

# Multivariate Stochastic Volatility Estimation with Sparse Grid Integration

## **Halil Erturk Esen**

Computational Science and Engineering Department, Informatics Institute, Istanbul Technical University, Istanbul, Turkey

Email: eesen@itu.edu.tr

Received 30 November 2015; accepted 16 February 2016; published 19 February 2016

Copyright © 2016 by author and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution-NonCommercial International License (CC BY-NC). http://creativecommons.org/licenses/by-nc/4.0/

Open Access

## Abstract

Multivariate stochastic volatility (MSV) models are nonlinear state space models that require either linear approximations or computationally demanding methods for handling the high dimensional integrals arising in the estimation problems of the latent volatilities and model parameters. Markov Chain Monte Carlo (MCMC) methods, which are based on Monte Carlo simulations using special sampling schemes, are by far the most studied method with several extensions and versions in previous stochastic volatility estimation studies. Exact nonlinear filters and particularly numerical integration based methods, such as the method proposed in this paper, were neglected and not studied as extensively as MCMC methods especially in the multivariate settings of stochastic volatility models. Filtering, smoothing, prediction and parameter estimation algorithms based on the sparse grid integration method are developed and proposed for a general MSV model. The proposed algorithms for estimation are compared with an implementation of MCMC based algorithms in a simulation study followed by an illustration of the proposed algorithms on empirical data of foreign exchange rate returns of US dollars and Euro. Results showed that the proposed algorithms based on the sparse grid integration method can be promising alternatives to the MCMC based algorithms especially in practical applications with their appealing characteristics.

## Keywords

Multivariate Stochastic Volatility, Nonlinear State Space, Sparse Grid Integration, Markov Chain Monte Carlo

# **1. Introduction**

Modeling, analysis, and estimation of volatilities of asset returns in financial markets have been a major research

area in the last three decades because of the significant conceptual role of volatility in mathematical and quantitative finance. Reliable volatility estimates of asset returns are indispensible inputs to various mathematical models in financial frameworks including but not limited to risk management, option pricing, portfolio and asset management. A considerably rich literature on volatility research describes the well studied stylized facts about volatility including the time-varying nature of volatility, leverage effects, volatility spillovers, heavy tails and long memory. Comprehensive treatment of financial volatility and discussions on the stylized facts can be found in [1]-[4].

Two main modeling approaches addressing the mentioned stylized facts are stochastic volatility (SV) and autoregressive conditionally heteroskedastic (ARCH) models. SV models differ from ARCH models by modeling the conditional volatility as a separate stochastic process having its own error term instead of explicitly modeling it. As discussed in [5], this stochastic approach makes the SV models flexible and versatile in capturing the mentioned stylized facts; however these advantages come with computational costs associated with the estimation methods. SV models are essentially nonlinear state space models requiring approximations or computationally demanding numerical methods for the associated estimation problems.

Several extensions on the univariate SV models addressing the mentioned stylized facts have been studied and proposed after the SV model of Taylor which dealt only with volatility clustering as described in [6] [7]. After the first multivariate stochastic volatility (MSV) model in the literature, given in [8], MSV modeling and estimation had become a rich research area composed of several model specifications addressing different and more complicated stylized facts, not only about volatilities but also about co-volatilities and their dynamics. [9] [10], [11] are among many of those works and a review can be found in [12]. While addressing the stylized facts and flexibility, another objective was keeping the complexity under control in those MSV modeling efforts since dimensionality brought additional complexity on top of the inherent complexity due to the nonlinearity in SV models. Being non-linear state space models, even univariate SV models require methods that can handle high dimensional integrals in obtaining smoothing, filtering and prediction estimates of time-varying volatilities and parameter estimates as mentioned in [13] [14]. Furthermore, an extra complexity is introduced in MSV models due to the dimensionality of latent volatilities.

Several studies incorporated algorithms, providing either fast or simplified approximations based on the wellknown Kalman filter and its extensions, using Laplace approximations, variations of moment matching and method of moments, and quasi likelihood methods. Although being fast and simple those methods generally suffered from poor performance as discussed in [2] [13]. Illustrations and examples of these methods can be found in [2] [4] [7] [8] [15] [16].

Difficulties with estimation of SV models quickly incited the usage of computationally intensive simulation based Monte Carlo methods for better estimations with advances in computational resources. Several Monte Carlo based algorithms incorporating resampling, particle filters, rejection sampling and importance sampling algorithms have been proposed with examples that can be found in [13] [14] [17] [18].

A major step in SV estimation was with studies such as [19]-[23] which introduced the Markov Chain Monte Carlo (MCMC) methods to the econometrics and SV fields. MCMC methods including the influential Metropolis-Hastings and Gibbs sampling algorithms quickly became central to the SV modeling and estimation studies, and an appreciable amount of literature on applications of different variations of MCMC methods was built up on several types of SV models, especially for the MSV models. Theoretically, MCMC methods are immune to the curse of dimensionality by construction, and they can easily be implemented in a Bayesian setting where the parameter estimation can also be handled without a maximization routine for the likelihood, hence without an explicit evaluation of the likelihood function as described in [23]. These appealing characteristics of MCMC methods made them a natural first choice in MSV estimation studies. However, MCMC algorithms are not flawless. They still require intense computational resources for complicated iterative sampling schemes for estimation. They are not exact as being Monte Carlo simulation based methods, and additional issues on error control and convergence are inherent particularly for MCMC methods.

Multidimensional integrals arising in estimation of SV models can be handled by classical numerical integration methods. Being exact methods with a deterministic nature, convergence properties of classical numerical integration methods are superior to probabilistic methods. However, when the problem dimension increases these methods become computationally infeasible since the number of dimensions increases the complexity of these type of algorithms exponentially. Unsurprisingly, studies on application of the numerical integration methods to nonlinear state space models and particularly to MSV models are quite rare compared to the approximation based methods and Monte Carlo simulation based methods including the MCMC. Refs. [14] and [24] are the main studies which discussed the application of numerical integration methods to nonlinear state space models.

Sparse grid integration method is a smartly reshaped version of classical numerical integration method to handle multidimensional integrals by constructing multi-dimensional integration formulas in a way that the dimensionality effect is decreased to a certain extent which allows practical implementation in high dimensional cases in contrast to the classical numeric integration methods. Sparse grid integration approach starts with [25] and detailed treatment of different approaches based on the idea is available in [26] [27]. The sparse grid integration approach was applied to some economics and financial problems (e.g., discrete choice analysis in [27], collateral mortgage optimization problem in [28], derivative and option pricing in [29] and asset liability in life insurance in [30]). However, estimation algorithms based on the sparse grid integration methods for SV models have been neither studied nor mentioned in the literature.

The goal of this paper is to show that the practical implementation of numerical integration method is possible by developing estimation algorithms incorporating the sparse grid integration methods as an alternative, and to depict that some important advantages of sparse grid integration methods over MCMC methods can be realized for MSV models. In this study, the sparse grid integration method is applied to the estimation problems of a given MSV model by developing density based estimation algorithms which include smoothing, filtering, prediction and parameter estimation incorporating the sparse grid integration method. Additionally, the estimation algorithms based on the proposed sparse grid integration method is illustrated on simulated and empirical data with comparisons.

