57469,2.54498,4.16591,7.33342) #
      variance
  p = c(0.00609,0.04775,0.13057,0.20674,0.22715,0.18842,0.12047,0.05591,0.01575,0.00115) #
      probability
  #transforming the inputs
  y_star= log(y^2)
  sq_sr= sqrt(sr)
  n =length(y)
  r = numeric(length=n)
  eps =numeric(length=n)
  ## approximating the log chi^(2) distribution
  for (i in 1:n){
    eps[i] = y_star[i]- vol[i]  # log(eps^(2))
    weight= dnorm(eps[i],mr,sq_sr)*p
    r[i]= sample(1:10,size=1,prob=weight/sum(weight))
  }
  ## extracting the log-volatilities and the indicator vector
  hh = log_volatilities(y_star,FF,G,mu,mr[r],a_1,R_1,sigma_sq,sr[r])
  return(list(hh=hh,mr=mr[r],sr=sr[r]))
}

## multivariate normal density function
binorm_dens <- function(x,meaan,Sigma){
  1/(2*pi)*det(Sigma)^(-0.5)*exp(-0.5*t(x-meaan)%*%solve(Sigma)%*%(x-meaan))
}


## iteration steps for the centered method
for (ii in 1:M) {
  sigma_new = 1/rgamma(1, (length(h)-1)/2,C_T)
  ww=mvrnorm(1, b_T, (sigma_new*B_T))
  phi_new = ww[2]
  gamma_new = ww[1]

  if (phi_new<0.999999){
    R_11 = dnorm(h_vol_0,gamma_new/(1-phi_new),sqrt(sigma_new/(1-phi_new^(2))))*dnorm(gamma_
        new,b_mu*(1-phi_new),sqrt(B_mu)*(1-phi_new))*(1/(2*beta(a_zero,b_zero))*((1+phi_new)/
        2)^(a_zero-1)*((1-phi_new)/2)^(b_zero-1))*dgamma(sigma_new,0.5,0.5*B_sigma)
    R_12 = dnorm(h_vol_0,gamma_old/(1-phi_old),sqrt(sigma_sq_old/(1-phi_old^(2))))*dnorm(gamma
        _old,b_mu*(1-phi_old),sqrt(B_mu)*(1-phi_old))*(1/(2*beta(a_zero,b_zero))*((1+phi_old)/
        2)^(a_zero-1)*((1-phi_old)/2)^(b_zero-1))*dgamma(sigma_sq_old,0.5,0.5*B_sigma)
    R_21 = binorm_dens(x=c(gamma_old,phi_old),mean_param,(B_T*sigma_sq_old))*dinvgamma(sigma_
        sq_old,(length(h)-1)/2,C_T)
    R_22 = binorm_dens(x=c(gamma_new,phi_new),mean_param,(B_T*sigma_new))*dinvgamma(sigma_new
        ,(length(h)-1)/2,C_T)
    R = (R_11)/(R_12+.Machine$double.eps) * (R_21)/(R_22+.Machine$double.eps)
    accept_prob[ii] = min(1,R)
    if (runif(1)< accept_prob[ii]){
      sigma_sq_old =sigma_new
      phi_old = phi_new
      gamma_old=gamma_new
      accepted=accepted+1
    }
```

```
  }
  param_c[ii,]=c(gamma_old,phi_old,sigma_sq_old)
  listed<-approx_log_vol(yy,h,1.0,phi_old,gamma_old,0,1000,sigma_sq_old)
  h<-listed$hh #$
  h_vol_0 = rnorm(1,mu_dr+phi*(h[1]-mu_dr),sqrt(sigma_sq))
  h_T =c(h_vol_0,head(h,-1))  # h_{-T}  where last element is omitted but the initial value is
       kept
  X=cbind(1,h_T)     # Design matrix
  B_T = solve(t(X)%*%X + solve(B_0))
  b_T = B_T%*%t(X)%*%h
  C_T = 0.5*(sum(h^2)-t(b_T)%*%t(X)%*%h)

