Computational Statistics and Data Analysis 76 (2014) 408–423

Contents lists available at ScienceDirect

Computational Statistics and Data Analysis

journal homepage: www.elsevier.com/locate/csda

Ancillarity-sufficiency interweaving strategy (ASIS) for boosting MCMC estimation of stochastic volatility models

Gregor Kastner *, Sylvia Frühwirth-Schnatter

Institute for Statistics and Mathematics, WU Vienna University of Economics and Business, Augasse 2-6, 1090 Vienna, Austria

ARTICLE INFO

Article history:
Received 6 June 2012
Received in revised form 28 November 2012
Accepted 4 January 2013
Available online 14 January 2013

Keywords:
Markov chain Monte Carlo
Non-centering
Auxiliary mixture sampling
Massively parallel computing
State space model
Exchange rate data

ABSTRACT

Bayesian inference for stochastic volatility models using MCMC methods highly depends on actual parameter values in terms of sampling efficiency. While draws from the posterior utilizing the standard centered parameterization break down when the volatility of volatility parameter in the latent state equation is small, non-centered versions of the model show deficiencies for highly persistent latent variable series. The novel approach of ancillarity-sufficiency interweaving has recently been shown to aid in overcoming these issues for a broad class of multilevel models. It is demonstrated how such an interweaving strategy can be applied to stochastic volatility models in order to greatly improve sampling efficiency for all parameters and throughout the entire parameter range. Moreover, this method of "combining best of different worlds" allows for inference for parameter constellations that have previously been infeasible to estimate without the need to select a particular parameterization beforehand.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

Returns of financial and economic time series often exhibit time-varying volatilities. To account for this behavior, Taylor (1982) suggests in his pioneering paper to model the logarithm of the squared volatilities by latent autoregressive processes of order one. This specification, commonly referred to as the *stochastic volatility (SV)* model, presents itself as a competitive alternative to GARCH-type designs by modeling the volatilities non-deterministically. Also, it arises naturally as a discretization of continuous-time models frequently appearing in the mathematical finance literature (see e.g. Hull and White, 1987).

Following e.g. Jacquier et al. (1994) or Kim et al. (1998), observed log-returns are denoted $\mathbf{y} = (y_1, y_2, \ldots, y_T)'$ and the SV model is specified as

$$y_t = \mathrm{e}^{h_t/2}\epsilon_t, \tag{1}$$

$$h_t = \mu + \phi(h_{t-1} - \mu) + \sigma\eta_t, \tag{2}$$

where it is assumed that the i.i.d. standard normal innovations ϵ_t and η_s are independent for $t, s \in \{1, \ldots, T\}$. The unobserved process $\mathbf{h} = (h_0, h_1, \ldots, h_T)$ appearing in state equation (2) is usually interpreted as the latent time-varying *volatility process* with initial state distributed according to the stationary distribution, i.e. $h_0|\mu, \phi, \sigma \sim N\left(\mu, \sigma^2/(1-\phi^2)\right)$. From now on, we will refer to Eqs. (1) and (2) as the SV model in its *centered parameterization (C)*.

Simulation efficiency in state-space models can often be improved through *model reparameterization*. Papers related to this matter include Gelfand et al. (1995), Pitt and Shephard (1999), Frühwirth-Schnatter (2004), Roberts et al. (2004) and Frühwirth-Schnatter and Sögner (2008) and Strickland et al. (2008). The pioneering paper by Taylor (1982) as well as several other papers like Kim et al. (1998) or Liesenfeld and Richard (2006) consider a *partially non-centered parameterization*, where

* Corresponding author. Tel.: +43 1 31336 5593; fax: +43 1 31336 774.
E-mail addresses: gregor.kastner@wu.ac.at (G. Kastner), sylvia.fruehwirth-schnatter@wu.ac.at (S. Frühwirth-Schnatter).