The control of covariance matrices in MSV modeling and their handling in associated estimation algorithms are among the core topics of the MSV research. In this context, an approach which is new to the MSV modeling and estimation literature, to parameterize the correlation matrices, hence providing an alternative mechanism for handling covariance matrices in estimation, is also proposed and applied to the MSV model estimation algorithms in this paper. The approach is based on the slightly modified versions of the methods described in [31] [32] in a general setting.

The rest of this paper is organized as follows. A general multivariate stochastic volatility model and its state space representation are provided in Section 2. Development of estimation algorithms based on the proposed sparse grid integration method for the MSV model is presented and their properties are discussed in Section 3. An implementation of an MCMC algorithm for the MSV model is provided in Section 4. In Section 5, estimation algorithms with the proposed method are compared with the algorithms based on an MCMC implementation on simulated data. In Section 6, estimation algorithms with the proposed method are illustrated on empirical data. Finally, in Section 7 concluding remarks and further research directions are presented.

## 2. Multivariate Stochastic Volatility Model

The first MSV model in the literature is given in [8]. The specification of the basic MSV model based on this model starts with modeling the asset returns by

$$y_t = V_{y,t}^{1/2} \varepsilon_t, \quad t = 1, \cdots, T$$
, (1)

where  $\mathbf{y}_t = (y_{1t}, \dots, y_{pt})'$  is a *p*-dimensional vector of asset returns, and  $\boldsymbol{\varepsilon}_t = (\varepsilon_{1t}, \dots, \varepsilon_{pt})'$  is a *p*-dimensional vector of observation disturbances which have a zero mean and a correlation matrix,  $\boldsymbol{P}_{\varepsilon}$ , defining the correlations between asset returns. In Equation (1),  $\boldsymbol{V}_{y,t} = \text{diag}(\exp(h_{1,t}), \dots, \exp(h_{p,t}))$  is a diagonal matrix whose diagonal entries are the time-varying variances and their logarithms follow an AR(1) process,

$$\boldsymbol{h}_{t+1} = \boldsymbol{\gamma} + \boldsymbol{\varphi} \boldsymbol{h}_t + \boldsymbol{\eta}_t, \quad t = 1, \cdots, T , \qquad (2)$$

where,  $\mathbf{h}_{t} = (h_{1t}, \dots, h_{pt})'$  is the vector of unobserved log-volatilities,  $\mathbf{\eta}_{t} = (\eta_{1t}, \dots, \eta_{pt})'$  is a vector of disturbances with zero mean and a variance matrix  $\mathbf{V}_{\eta} = \text{diag}(\sigma_{1\eta}^{2}, \dots, \sigma_{p\eta}^{2})$ . In Equation (2),  $\gamma = (\gamma_{1}, \dots, \gamma_{p})'$  is the intercept parameter vector and  $\boldsymbol{\varphi} = \text{diag}(\varphi_{1}, \dots, \varphi_{p})$  is the diagonal matrix of persistence parameters of the AR(1) process which drives the time-varying log-volatilities.

In Equation (1), the error vector  $\boldsymbol{\varepsilon}_t = (\varepsilon_{1t}, \dots, \varepsilon_{Pt})'$  enters the model multiplicatively, thus the model is nonlinear. Being a time-varying parameter model and nonlinear in structure, multivariate stochastic volatility model given by Equation (1) and Equation (2) is a non-linear state space model in essence. Equation (1) is considered to be the observation or measurement equation and implies the one step conditional density of observations (*i.e.* asset returns) given by

$$p_{\mathbf{y}}(\mathbf{y}_t | \mathbf{h}_t, \mathbf{P}_{\varepsilon}), \quad t = 1, \cdots, T,$$
(3)

and Equation (2) is the state equation, and implies the one step conditional density of states (*i.e.*, log-volatilities) given by

$$p_h(\boldsymbol{h}_t | \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}), \quad t = 1, \cdots, T,$$
(4)

where, the initial step has an assumed density,  $p_h(\boldsymbol{h}_1 | \boldsymbol{h}_0, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_\eta) \sim N(\boldsymbol{\mu}_0, \boldsymbol{V}_0)$ .

Several types of extensions and alternatives of the basic multivariate stochastic volatility model given in Equation (1) and Equation (2) have been proposed and implemented in the literature addressing various stylized facts about volatilities and co-volatilities. Examples can be found in [2] [4] [8]-[12].

In this paper, the estimation algorithms with the proposed estimation algorithms and MCMC based estimation algorithms are implemented on the basic model given by Equation (1) and Equation (2) and this particular model will be referred as MSVb in the subsequent sections of this study.

## 3. Estimation with Sparse Grid Integration

#### 3.1. Preliminaries

Similar to classical numerical integration methods, sparse grid integration methods are also based on integration formulas which are simply represented by a set of function evaluation points and corresponding weights. These points and weights are then used to evaluate the integral of a given function with

$$I^{(d)}f = \int_{\Omega} f(\mathbf{x}) d\mathbf{x} \approx \sum_{i=1}^{N_{i}} \omega_{il} f(\mathbf{x}_{il}), \qquad (5)$$

where, d is the dimension, l is the level,  $x_{li}$  are the p-dimensional vectors representing points,  $\omega_i$  are the weights and  $N_l$  is the number of points of the integration formula which is represented by,

$$Q_l^{(d)}f = \sum_{i=1}^{N_l} \omega_{il} f\left(\mathbf{x}_{il}\right).$$
(6)

In Equation (5) and Equation (6), level  $l = 1, 2, \cdots$  determines the number of points,  $N_l$ , in the formula. Relationship between the level l and the number of points,  $N_l$  depends on the type of the formula. Further details on numeric integration and quadrature formulas can be found in [26], and [27].

In order to implement sparse grid integration to the MSVb model, sparse grid integration formula and the associated sparse grid is constructed first by the telescoping sum,

$$Q_l^{(p)} = \sum_{|\mathbf{k}|_1 \le l+p-1} \left( \Delta_{k_1}^{(1)} \otimes \dots \otimes \Delta_{k_p}^{(1)} \right), \tag{7}$$

where  $Q_l^{(p)}$  is an integration formula of dimension *p* and level *l*,  $\Delta_k^{(l)} = (Q_k^{(l)} - Q_{k-1}^{(l)})$  are the difference formulas obtained from a univariate integration formula having different levels,  $\mathbf{k} = (k_1, \dots, k_p)$  is a *p*-dimensional vector and  $|\cdot|_r$  is the *r*-norm operator. Here, Equation (7) simply constructs a *p*-dimensional integration formula from a given univariate integration formula of different levels. See [26]-[28] for details on the construction of regular sparse grids.

Classical construction of multivariate integration formulas differ in the approach used in Equation (7), that is, if  $|\mathbf{k}|_1 < l + p$ , is replaced by  $|\mathbf{k}|_{\infty} < l$  in Equation (7), then the complete tensor grid used in classical approach would be obtained. The main difference is that, the regular sparse grid constructed by Equation (7) involves  $O(N(\log N)^{p-1})$  degrees of freedom with N points in one dimension of the grid at the boundary under certain conditions, while classical multivariate numerical integration formula has an exponential dependence,  $O(N^{p})$ , on

the number of dimensions. As a result, although not completely, sparse grid method significantly loosens the dependence on number of dimensions for practical application. For details on error bounds and complexity of sparse grids, see [26]-[28].