}

## iteration for the non-centered method
for (ii in 2:M) {
  phi_new = rnorm(1,mean_nc_phi,sd = st_nc_phi)
  if (phi_new<0.999999){
    R_11 = dnorm(h_tilde_0,0,sd=sqrt(1/(1-phi_new^2)))*(1/(2*beta(a_zero,b_zero))*((1+phi_new)
        /2)^(a_zero-1)*((1-phi_new)/2)^(b_zero-1))/dnorm(phi_new,mean_nc_phi,st_nc_phi)
    R_12 = dnorm(h_tilde_0,0,sd=sqrt(1/(1-phi_old^2)))*(1/(2*beta(a_zero,b_zero))*((1+phi_old)
        /2)^(a_zero-1)*((1-phi_old)/2)^(b_zero-1))/dnorm(phi_old,mean_nc_phi,st_nc_phi)
    R = R_11/R_12
    accept_prob_nc[ii] = min(1,R)
    if (runif(1)< accept_prob_nc[ii]){
      phi_old = phi_new
      accepted_nc=accepted_nc+1
    }
  }

  params_nc[ii,2]=phi_old
  zz <- mvrnorm(1,b_tilde_T,B_tilde_T)
    params_nc[ii,1]=zz[2]
    params_nc[ii,3]=zz[1]

  mu_old=params_nc[ii,1]
  sigma_nc_old=params_nc[ii,3]
 listed<-approx_log_vol(yy,h,1.0,phi_old,mu_old*(1-phi_old),0,1000,sigma_nc_old^2)
  h=listed$hh
  mr_t =listed$mr
  sr_t =listed$sr
  h_vol_0=rnorm(1,mu_old+phi_old*(h[1]-mu_old),sigma_nc_old)

  h_tilde_0 = (h_vol_0 -mu_old)/sqrt(sigma_nc_old^2)
  for (i in 1:length(h)) h_tilde[i]=(h[i]-mu_old)/sqrt(sigma_nc_old^2)
  numerator = h_tilde_0*h_tilde[1]
  for (i in 1:(length(h)-1)) numerator = numerator + h_tilde[i]*h_tilde[i+1]
  denominator = h_tilde_0^2
  for (i in 1:(length(h)-1)) denominator = denominator + h_tilde[i]^2
  mean_nc_phi = (numerator/denominator)
  st_nc_phi = (1/sqrt(denominator))
  for (tt in 1:n) y_hat[tt] = (log(yy[tt]^2)-mr_t[tt])/sqrt(sr_t[tt])

  for (tt in 1:n){
    X_tilde[tt,] = c(h_tilde[tt]/sqrt(sr_t[tt]),1/sqrt(sr_t[tt]))
  }
```

```
  B_tilde_T = solve(solve(B_tilde) + t(X_tilde)%*%X_tilde)
  b_tilde_T = B_tilde_T%*%(solve(B_tilde)%*%b_tilde + t(X_tilde)%*%y_hat)

}

## ASIS method
for (ii in 1:M){
  ## centered method
  sigma_new = 1/rgamma(1, (length(h)-1)/2,C_T)
  ww=mvrnorm(1, b_T, (sigma_new*B_T))
  phi_new = ww[2]
  gamma_new = ww[1]