0167-9473/$ – see front matter © 2013 Elsevier B.V. All rights reserved.
doi:10.1016/j.csda.2013.01.002

the level μ of h_t – which defines the scale of y_t – is shifted from the state equation (2) to the observation equation (1) by setting $\bar{h}_t = h_t - \mu$. Kim et al. (1998) compare both parameterizations within a Bayesian inference. They show that the partially non-centered parameterization leads to very high inefficiency when sampling μ and recommend choosing the centered parameterization in any case. Nevertheless, the centered parameterization has several disadvantages. First, inefficiency when drawing σ is still high; see e.g. Table 1 in Kim et al. (1998). Second, the conclusions are only valid if ϕ is close to one, which is commonly the case when the SV model is applied to capture conditional heteroskedasticity of observed financial times series. However, this is not necessarily true when the SV model is applied in more general contexts such as capturing conditional heteroskedasticity in latent variables or regression residuals.

For the purpose of this paper, the (fully) *non-centered parameterization (NC)*, given through

$$y_t \sim N\left(0, \omega e^{\sigma \tilde{h}_t}\right), \tag{3}$$

$$\tilde{h}_t = \phi \tilde{h}_{t-1} + \eta_t, \quad \eta_t \sim N(0, 1), \tag{4}$$

where $\omega = \mathrm{e}^{\mu}$, is of particular importance. The initial value of $\tilde{h}_0|\phi$ is once again drawn from the stationary distribution of the latent process, i.e. $\tilde{h}_0|\phi \sim N\left(0, 1/(1-\phi^2)\right)$. Note that $\tilde{h}_t = (h_t - \mu)/\sigma$. For a moderate parameter range where $\phi_{\text{true}} \in \{0.8, 0.9, 0.95\}$ and $\sigma_{\text{true}} \in \{0.2, 0.3, 0.4\}$, Strickland et al. (2008) illustrate that this type of non-centering typically yields lowest inefficiency factors when estimating stochastic volatility and stochastic conditional duration models with randomly sized block updating. Also, in similar contexts, there are several papers showing that MCMC sampling improves a lot by considering a non-centered version of a state space model; see e.g. Frühwirth-Schnatter (2004) and Frühwirth-Schnatter and Wagner (2010). These authors show that non-centering is especially useful if the error variance in the state equation is considerably smaller than the error variance in the observation equation. Pitt and Shephard (1999) show for linear Gaussian state space models that the speed of convergence in the centered parameterization decreases as $|\phi|$ increases when the signal to noise ratio is fixed.

No matter which parameterization is chosen, the likelihood in the SV model has an intractable form. Thus, Bayesian estimation commonly relies on sampling the latent states $\mathbf{h}$ and treat these as known for updating the parameters μ, ϕ, and σ. In their seminal paper, Jacquier et al. (1994) propose a *single-move* Metropolis–Hastings (MH) algorithm. Each individual h_t is sampled conditional on past and future, i.e. drawn from $p(h_t|\mathbf{h}_{[-t]}, \sigma, \phi, \mu, \mathbf{y})$, where $\mathbf{h}_{[-t]}$ denotes all elements of $\mathbf{h}$ except h_t. Due to the commonly high persistence of the latent process, Shephard and Kim (1994) note that the draws obtained from this sampler are also highly correlated and thus only slowly converging to the stationary distribution. Alternatively, Shephard and Pitt (1997) propose a *multi-move* sampler, where volatility blocks of random length are updated at a time, while Shephard (1994) and Omori et al. (2007) propose a method to draw directly from $p(\mathbf{h}|\sigma, \phi, \mu, \mathbf{y})$. This becomes possible through a normal mixture approximation of $\log(\epsilon_t^2)$ and requires forward filtering backward sampling (FFBS) methods (Carter and Kohn, 1994; Frühwirth-Schnatter, 1994; Durbin and Koopman, 2002). Within a more general Gaussian state-space framework, Rue (2001) and McCausland et al. (2011) propose sampling the latent volatilities through Cholesky-factorization of the precision matrix by exploiting its band-diagonal structure. We adopt this method to sample the latent volatilities "all without a loop" (AWOL). For a more extensive review of both Bayesian and non-Bayesian SV estimation methods, see Bos (2012).