Construction of the sparse grid starts with the selection of a univariate quadrature rule,  $Q_l^{(1)}$ , among many alternatives such as trapezoidal rule, rectangle rule, Simpson's rule, Clenshaw-Curtis and Gaussian quadrature rules including but not limited to Gauss-Hermite, Gauss-Legendre, and Gauss-Patterson. In this paper, the well-known iterated trapezoid rule for open interval is used:

$$Q_{l}^{(1)}f = \frac{1}{N_{l}+1} \left(\frac{3}{2}f\left(\frac{1}{N_{l}+1}\right) + \sum_{i=2}^{N_{l}-1}f\left(\frac{i}{N_{l}+1}\right) + \frac{3}{2}f\left(\frac{N_{l}}{N_{l+1}}\right)\right),\tag{8}$$

where the number of points for level *l* is  $N_l = 2^l - 1$ .

#### 3.2. Smoothing, Filtering and Prediction Algorithms

In this section, the sparse grid method is applied to the density based estimation algorithms of the state-space model of MSVb. In the development of density based estimation algorithms for the MSVb model, a similar approach for the univariate nonlinear state-space models described in [14] and [24] is followed.

Let  $Q_{l,t}^{(p)}$  be *p*-dimensional sparse grid quadrature rule constructed from the choice of univariate quadrature rule (*i.e.* trapezoid rule in our case),  $Q_{l}^{(1)}$ , of level *l*, at time *t*, and let  $G_{t} = \{1, 2, \dots, N_{G}\}$  be the set of sparse grid point indices where  $N_{l} = 2^{l} - 1$  for the trapezoid rule given in Equation (8). Then, each point  $i \in G_{t}$  in the grid is characterized with a *p*-dimensional vector  $\mathbf{c}^{i} = (c_{1}^{i}, \dots, c_{p}^{i})$  representing the grid coordinates of the points and corresponding weight  $\omega_{t}^{i}$ . The constructed sparse grid can be applied to the state space model of MSVb by providing suitable integration intervals for each dimension of the state vector,  $\mathbf{h}_{t} = (\mathbf{h}_{t_{t}}, \dots, \mathbf{h}_{p_{t}})$ , for  $t = 1, \dots, T$ . Once the integration intervals provided, grid coordinates of points,  $\mathbf{c}^{i}$ , can be converted to actual point coordinates  $\mathbf{h}_{t}^{i}$ . A method for identifying suitable integration intervals for the states,  $\mathbf{h}_{t}$ , is incorporating the estimates form an extended Kalman filter applied to the MSVb model and setting integration intervals as  $(\hat{\mathbf{h}}_{t|s} - v\hat{\mathbf{d}}_{t|s}, \hat{\mathbf{h}} + v\hat{\mathbf{d}}_{t|s})$ , where  $\hat{\mathbf{h}}_{t|s}$  is the state estimates and  $\hat{\mathbf{d}}_{t|s}$  is the vector composed of square roots of the diagonal elements of the covariance estimate,  $\hat{\mathbf{\Sigma}}_{t|s}$  (*i.e.*, standard deviation estimates of states) obtained with the extended Kalman filter, and v is a scalar. In this approach, appropriate Kalman filter algorithm is selected depending on the type of the estimation problem, namely filtering (s = t), smoothing (s = T), or prediction (s = T + L). See [4] [8] [14] [15] for different usages of the Kalman filter in SV estimation problems, and [4] and [14] for details and derivation of the Kalman filter estimation algorithms.

Conditional filtering density of states for the MSVb model can be obtained recursively by the following two-step algorithm:

$$p(\boldsymbol{h}_{t} | \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}) = \int p_{h}(\boldsymbol{h}_{t} | \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}) p(\boldsymbol{h}_{t-1} | \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}) d\boldsymbol{h}_{t-1},$$

$$p(\boldsymbol{h}_{t} | \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}) = \frac{p_{y}(\boldsymbol{y}_{t} | \boldsymbol{h}_{t}, \boldsymbol{P}_{\varepsilon}) p(\boldsymbol{h}_{t} | \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon})}{\int p_{y}(\boldsymbol{y}_{t} | \boldsymbol{h}_{t}, \boldsymbol{P}_{\varepsilon}) p(\boldsymbol{h}_{t} | \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}) d\boldsymbol{h}_{t}},$$
(9)

for  $t = 1, \dots, T$ . Equipped with the sparse grid points, weights, and set of their indices, the integrals in the filtering algorithm can be handled numerically as

$$p\left(\boldsymbol{h}_{t}^{i} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \approx \sum_{j \in G_{t-1}} p_{h}\left(\boldsymbol{h}_{t}^{i} \mid \boldsymbol{h}_{t-1}^{j}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right) p\left(\boldsymbol{h}_{t-1}^{j} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \omega_{t-1}^{j}, \quad i \in G_{t},$$

$$p\left(\boldsymbol{h}_{t}^{i} \mid \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right) = \frac{p_{y}\left(\boldsymbol{y}_{t} \mid \boldsymbol{h}_{t}^{i}, \boldsymbol{P}_{\varepsilon}\right) p\left(\boldsymbol{h}_{t}^{i} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right)}{\sum_{i \in G_{t}} p_{y}\left(\boldsymbol{y}_{t} \mid \boldsymbol{h}_{t}^{i}, \boldsymbol{P}_{\varepsilon}\right) p\left(\boldsymbol{h}_{t}^{i} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \omega_{t}^{i}}.$$
(10)

Starting with the filtering density obtained in Equation (10), *L*-step ahead prediction density can be recursively obtained by

$$p\left(\boldsymbol{h}_{t+L} \mid \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) = \int p_{h}\left(\boldsymbol{h}_{t+L} \mid \boldsymbol{h}_{t+L-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right) p\left(\boldsymbol{h}_{t+L-1} \mid \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \mathrm{d}\boldsymbol{h}_{t+L-1},$$
(11)

and using the sparse grid integration, approximated by

H. E. Esen

$$p\left(\boldsymbol{h}_{t+L}^{i} \mid \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) = \sum_{j \in G_{t+L-1}} p_{h}\left(\boldsymbol{h}_{t+L}^{i} \mid \boldsymbol{h}_{t+L-1}^{j}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right) p\left(\boldsymbol{h}_{t+L-1}^{j} \mid \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \omega_{t+L-1}^{j}, \quad i \in G_{t+L}.$$
(12)

Given the filtering density, smoothing density is obtained with the backward recursion,

$$p(\boldsymbol{h}_{t} | \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}) = p(\boldsymbol{h}_{t} | \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}) \int \frac{p(\boldsymbol{h}_{t+1} | \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}) p_{h}(\boldsymbol{h}_{t+1} | \boldsymbol{h}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta})}{p(\boldsymbol{h}_{t+1} | \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon})} d\boldsymbol{h}_{t+1},$$
(13)

and sparse grid integration approximation is given by

$$p\left(\boldsymbol{h}_{t}^{i} | \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) = p\left(\boldsymbol{h}_{t}^{i} | \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) p_{h}\left(\boldsymbol{h}_{t+1}^{j} | \boldsymbol{h}_{t}^{i}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right) \\ = p\left(\boldsymbol{h}_{t}^{i} | \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \sum_{i \in G_{t}} \frac{p\left(\boldsymbol{h}_{t+1}^{j} | \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) p_{h}\left(\boldsymbol{h}_{t+1}^{j} | \boldsymbol{h}_{t}^{i}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right)}{p\left(\boldsymbol{h}_{t+1}^{j} | \boldsymbol{Y}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right)} \omega_{t}^{i}, \quad j \in G_{t+1},$$

$$(14)$$

for  $t = T - 1, T - 2, \dots, 1$ .