  if (phi_new<0.999999){
    R_11 = dnorm(h_vol_0,gamma_new/(1-phi_new),sqrt(sigma_new/(1-phi_new^(2))))*dnorm(gamma_
        new,b_mu*(1-phi_new),sqrt(B_mu)*(1-phi_new))*(1/(2*beta(a_zero,b_zero))*((1+phi_new)/
        2)^(a_zero-1)*((1-phi_new)/2)^(b_zero-1))*dgamma(sigma_new,0.5,0.5*B_sigma)
    R_12 = dnorm(h_vol_0,gamma_old/(1-phi_old),sqrt(sigma_sq_old/(1-phi_old^(2))))*dnorm(gamma
        _old,b_mu*(1-phi_old),sqrt(B_mu)*(1-phi_old))*(1/(2*beta(a_zero,b_zero))*((1+phi_old)/
        2)^(a_zero-1)*((1-phi_old)/2)^(b_zero-1))*dgamma(sigma_sq_old,0.5,0.5*B_sigma)
    R_21 = binorm_dens(x=c(gamma_old,phi_old),mean_param,(B_T*sigma_sq_old))*dinvgamma(sigma_
        sq_old,(length(h)-1)/2,C_T)
    R_22 = binorm_dens(x=c(gamma_new,phi_new),mean_param,(B_T*sigma_new))*dinvgamma(sigma_new
        ,(length(h)-1)/2,C_T)
    R= (R_11)/(R_12+.Machine$double.eps) * (R_21)/(R_22+.Machine$double.eps)
    accept_prob[ii] = min(1,R)
    if (runif(1)< accept_prob[ii]){
      sigma_sq_old =sigma_new
      phi_old = phi_new
      gamma_old=gamma_new
      accepted=accepted+1
    }
  }
  param_half[ii,]=c(gamma_old,phi_old,sigma_sq_old)

  ## then transforming h to h_tilde with the drawn parameters:
  hh_tilde=(h-(param_half[ii,1]/(1-param_half[ii,2])))/sqrt(param_half[ii,3])
  hh_tilde_0 = (h_vol_0 -(param_half[ii,1]/(1-param_half[ii,2])))/sqrt(param_half[ii,3])

  numerator = hh_tilde_0*hh_tilde[1]
  for (i in 1:(length(h)-1)) numerator = numerator + hh_tilde[i]*hh_tilde[i+1]
  denominator = hh_tilde_0^2
  for (i in 1:(length(h)-1)) denominator = denominator + hh_tilde[i]^2
  mean_nc_phi = (numerator/denominator)
  st_nc_phi = (1/sqrt(denominator))


  phi_new = rnorm(1,mean_nc_phi,sd = st_nc_phi)
  if (phi_new<0.999999){
    R_11 = dnorm(hh_tilde_0,0,sd=sqrt(1/(1-phi_new^2)))*(1/(2*beta(a_zero,b_zero))*((1+phi_new
        )/2)^(a_zero-1)*((1-phi_new)/2)^(b_zero-1))/dnorm(phi_new,mean_nc_phi,st_nc_phi)
    R_12 = dnorm(hh_tilde_0,0,sd=sqrt(1/(1-phi_old^2)))*(1/(2*beta(a_zero,b_zero))*((1+phi_old
        )/2)^(a_zero-1)*((1-phi_old)/2)^(b_zero-1))/dnorm(phi_old,mean_nc_phi,st_nc_phi)
    R = R_11/R_12
    accept_prob_nc[ii] = min(1,R)
    if (runif(1)< accept_prob_nc[ii]){
```

```
      phi_old = phi_new
      accepted_nc=accepted_nc+1
    }
  }
  param_int[ii,2]=phi_old
  for (tt in 1:n){
    X_tilde[tt,] = c(hh_tilde[tt]/sqrt(sr_t[tt]),1/sqrt(sr_t[tt]))
  }
  #X_tilde
  y_hat = (y_log-mr_t)/sqrt(sr_t)
  B_tilde_T = solve(solve(B_tilde) + t(X_tilde)%*%X_tilde)
  b_tilde_T = B_tilde_T%*%(solve(B_tilde)%*%b_tilde + t(X_tilde)%*%y_hat)