The contribution of this paper is threefold. First, we explore the impact of alternative parameterizations for a wide parameter range including empirically plausible values and more extreme ones that can be relevant for applications of SV models within more general frameworks such as SV factor models or regression analysis. It turns out that simulation efficiency heavily depends on the true parameter values of the data generating process, thus no single "best" parameterization exists. Second, we provide a strategy to overcome this deficiency by *interweaving* C and NC utilizing an ancillarity-sufficiency interweaving strategy (ASIS) introduced by Yu and Meng (2011). This results in a robustly efficient sampler that always outperforms the more efficient parameterization with respect to all parameters at little extra cost in terms of design and computation. Third, we provide evidence that empirical sampling efficiency heavily depends on the realization of the data generating process by massively parallel simulation experiments.

The paper is structured as follows: Section 2 gives insight into the estimation procedure for each of the two selected parameterizations. Section 3 explains how ASIS can be applied in order to interweave these parameterizations. Extensive simulation results presented in Section 4 compare sampling efficiency for all parameters amongst the different parameterizations, Section 5 provides real-data results for several daily exchange rates, and Section 6 concludes.

2. Bayesian inference in the SV model

2.1. Prior distributions

To perform Bayesian inference, a prior distribution $p(\mu, \phi, \sigma)$ needs to be specified. For both parameterizations, we choose the same independent components for each parameter. The level $\mu \in \mathbb{R}$ is equipped with the usual normal prior $\mu \sim N\left(b_\mu, B_\mu\right)$. For the persistence parameter $\phi \in (-1, 1)$, we choose $(\phi + 1)/2 \sim \mathcal{B}(a_0, b_0)$ as in Kim et al. (1998), implying

$$p(\phi) = \frac{1}{2B(a_0, b_0)} \left(\frac{1+\phi}{2}\right)^{a_0 - 1} \left(\frac{1-\phi}{2}\right)^{b_0 - 1}. \tag{5}$$

Clearly, the support of this distribution is the unit ball and thus guarantees stationarity of the autoregressive volatility process. For the volatility of volatility $\sigma \in \mathbb{R}^+$, we choose $\sigma^2 \sim B_\sigma \cdot \chi^2_1 = \mathcal{G}(1/2, 1/2B_\sigma)$. Note that this specification differs from the commonly employed conjugate Inverse-Gamma prior $\sigma^2 \sim \mathcal{G}^{-1}(c_0, C_0)$ and is motivated by Frühwirth-Schnatter and Wagner (2010), who equivalently stipulate the prior for $\pm\sqrt{\sigma^2}$ to follow a centered normal distribution, i.e. $\pm\sqrt{\sigma^2} \sim N(0, B_\sigma)$. It turns out that this choice is less influential when the true volatility of volatility is small because σ is not bound away from zero a priori.

2.2. MCMC methodology

Observation equation (1) can easily be rewritten as

$$\tilde{y}_t = h_t + \log(\epsilon_t^2), \quad \epsilon_t \sim N(0, 1), \tag{6}$$

where $\tilde{y}_t$ denotes $\log y_t^2$, or the transformed de-meaned returns $\log(y_t - \bar{y})^2$, but might just as well be taken $\log((y_t - \bar{y})^2 + c)$ with a fixed offset constant $c = 10^{-3}$ as in Kim et al. (1998) or $\log(y_t^2 + c)$ with $c = 10^{-4}$ as in Omori et al. (2007) in order to avoid values equal to zero. Eq. (6) now takes the form of a linear but non-Gaussian state space model. Moreover, one can approximate the distribution of $\log(\epsilon_t^2)$ by a mixture of normal distributions, i.e. $\log(\epsilon_t^2)|r_t \sim N(m_{r_t}, s^2_{r_t})$. Here, $r_t \in \{1, \ldots, 10\}$ defines the mixture component indicator at time t, while m_{r_t} and $s^2_{r_t}$ denote mean and variance of the r_tth mixture component as tabulated in Omori et al. (2007). This representation allows rewriting (6) as a linear and conditionally Gaussian state space model,

$$\tilde{y}_t = m_{r_t} + h_t + \epsilon_t, \quad \epsilon_t \sim N(0, s^2_{r_t}), \tag{7}$$

where speedy MCMC sampling becomes possible in three steps.