Once the estimation density of interest is obtained with one of the recursive algorithms approximated by sparse grid integration, estimation of statistical properties such as the mean and the variance of a function,  $f(\mathbf{h}_r)$  can be computed by the following expectation and its approximation with sparse grid integration:

$$E\left[f\left(\boldsymbol{h}_{r}\right)|\boldsymbol{Y}_{s}\right]=\int f\left(\boldsymbol{h}_{r}\right)p\left(\boldsymbol{h}_{r}|\boldsymbol{Y}_{s},\boldsymbol{\gamma},\boldsymbol{\varphi},\boldsymbol{V}_{\eta}\right)d\boldsymbol{h}_{i}\approx\sum_{i\in G_{r}}f\left(\boldsymbol{h}_{r}^{i}\right)p\left(\boldsymbol{h}_{r}^{i}|\boldsymbol{Y}_{s},\boldsymbol{\gamma},\boldsymbol{\varphi},\boldsymbol{V}_{\eta}\right)\boldsymbol{\omega}_{r}^{i},$$
(15)

where, (r,s) = (T,T), (t,t), (t+L,t) for smoothing, filtering and prediction respectively.

Estimation algorithms derived so far, with the exception of smoothing, are all sequential (*i.e.*, on-line) algorithms which do not require all the past information at each time period; instead the estimations from the previous step are used. This is important and advantageous from the computational perspective especially in practical applications, because the computational burden in prediction and filtering can be split across time periods in contrast with batch-type algorithms.

A possible issue with the recursive algorithms with sparse grid integration is the accumulation of numerical errors during recursions, where controls and corrections satisfying the requirement,

$$\int p(\boldsymbol{h}_r | \boldsymbol{Y}_s) d\boldsymbol{h}_r = \sum_{i \in G_r} p(\boldsymbol{h}_r^i | \boldsymbol{Y}_s) \omega_r^i = 1$$
(16)

can be implemented easily with only an extra computational cost of recalculation of  $p(h_r | Y_s)$  as needed.

An advantage of the algorithms presented in this section based on the sparse grid integration is their computational parallelizability on multi-core and/or massively parallel architectures such as graphics processing units (GPUs). The recursive filtering algorithm given in Equation (10) can be effectively parallelized at each time period by assigning sparse grid nodes and corresponding computations to parallel processors with a simple load balancing strategy since they are not dependent to each other. This provides significant speed-up in implementation. Since the prediction and smoothing algorithms use the filtering algorithm, same speed-ups are effective for those as well in a parallel implementation.

### 3.3. Parameter Estimation

Parameter estimation for the MSVb model can be performed by maximizing the log-likelihood function given by

$$\log\left(p\left(\boldsymbol{Y}_{T} \mid \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right)\right) = \log\left(\prod_{t=1}^{T} \left(\int p_{y}\left(\boldsymbol{y}_{t} \mid \boldsymbol{h}_{t}, \boldsymbol{P}_{\varepsilon}\right) p\left(\boldsymbol{h}_{t} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \mathrm{d}\boldsymbol{h}_{t}\right)\right)$$

$$= \sum_{t=1}^{T} \log\left(\int p_{y}\left(\boldsymbol{y}_{t} \mid \boldsymbol{h}_{t}, \boldsymbol{P}_{\varepsilon}\right) p\left(\boldsymbol{h}_{t} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \mathrm{d}\boldsymbol{h}_{t}\right),$$
(17)

and approximated with sparse grid integration by

$$\log\left(p\left(\boldsymbol{Y}_{T} \mid \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right)\right) = \sum_{t=1}^{T} \log\left(\sum_{i \in G_{t}} p_{y}\left(\boldsymbol{y}_{t} \mid \boldsymbol{h}_{t}^{i}, \boldsymbol{P}_{\varepsilon}\right) p\left(\boldsymbol{h}_{t}^{i} \mid \boldsymbol{Y}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \omega_{t}^{i}\right),$$
(18)

with respect to the parameters  $\gamma$ ,  $\varphi$ ,  $V_{\eta}$ , and  $P_{\varepsilon}$ . Here, the log-likelihood function is actually the denominator of the update step of filtering algorithm given in Equation (10) and readily available as a byproduct of filtering algorithm. Therefore, obtaining the log-likelihood is also sequential as discussed in the Section 3.2 and no extra computational burden is incurred. Parallelizability advantage described in Section 3.2 is also effective for parameter estimation as it is based on filtering algorithm.

A challenge that must be sorted out in practical implementation of maximization routine for the parameter estimation is finding a way to handle the correlation matrix  $P_{\varepsilon}$ . Correlation matrix,  $P_{\varepsilon}$ , is a symmetric matrix with ones in the diagonal, off-diagonal entries having values in the interval (-1,1), and moreover it must be positive-definite. To handle the mentioned restrictions on the correlation matrix,  $P_{\varepsilon}$  and to be able to incorporate common non-linear optimization methods and tools in the parameter estimation problem, a parameterization approach of the correlation matrix,  $P_{\varepsilon}$ , which is a slightly different version of the parameterization described in [31]-[33] in a general setting, was proposed in this study.

In the proposed parameterization approach, positive definiteness of the correlation matrix is ensured by the relation,  $P_{\varepsilon} = BB'$  where the  $(p \times p)$  matrix **B** is

$$\boldsymbol{B} = \begin{bmatrix} b_{1,1} & b_{1,2} & b_{1,p} \\ b_{2,1} & & 0 \\ & & 0 & 0 \\ b_{p,1} & 0 & 0 & 0 \end{bmatrix}$$
(19)

and its entries are constructed with the relation,

$$b_{ij} = \begin{cases} \cos(\alpha_{ij}) \prod_{k=1}^{j-1} \sin(\alpha_{ik}), & \text{if } j \ge i \\ 0, & \text{otherwise,} \end{cases}$$
(20)

where  $i = 1, \dots, p, j = 1, \dots, p$  and  $0 \le \alpha_{ij} \le \pi$ . Let angles,  $\alpha_{ij}$ , be the entries of a  $(p \times p)$  matrix,

$$A = \begin{bmatrix} \alpha_{11} & \alpha_{p-1,1} & 0 \\ & 0 & 0 \\ \alpha_{1,p-1} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}.$$
 (21)

In matrix A, there are  $p \times (p-1)/2$  angles for a given  $(p \times p)$  correlation matrix,  $P_{\varepsilon}$ . The parameterization given in Equations (19)-(21) allow one to replace the parameter matrix,  $P_{\varepsilon}$ , with matrix A in the maximization routine since matrix A does not have complex restriction on its entries for common optimization methods. In the notations of the estimation algorithms developed so far,  $P_{\varepsilon}$  can be replaced by A.

The log-likelihood function obtained by Equation (18) can be maximized using a general purpose nonlinear optimization method. Well known Newton's method, derivative free search algorithms such as Nelder-Mead, or quasi-Newton methods such as Broyden-Fletcher-Goldfarb-Shanno (BFGS) are some of the examples. Using derivative free optimization methods can significantly improve computational performance. However, methods calculating or approximating the Hessian matrix, H, at the maximum log-likelihood readily provide the observed Fisher information matrix, I in accordance with

$$I(\tau) = -H(\tau) = \frac{\partial^2}{\partial \tau_i \partial \tau_j} l(\tau), \qquad (22)$$

where  $\tau$  is the vector of parameters where the elements of  $\gamma$ ,  $\varphi$ ,  $V_{\eta}$ , and A are stacked in order. Then, the standard errors for parameter estimates can be obtained easily from the inverse of the observed Fisher information matrix at the maximum log-likelihood,  $\Gamma^{1}(\tau_{\text{ML}})$ .