  ## sampling mu and sigma:
  qq =mvrnorm(1,b_tilde_T,B_tilde_T)

  param_int[ii,1]<-qq[2]  ##mu
  param_int[ii,3]<-qq[1]  ## sigma square root


  gamma_old<-param_int[ii,1]*(1-param_int[ii,2])
  phi_old<-param_int[ii,2]
  sigma_sq_old<-param_int[ii,3]^2
  h=param_int[ii,1]+param_int[ii,3]*hh_tilde
  listed<-approx_log_vol(yy,h,1.0,phi_old,gamma_old,0,1000,sigma_sq_old)
  h=listed$hh
  h_vol_0= rnorm(1,gamma_old+phi*(h[1]),sqrt(sigma_sq_old))
  mr_t =listed$mr
  sr_t =listed$sr
  h_T =c(h_vol_0,head(h,-1))  # h_{-T}  where last element is omitted but the initial value is
       kept
  h_T
  X=cbind(1,h_T)    # Design matrix
  B_T = solve(t(X)%*%X + solve(B_0))
  b_T = B_T%*%t(X)%*%h
  C_T = 0.5*(sum(h^2)-t(b_T)%*%t(X)%*%h)

}

## code for centered method in Chapter 2.2.3

D_inv =matrix(0,nrow=n,ncol=n)
D_inv[1,1]=1
D_inv[n,n]=1
for (i in 2:n){
  D_inv[i,i]=1+psi^2
  D_inv[i-1,i]=-psi
  D_inv[i,i-1]=-psi
}
D_inv=(1/sig_n2)*D_inv
Sig=matrix(0,nrow=n,ncol=n)
D <- solve(D_inv)
Sig=sig_e2*diag(x = 1,nrow=n,ncol=n)+D
Sig_inv =solve(Sig)
V=matrix(0,nrow=n,ncol=n)
V=sig_e2 *Sig_inv %*% D
```

```
One=matrix(1,nrow=n,ncol=1)
for (iter in 1:niter){
  # Learning mu:
  b=(sig_e2)^(-1)*y +D_inv %*% One* mu_dr[iter]
  Meann =V%*%b
  # Learning w:
  for (k in 1:n){
    w[k] = rnorm(1,Meann[k],sqrt(V[k,k]))
  }

  # Learning mu:
  p = (n-1)*(1-psi)^2+(1-psi^2)
  q=(1-psi^2)*w[1]
  for (ii in 2:n) q=q +(1-psi)*(w[ii]-psi*w[ii-1])

  mu_dr[iter+1] =q/p+ rnorm(1,0,sqrt(sig_n2/p))

  # Storing the draws
  draws[iter,] = c(w,mu_dr[iter+1])
}

## code for non-centered method in Chapter 2.2.3.
D_inv =matrix(0,nrow=n,ncol=n)
D_inv[1,1]=1
D_inv[n,n]=1
for (i in 2:n){
  D_inv[i,i]=1+psi^2
  D_inv[i-1,i]=-psi
  D_inv[i,i-1]=-psi
}
D_inv=(1/sig_n2)*D_inv
Sig=matrix(0,nrow=n,ncol=n)
D <- solve(D_inv)
Sig=sig_e2*diag(x = 1,nrow=n,ncol=n)+D
Sig_inv =solve(Sig)
V=matrix(0,nrow=n,ncol=n)
V=sig_e2*Sig_inv %*% D
One=matrix(1,nrow=n,ncol=1)
mu_dr =numeric(length=(burnin+thin*M+1))
mu_dr[1]= 1.0 # flat prior for mu
for (iter in 1:niter){
  # Learning mu:
  b=(sig_e2)^(-1)*y +D_inv %*% One* mu_dr[iter]
  Meann =V%*%b - One*mu_dr[iter]
  # Learning h:
  for (k in 1:n){
    h[k] = Meann[k]+rnorm(1,0,sqrt(V[k,k]))
  }
  # Learning mu:
  y_m = mean(y)
  h_m = mean(h)
  mu_dr[iter+1] = y_m-h_m+rnorm(1,0,sqrt(sig_e2/n))
  # Storing the draws
  draws2[iter,] = c(h,mu_dr[iter+1])
}
```