Algorithm 1 (*AWOL Sampler*)**.** Choose appropriate starting values for the parameters μ, ϕ, σ and the indicators $\mathbf{r} = (r_1, r_2, \ldots, r_T)'$ – e.g. start with components with high weights – and repeat the following steps:

(a) Sample the latent volatilities AWOL by drawing from $\mathbf{h}_{[-0]}|\mathbf{y}, \mathbf{r}, \mu, \phi, \sigma^2$ or $\tilde{\mathbf{h}}_{[-0]}|\mathbf{y}, \mathbf{r}, \mu, \phi, \sigma^2$, respectively. The initial value is drawn from $h_0|h_1, \mu, \phi, \sigma^2$ or from $\tilde{h}_0|\tilde{h}_1, \phi$.
(b) Sample μ, ϕ, σ^2 via Bayesian regression.
  – For C, we investigate a 1-block sampler, drawing from $\mu, \phi, \sigma^2|\mathbf{h}$, a 2-block sampler, where σ^2 is drawn from $\sigma^2|\mathbf{h}, \mu, \phi$, while μ and ϕ are sampled jointly from $\mu, \phi|\mathbf{h}, \sigma^2$, and a 3-block sampler, where all parameters are individually drawn from the full conditionals. Due to non-conjugacy of the chosen priors, MH updates are used in all variants.
  – In NC, MH is needed only for updating ϕ by drawing from $\phi|\tilde{\mathbf{h}}$, while μ and σ^2 can be Gibbs-updated jointly from $\mu, \sigma^2|\mathbf{y}, \tilde{\mathbf{h}}, \mathbf{r}$ (2-block) or individually from $\mu|\mathbf{y}, \tilde{\mathbf{h}}, \mathbf{r}, \sigma^2$ and $\sigma^2|\mathbf{y}, \tilde{\mathbf{h}}, \mathbf{r}, \mu$ (3-block).
(c) Update the indicators $\mathbf{r}$ from $\mathbf{r}|\mathbf{y}, \mathbf{h}$ in C, or $\mathbf{r}|\mathbf{y}, \tilde{\mathbf{h}}, \mu, \sigma^2$ in NC, via inverse transform sampling.

2.3. Step (a): sampling the latent volatilities AWOL

Conditional on all other variables, the joint density for $\mathbf{h}$ (and $\tilde{\mathbf{h}}$) is multivariate normal. Due to the order-one autoregressive nature of the latent volatility process, this distribution can be written in terms of the tridiagonal precision matrix $\boldsymbol{\Omega}$, giving rise to sampling *all without a loop (AWOL)*. This method is employed in Rue (2001) and McCausland et al. (2011) and does not require the "end-user" to implement any loops—hence the name. Thus, it is very convenient in terms of implementation and fast in terms of computation. No FFBS methods are needed, there is no need to invert the tridiagonal precision matrix $\boldsymbol{\Omega}$ and it is fast due to the availability of band back-substitution already implemented in practically all widely used programming libraries.

In the centered parameterization, we draw from $\mathbf{h}_{[-0]}|\mu, \sigma, \phi, \mathbf{r}, \mathbf{y} \sim N_T\left(\boldsymbol{\Omega}^{-1}\mathbf{c}, \boldsymbol{\Omega}^{-1}\right)$ with

$$\boldsymbol{\Omega} = \begin{bmatrix} \frac{1}{s^2_{r_1}} + \frac{1}{\sigma^2} & \frac{-\phi}{\sigma^2} & 0 & \cdots & 0 \\ \frac{-\phi}{\sigma^2} & \frac{1}{s^2_{r_2}} + \frac{1+\phi^2}{\sigma^2} & \frac{-\phi}{\sigma^2} & \ddots & \vdots \\ 0 & \frac{-\phi}{\sigma^2} & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \frac{1}{s^2_{r_{T-1}}} + \frac{1+\phi^2}{\sigma^2} & \frac{-\phi}{\sigma^2} \\ 0 & \cdots & 0 & \frac{-\phi}{\sigma^2} & \frac{1}{s^2_{r_T}} + \frac{1}{\sigma^2} \end{bmatrix},$$