## 4. Estimation with Markov Chain Monte Carlo

MCMC methods are simulation based methods that are proved to be quite successful and that are probably the most preferred methods in SV modeling and estimation. There is a large collection of papers with extensions

and variations of MCMC methods for estimating SV models (e.g., [2] [9]-[11]).

MCMC methods directly sample from the conditional joint density of interest, which is often the conditional joint smoothing density of states, through dependent samples obtained with the help of a constructed Markov Chain so that the limiting probability is the conditional estimation density of interest.

For parameter estimation, two main approaches are available. First approach incorporates maximization of the log-likelihood in an expectation maximization (EM) algorithm or approximations with other Monte Carlo methods that use smoothing estimates obtained by the MCMC as described in [2] and [14]. Second approach is augmenting the parameter space to the state space and obtaining the estimations of parameters along with the smoothing estimates of states through MCMC sampling in a Bayesian setting without the need for explicitly calculating or approximating the log-likelihood and a maximization routine as described in [9]-[11] [23]. In this paper, the former approach is followed for implementation, that is, a MCMC algorithm which samples from the conditional joint smoothing density of states is implemented first and then an EM algorithm is implemented using the samples obtained by the MCMC.

Let,  $\boldsymbol{H}_{t} = \{\boldsymbol{h}_{1}, \boldsymbol{h}_{2}, \dots, \boldsymbol{h}_{t}\}$ , be the set of state vectors and  $\boldsymbol{Y}_{t} = \{\boldsymbol{y}_{1}, \boldsymbol{y}_{2}, \dots, \boldsymbol{y}_{t}\}$  be the set of observation vectors up to time *t*, where  $t = 1, \dots, T$ . If  $\boldsymbol{H}_{\setminus t} = \{\boldsymbol{h}_{1}, \dots, \boldsymbol{h}_{t-1}, \boldsymbol{h}_{t+1}, \dots, \boldsymbol{h}_{t}\}$  is defined as the set of state vectors up to time *T* excluding the state vector of time *t*, then sampling from the conditional joint smoothing density  $p(\boldsymbol{H}_{T} | \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon})$  can be performed with Gibbs sampling at each step sampling from the kernel density  $p(\boldsymbol{h}_{L} | \boldsymbol{H}_{i}, \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{n}, \boldsymbol{P}_{\varepsilon})$  where,

$$p(\boldsymbol{h}_{t} | \boldsymbol{H}_{\boldsymbol{\lambda}_{t}}, \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon})$$

$$\approx \begin{cases} p_{h}(\boldsymbol{h}_{t} | \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}) p_{h}(\boldsymbol{h}_{t+1} | \boldsymbol{h}_{t}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}) p_{y}(\boldsymbol{y}_{t} | \boldsymbol{h}_{t}, \boldsymbol{P}_{\varepsilon}), & \text{if } t = 1, \cdots, T-1, \\ p_{h}(\boldsymbol{h}_{t} | \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}) p_{y}(\boldsymbol{y}_{t} | \boldsymbol{h}_{t}, \boldsymbol{P}_{\varepsilon}), & \text{if } t = T. \end{cases}$$

$$(23)$$

Given the observations and parameters, sampling from this kernel N times for  $t = 1, \dots, T$  completes the MCMC algorithm and the joint smoothing density  $p(H_t | Y_T, \gamma, \varphi, V_\eta, P_\varepsilon)$  is obtained. Usually, the first M samples are discarded as burn-in samples depending on the mixing and convergence properties of the chain.

The kernel given in Equation (23) does not allow direct sampling since it is not in an analytically tractable form, thus Metropolis-Hastings algorithm with a choice of proposal density,  $p_*(\mathbf{h}_t)$  and Metropolis-Hasting weight,

$$w_{MH} = \begin{cases} \frac{p_h \left( \boldsymbol{h}_t^* \mid \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta} \right) p_h \left( \boldsymbol{h}_{t+1}^* \mid \boldsymbol{h}_t, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta} \right) p_y \left( \boldsymbol{y}_t \mid \boldsymbol{h}_t^*, \boldsymbol{P}_{\varepsilon} \right) p_* \left( \boldsymbol{h}_t \right)}{p_h \left( \boldsymbol{h}_t \mid \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta} \right) p_h \left( \boldsymbol{h}_{t+1} \mid \boldsymbol{h}_t, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta} \right) p_y \left( \boldsymbol{y}_t \mid \boldsymbol{h}_t, \boldsymbol{P}_{\varepsilon} \right) p_* \left( \boldsymbol{h}_t^* \right)}, & \text{if } t = 1, \cdots, T - 1\\ \frac{p_h \left( \boldsymbol{h}_t^* \mid \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta} \right) p_y \left( \boldsymbol{y}_t \mid \boldsymbol{h}_t^*, \boldsymbol{P}_{\varepsilon} \right) p_* \left( \boldsymbol{h}_t \right)}{p_h \left( \boldsymbol{h}_t \mid \boldsymbol{h}_{t-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta} \right) p_y \left( \boldsymbol{y}_t \mid \boldsymbol{h}_t^*, \boldsymbol{P}_{\varepsilon} \right) p_* \left( \boldsymbol{h}_t \right)}, & \text{if } t = T, \end{cases}$$

$$(24)$$

can be implemented for sampling. In the Metropolis-Hastings step, a candidate, generated from the proposal density,  $p_*(\mathbf{h}_i)$ , is accepted with probability  $\min(w_{MH}, 1)$  otherwise previous sample is kept as current sample. As a proposal density, several alternatives are available with examples in [14]. Following the same approach described in [14], a normal density based on the extended Kalman filter,

$$p_*\left(\boldsymbol{h}_{t|T}\right) = f_N\left(\boldsymbol{h}_t; \hat{\boldsymbol{h}}_{t|T}, \boldsymbol{v}\hat{\boldsymbol{\Sigma}}_{t|T}\right),$$
(25)

can be incorporated where  $\hat{h}_t$  and  $\hat{\Sigma}_t$  are the extended Kalman filter smoothing estimates of states and covariance matrices respectively and v is a scalar.

Filtering density is simply obtained by repeating the above procedure for smoothing T times, at each step replacing T by t. Once the filtering density at time T or equivalently smoothing density at time T is obtained, prediction density can be constructed through resampling with the following recursive procedure:

$$p\left(\boldsymbol{h}_{T+L} \mid \boldsymbol{Y}_{T}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}, \boldsymbol{P}_{\varepsilon}\right) \approx \frac{1}{N} \sum_{j=1}^{N} p\left(\boldsymbol{h}_{T+L} \mid \boldsymbol{h}_{j,T+L-1}, \boldsymbol{\gamma}, \boldsymbol{\varphi}, \boldsymbol{V}_{\eta}\right).$$
(26)

Parameter estimation incorporates the joint smoothing density of states given the observations and parameters,

 $p(\mathbf{H}_T | \mathbf{Y}_T, \mathbf{\gamma}, \boldsymbol{\varphi}, \mathbf{V}_{\eta}, \mathbf{P}_{\varepsilon})$ , obtained with the smoothing procedure above, in an expectation maximization algorithm, where at each iteration samples from,  $p(\mathbf{H}_T | \mathbf{Y}_T, \mathbf{\gamma}, \boldsymbol{\varphi}, \mathbf{V}_{\eta}, \mathbf{P}_{\varepsilon})$  are drawn and the expected log-likelihood function,

$$\begin{aligned} & \mathbf{E}_{H|Y,\gamma,\boldsymbol{\varphi},\boldsymbol{V}_{\eta},\boldsymbol{P}_{\varepsilon}} \left[ \log \left( p\left(\boldsymbol{H}_{T},\boldsymbol{Y}_{T} \mid \boldsymbol{\gamma},\boldsymbol{\varphi},\boldsymbol{V}_{\eta},\boldsymbol{P}_{\varepsilon} \right) \right) \right] \\ &= \mathbf{E}_{H|Y,\gamma,\boldsymbol{\varphi},\boldsymbol{V}_{\eta},\boldsymbol{P}_{\varepsilon}} \left[ \log \left( p_{h}\left(\boldsymbol{H}_{T} \mid \boldsymbol{\gamma},\boldsymbol{\varphi},\boldsymbol{V}_{\eta}\right) p_{y}\left(\boldsymbol{Y}_{T} \mid \boldsymbol{H}_{T},\boldsymbol{P}_{\varepsilon}\right) \right) \right] \\ &\approx \sum_{i=M+1}^{N} \log \left( p_{h}\left(\boldsymbol{H}_{T}^{i} \mid \boldsymbol{\gamma},\boldsymbol{\varphi},\boldsymbol{V}_{\eta}\right) p_{y}\left(\boldsymbol{Y}_{T} \mid \boldsymbol{H}_{T}^{i},\boldsymbol{P}_{\varepsilon}\right) \right), \end{aligned}$$
(27)

is calculated from those samples and maximized with respect to the parameters  $\gamma$ ,  $\varphi$ ,  $V_{\eta}$ , and  $P_{\varepsilon}$  followed by the next iteration starting with sampling again from  $p(H_T | Y_T, \gamma, \varphi, V_{\eta}, P_{\varepsilon})$  with the explained MCMC procedure for smoothing using the updated values of parameters.

In the implementation of maximization routine, the same approach for parameterization of the correlation matrix,  $P_{\varepsilon}$ , described in Equations (19)-(21) in Section 3.3 is used to ensure the restrictions on the correlation matrix.

Filtering and prediction are often hard tasks for MCMC methods, while the real advantage of MCMC is apparent in smoothing and parameter estimation problems because the filtering, prediction and parameter estimation with the above MCMC based estimation algorithms require construction of the smoothing density first, in contrast with the estimation approaches given in the Section 3.2. Furthermore, the observations,  $y_t$ , for all time periods  $t = 1, \dots, T$  are required to construct the smoothing density, hence for the filtering and prediction densities and parameter estimations too. Therefore, MCMC based estimation algorithms are called batch algorithms, whereas the estimation algorithms given in Section 3.2 were sequential (*i.e.* on-line) algorithms. Batch algorithms use all the information from the beginning to the last period, and all information are used again with the arrival of new information. Additionally, computational burden cannot be split across time periods which is a clear disadvantage for practical applications.

From the computational point of view, although being advantageous in serial implementations, MCMC based algorithms are generally not suitable for parallelization since they are batch algorithms producing dependent samples at each iteration where both temporal and spatial dependence exists. A limited parallelization is still possible in the implementation approach used in this paper though. Furthermore, parallelization in the implementations, which conduct parameter estimation through MCMC sampling in a Bayesian setting, is much harder.

## 5. Application on Simulated Data

This section illustrates the implementation of the proposed sparse grid integration method and compares the method with the implemented MCMC method on simulated data. The data are simulated using the MSVb model specifications with the following parameters:

$$\boldsymbol{\gamma} = \begin{pmatrix} -0.25 \\ -0.25 \\ -0.25 \end{pmatrix}, \quad \boldsymbol{\varphi} = \begin{pmatrix} 0.95 \\ 0.95 \\ 0.95 \end{pmatrix}, \quad \boldsymbol{V}_{\eta} = \begin{pmatrix} 0.04 & 0 & 0 \\ 0 & 0.04 & 0 \\ 0 & 0 & 0.04 \end{pmatrix}, \quad \boldsymbol{P}_{\varepsilon} = \begin{pmatrix} 1 & 0.6 & 0.6 \\ 0.6 & 1 & 0.6 \\ 0.6 & 0.6 & 1 \end{pmatrix},$$

which are close to typical values in previous empirical studies in the stochastic volatility literature. Using these parameters, observations  $y_t$ , and log-volatilities,  $h_t$ , are generated for T = 1,000 periods. The proposed sparse grid integration method and the MCMC method described in the previous sections are applied on the simulated data to find filtering, smoothing one-step prediction estimates of log-volatilities, and parameter estimates. This procedure is repeated R = 100 times to capture the statistics on estimations. The root mean squared error (RMSE) of the log-volatility estimates is calculated by

RMSE-h<sub>t|s</sub> = 
$$\frac{1}{T} \sum_{t=t_s}^{T} \sqrt{\frac{1}{R} \sum_{r=1}^{R} \left( \boldsymbol{h}_{t|s}^{(r)} - \boldsymbol{h}_{t}^{(r)} \right)^2}$$
 (28)

where  $\mathbf{h}_{t|s}^{(r)}$  is the log-volatility estimate obtained with the estimation method, and  $\mathbf{h}_{t}^{(r)}$  is the log-volatility from the simulated data where  $(s, t_{s}) = (t, 1)$  for filtering,  $(s, t_{s}) = (T, 1)$  for smoothing and  $(s, t_{s}) = (2, t-1)$  for

prediction. Similarly, the RMSE of the parameters are calculated by

RMSE-
$$\tau = \sqrt{\frac{1}{R} \sum_{r=1}^{R} (\hat{\tau}^{(r)} - \tau^{(r)})^2}$$
 (29)

where  $\hat{\tau}^{(r)}$  is the parameter estimated with the estimation method and  $\tau_t^{(r)}$  is the actual values of the parameters which are used in generating the simulated data.

Using the MCMC algorithm described in Section 4, N = 1,000,000 samples are generated and M = 200,000 samples are discarded as burn-in samples. In the sparse grid integration method, univariate trapezoid rule given in Equation (8) is implemented with level l = 7 which leads to 127 points in each dimension at the boundary and a total of 2815 points in the three-dimensional sparse grid at each time period.

The scalar used in identification of the integration intervals and the proposal density in the MCMC algorithm with the Kalman filter covariance estimates was taken as v = 6.0. For parameter estimation, Newton's method is used as the maximization routine in both methods.

To speed-up the computations, graphical processing unit (GPU) computing is used in the implementations of the sparse grid integration method algorithms and this provided a significant computational advantage since the estimation algorithms described in Section 3.2 are highly parallelizable.

**Table 1** shows the estimation results for the parameters, where the mean and the RMSE of parameter estimates are calculated from the simulation repeats using the formula in Equation (29). In general, parameter estimates are quite close in both methods when the means and true values are compared. However, the proposed sparse grid integration performs slightly better when RMSE values are considered with lower values for most of the parameters with exceptions  $\gamma_1$  and  $\sigma_1$ . The level of the integration formula used, l = 7, provided enough accuracy that can be compared with that of MCMC method with 1,000,000 samples in this simulation study. However, for different choices of parameters especially for higher persistence parameter values the minimum requirement for accuracy levels would change.

**Table 2** shows the RMSE of the estimated log-volatilities calculated by Equation (28) for comparison of latent log-volatility estimates generated by the two methods for filtering, smoothing and prediction problems. It can be observed that the RMSEs for all estimation problems are close with slightly better performance of sparse grid integration method in filtering and one-step prediction estimations while a better performance of MCMC method is observed in the smoothing estimates in **Table 2**.

Turice 1. Estimation results for parameters.							
	i	True	Sparse grid integration		МСМС		
		IIue —	Mean	RMSE	Mean	RMSE	
$\gamma_i$	1	-0.25	-0.237	0.072	-0.260	0.067	
	2	-0.25	-0.248	0.029	-0.261	0.071	
	3	-0.25	-0.245	0.034	-0.231	0.096	
$\varphi_i$	1	0.95	0.947	0.024	0.956	0.069	
	2	0.95	0.952	0.016	0.959	0.065	
	3	0.95	0.946	0.026	0.961	0.071	
$\sigma_{\eta i}$	1	0.2	0.209	0.045	0.206	0.040	
	2	0.2	0.191	0.042	0.207	0.054	
	3	0.2	0.205	0.037	0.214	0.067	
$ ho_{ij}$	12	0.6	0.605	0.053	0.592	0.059	
	13	0.6	0.594	0.041	0.590	0.073	
	23	0.6	0.606	0.039	0.592	0.058	

 Table 1. Estimation results for parameters.

<b>Table 2.</b> Root mean squared errors for estimated log-volatilities, $h_t$ .							
		RMSEs for sparse grid integration			RMSEs for MCMC		
	i	Filtering, $h_{t t}$	Smoothing, $h_{t T}$	Prediction, $h_{t+1 t}$	Filtering, $h_{t/t}$	Smoothing, $h_{t/T}$	Prediction, $h_{t+1/t}$
$h_i$	1	0.204	0.195	0.246	0.208	0.185	0.251
	2	0.196	0.186	0.238	0.199	0.179	0.245
	3	0.199	0.189	0.233	0.203	0.182	0.241

# **6. Empirical Application**

In this section, the proposed sparse grid integration method is applied to fit the MSVb model to foreignexchange rate series of Euro (EUR)/Turkish Lira (TRL) and US Dollar (USD)/Turkish Lira (TRL) from March 1, 2001 to September 30, 2015. Special days and holidays, when the market was closed, were excluded. In the resulting data, there are 3669 trading days for each series. The returns are defined by the log-difference of each series. The return series were first checked for AR(1) effects and those effects were removed from the series. Returns of two foreign-exchange rates are shown in **Figure 1** where the co-movement of the returns is observable.

The MSVb model was fitted to the data using the proposed sparse grid integration method where the trapezoid rule of level l = 8 is used as the basis univariate integration formula to construct the sparse grid as described in Section 3.1. The resulting two dimensional sparse grid has 1793 points for each time period with 255 points at each dimension. For identifying the integration intervals, extended Kalman filter is used and the integration interval for each time period t was set as described in Section 3.2, where v = 6.0 is used for the interval  $(\hat{h}_{t|s} - v\hat{d}_{t|s}, \hat{h} + v\hat{d}_{t|s})$ . For the parameter estimation, log-likelihood given in Equation (18) was maximized using

the Newton's method. The standard errors and confidence intervals on parameters are obtained from the Hessian produced by the Newton's methods as explained in Section 3.3.

**Table 3** summarizes the parameter estimation results on the foreign exchange data. The MSVb model estimated by the sparse grid integration method successively captures the high persistence of the log-volatilities ( $\varphi_1 = 0.971$ ,  $\varphi_2 = 0.971$ ) and the correlations between returns ( $\rho_{12} = 0.649$ ).

As it can be seen in **Table 4**, the MSVb model addresses the patterns of log-volatilities in the logarithms of squared returns better than the approximation with extended Kalman filter with lower RMSEs for all estimation types namely filtering, smoothing and one-step ahead predictions.

**Figure 2** shows the smoothing estimates of log-volatilities obtained by the sparse grid integration and the Kalman filter methods, along with the logarithms of squared returns as benchmark, where better coherence of sparse grid integration estimation patterns is visually supported.

## 7. Conclusions

The results obtained from the simulated data and empirical application show that the estimation algorithms based on the proposed sparse grid integration method perform well enough to be considered as an alternative estimation method for the MSV models. Sequential algorithmic nature, highly parallelizable structure, better error control and convergence properties of sparse grid integration based estimation algorithms, compared to the batch nature of MCMC method, their limited parallelizability capabilities, and convergence issues, make sparse grid integration based estimation algorithms duite promising, especially for practical applications. Moreover, for the estimation problems of filtering and prediction, sparse grid integration based algorithms achieve better performance than MCMC based algorithms in terms of both accuracy and computational requirements. Filtering and prediction are as important as smoothing and parameter estimation, especially in practical applications, so the proposed algorithms based on the sparse grid integration provide the required versatility.

Further research includes construction of sparse grid formulas from other univariate formulas such as Gaussian quadrature rules which may and probably be more effective and suitable to MSV model density structures. Another direction for improvement would be the construction strategy of the sparse grid; a dynamic construction of the sparse grid by adjusting the level of integration formula at each time step for better error control and





2001 2002 2003 2004 2005 2006 2007 2008 2009 2010 2011 2012 2013 2014 2015 Figure 2. Smoothing estimates of log-volatilities.

-16.0 -19.0 -22.0 -25.0 -28.0

Table 3. Estimated MSVb model parameters for EUR/TL and USD/TL returns.							
Parameter	Series	Mean	Standard error	95% interval			
γi	1	-0.289	0.027	[-0.342, -0.237]			
	2	-0.378	0.033	[-0.440, -0.314]			
$arphi_i$	1	0.971	0.0092	[0.954, 0.987]			
	2	0.962	0.0084	[0.946, 0.976]			
$\sigma_{\eta i}$	1	0.188	0.019	[0.154, 0.225]			
	2	0.184	0.023	[0.146, 0.229]			
$ ho_{ij}$	1,2	0.649	0.039	[0.572, 0.725]			

a. EUR/TL is Series 1 and USD/TL is Series 2.

Table 4. Comparison of MSVb model estimates using sparse-grid integration and extended Kalman filter.

		RMSEs for SV with sparse grid integration			RMSEs for extended Kalman filter		
	i	Filtering, $h_{t/t}$	Smoothing, $h_{t/T}$	Prediction, $h_{t+1/t}$	Filtering, $h_{t/t}$	Smoothing, $h_{t/T}$	Prediction, $h_{t+1/t}$
$h_i$	1	2.576	2.436	2.660	2.711	2.652	2.754
	2	2.583	2.442	2.687	2.691	2.644	2.751

computational efficiency can significantly improve the method from the algorithmic perspective. Hybrid approaches combining Monte Carlo based methods with sparse grid integration based methods for computational advantage and better convergence could lead to further research.

## References

- Xiao, L. and Abdurrahman, A. (2007) Volatility Modelling and Forecasting in Finance. In: Knight, J. and Satchell, S., Eds., *Volatility in the Financial Markets*, Elsevier, 1-45. <u>http://dx.doi.org/10.1016/b978-075066942-9.50003-0</u>
- [2] Shephard, N. and Andersen, T.G. (2008) Stochastic Volatility: Origins and Overview. Discussion Paper Series, University of Oxford Department of Economics, Oxford.
- [3] Poon, S.H. and Granger, C.J.W. (2003) Forecasting Volatility in Financial Markets: A Review. *Journal of Economic Literature*, **41**, 478-539. <u>http://dx.doi.org/10.1257/.41.2.478</u>
- [4] Tsay, R.S. (2002) Analysis of Financial Time Series: Financial Econometrics. John Wiley, New York. <u>http://dx.doi.org/10.1002/0471264105</u>
- [5] Ghysels, E., Harvey, A. and Renault, E. (1996) Stochastic Volatility. In: Maddala, G.S. and Rao, C.R., Eds., Handbook of Statistics (14) Statistical Methods in Finance, Elsevier, Amsterdam, 119-191. http://dx.doi.org/10.1016/s0169-7161(96)14007-4
- [6] Taylor, S.J. (1982) Financial Returns Modelled by the Product of Two Stochastic Processes—A Study of Daily Sugar Prices 1961-79. In: Anderson, O.D., Ed., *Time Series Analysis: Theory and Practice* 1, North-Holland, Amsterdam, 203-226.
- [7] Taylor, S.J. (1986) Modelling Financial Time Series. John Wiley, Chichester.
- [8] Harvey, A., Ruiz, E. and Shephard, N. (1994) Multivariate Stochastic Variance Models. *Review of Economic Studies*, 61, 247-264. <u>http://dx.doi.org/10.2307/2297980</u>
- [9] Asai, M. and McAleer, M. (2006) Asymmetric Multivariate Stochastic Volatility. *Econometric Reviews*, 25, 453-473. <u>http://dx.doi.org/10.1080/07474930600712913</u>
- [10] Asai, M. and McAleer, M. (2009) The Structure of Dynamic Correlations in Multivariate Stochastic Volatility Models. *Journal of Econometrics*, **150**, 182-192. <u>http://dx.doi.org/10.1016/j.jeconom.2008.12.012</u>
- [11] Ishihara, T. and Omori, Y. (2012) Efficient Bayesian Estimation of a Multivariate Stochastic Volatility Model with Cross-Leverage and Heavy-Tailed Errors. *Computational Statistics and Data Analysis*, 56, 3674-3689. http://dx.doi.org/10.1016/j.csda.2010.07.015
- [12] Jiang, G., Knight, J. and Rossi, P.E. (2005) Alternative Specifications of Stochastic Volatility Models. Working Paper,

University of Western Ontario, Ontario.

- [13] Watanabe, T. (1999) A Non-Linear Filtering Approach to Stochastic Volatility Models with an Application to Daily Stock Returns. *Journal of Applied Econometrics*, 14, 101-121. http://dx.doi.org/10.1002/(SICI)1099-1255(199903/04)14:2<101::AID-JAE499>3.0.CO;2-A
- [14] Tanizaki, H. (1997) Nonlinear and Nonnormal Filters Using Monte Carlo Methods. Computational Statistics & Data Analysis, 25, 417-439. <u>http://dx.doi.org/10.1016/S0167-9473(97)00016-9</u>
- [15] Harvey, A. and Shephard, N. (1996) Estimation of an Asymmetric Stochastic Volatility Model for Asset Returns. *Journal of Business and Economic Statistics*, 14, 429-434.
- [16] Galant, A.R., Hsieh, D.A. and Tauchen, G. (1997) Estimation of Stochastic Volatility Models with Diagnostics. *Journal of Econometrics*, 81, 159-192. <u>http://dx.doi.org/10.1016/S0304-4076(97)00039-0</u>
- [17] Carlin, B.P., Polson, N.G. and Stoffer, D.S. (1992) A Monte Carlo Approach to Nonnormal and Nonlinear State Space Modeling. *Journal of American Statistical Association*, 87, 493-500. http://dx.doi.org/10.1080/01621459.1992.10475231
- [18] Sandman, G. and Koopman, S.J. (1998) Estimation of Stochastic Volatility Models via Monte Carlo Maximum Likelihood. *Journal of Econometrics*, 87, 271-301. <u>http://dx.doi.org/10.1016/S0304-4076(98)00016-5</u>
- [19] Tanner, M.A. and Wong, W.H. (1987) The Calculation of Posterior Distributions by Data Augmentation. *Journal of the American Statistical Association*, 82, 528-550. <u>http://dx.doi.org/10.1080/01621459.1987.10478458</u>
- [20] Tierney, L. (1994) Markov Chains for Exploring Posterior Distributions. *The Annals of Statistics*, **22**, 1701-1762. http://dx.doi.org/10.1214/aos/1176325750
- [21] Chib, S. and Greenberg, E. (1995) Understanding the Metropolis-Hastings Algorithm. *The American Statistician*, **49**, 327-335.
- [22] Chib, S. and Greenberg, E. (1996) Markov Chain Monte Carlo Simulation Methods in Econometrics. *Econometric Theory*, **12**, 409-431. <u>http://dx.doi.org/10.1017/S0266466600006794</u>
- [23] Jacquier, E., Polson, N.G. and Rossi, P.E. (1994) Bayesian Analysis of Stochastic Volatility Models. *Journal of Business and Economic Statistics*, **12**, 371-417.
- [24] Kitagawa, G. (1987) Non-Gaussian State Space Modeling of Nonstationary Time Series. Journal of American Statistics Association, 82, 1032-1063.
- [25] Smolyak, S.A. (1963) Quadrature and Interpolation Formulas for Tensor Products of Certain Classes of Functions. Soviet Mathematics Doklady, 4, 240-243.
- [26] Heiss, F. and Winschel, V. (2006) Estimation with Numerical Integration on Sparse Grids. Technical Report, University of Munich, Munich.
- [27] Bungarts, H.J. and Griebel, M. (2004) Sparse Grids. Acta Numerica, 13, 147-270. http://dx.doi.org/10.1017/cbo9780511569975.003
- [28] Gerstner, T. and Griebel, M. (1998) Numerical Integration Using Sparse Grids. Numerical Algorithms, 18, 209-232. http://dx.doi.org/10.1023/A:1019129717644
- [29] Gerstner, T. (2007) Sparse Grid Quadrature Methods for Computational Finance. University of Bonn, Habilitation.
- [30] Holtz, M. (2010) Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance. Vol. 77, Springer Science & Business Media, Berlin.
- [31] Rebonato, R. and Jäckel, P. (2011) The Most General Methodology to Create a Valid Correlation Matrix for Risk Management and Option Pricing Purposes.
- [32] Kercheval, A.N. (2008) On Rebonato and Jackel's Parametrization Method for Finding Nearest Correlation Matrices. International Journal of Pure and Applied Mathematics, 45, 383-390.
- [33] Rapisarda, F., Brigo, D. and Mercurio, F. (2007) Parameterizing Correlations: A Geometric Interpretation. IMA Journal of Management Mathematics, 18, 55-73. <u>http://dx.doi.org/10.1093/imaman/dpl010</u>