and

$$\mathbf{c} = \begin{bmatrix} \frac{1}{s^2_{r_1}}(\tilde{y}_1 - m_{r_1}) + \frac{\mu(1-\phi)}{\sigma^2} \\ \frac{1}{s^2_{r_2}}(\tilde{y}_2 - m_{r_2}) + \frac{\mu(1-\phi)^2}{\sigma^2} \\ \vdots \\ \frac{1}{s^2_{r_{T-1}}}(\tilde{y}_{T-1} - m_{r_{T-1}}) + \frac{\mu(1-\phi)^2}{\sigma^2} \\ \frac{1}{s^2_{r_T}}(\tilde{y}_T - m_{r_T}) + \frac{\mu(1-\phi)}{\sigma^2} \end{bmatrix}.$$

Analogously, in the noncentered case, we draw from $\tilde{\mathbf{h}}_{[-0]}|\mu, \sigma, \phi, \mathbf{r}, \mathbf{y} \sim N_T\left(\mathbf{\Omega}^{-1}\mathbf{c}, \mathbf{\Omega}^{-1}\right)$ with

$$\mathbf{\Omega} = \begin{bmatrix} \frac{\sigma^2}{s^2_{r_1}} + 1 & -\phi & 0 & \cdots & 0 \\ -\phi & \frac{\sigma^2}{s^2_{r_2}} + 1 + \phi^2 & -\phi & \ddots & \vdots \\ 0 & -\phi & \ddots & \ddots & 0 \\ \vdots & \ddots & \ddots & \frac{\sigma^2}{s^2_{r_{T-1}}} + 1 + \phi^2 & -\phi \\ 0 & \cdots & 0 & -\phi & \frac{\sigma^2}{s^2_{r_T}} + 1 \end{bmatrix},$$

and

$$\mathbf{c} = \begin{bmatrix} \frac{\sigma}{s^2_{r_1}}(\tilde{y}_1 - m_{r_1} - \mu) \\ \frac{\sigma}{s^2_{r_2}}(\tilde{y}_2 - m_{r_2} - \mu) \\ \vdots \\ \frac{\sigma}{s^2_{r_{T-1}}}(\tilde{y}_{T-1} - m_{r_{T-1}} - \mu) \\ \frac{\sigma}{s^2_{r_T}}(\tilde{y}_T - m_{r_T} - \mu) \end{bmatrix}.$$

For both parameterizations, this is accomplished by first computing the Cholesky decomposition $\mathbf{\Omega} = \mathbf{L}\mathbf{L}'$. Due to the band structure of $\mathbf{\Omega}$, this is computationally inexpensive and can either be implemented directly or via the LAPACK-routine `dpbtrf` (Anderson et al., 1999), to name but one of the many widely available (and thoroughly tested) linear algebra routines designed for this task. Note that only main diagonal and lower first off-diagonal elements of $\mathbf{L}$ will be nonzero. Next, we draw $\boldsymbol{\epsilon} \sim N_T(\mathbf{0}, \mathbf{I}_T)$ and then efficiently solve $\mathbf{L}\mathbf{a} = \mathbf{c}$ for $\mathbf{a}$ and $\mathbf{L}'\mathbf{h} = \mathbf{a} + \boldsymbol{\epsilon}$ for $\mathbf{h}$ by using band back-substitution instead of actually calculating $\mathbf{L}^{-1}$. Finally, the initial value can be sampled from $h_0|h_1, \mu, \phi, \sigma \sim N\left(\mu + \phi(h_1 - \mu), \sigma^2\right)$ in C and from $\tilde{h}_0|\tilde{h}_1, \phi \sim N(\tilde{h}_1\phi^2, 1)$ in NC.

2.4. Step (b)-C: sampling of μ, ϕ and σ in C

For sampling $\boldsymbol{\theta} = (\mu, \phi, \sigma^2)$, it is helpful to rewrite the conditional AR(1) model as a conditional regression model with the lagged latent variables as regressors,

$$h_t = \gamma + \phi h_{t-1} + \eta_t, \quad \eta_t \sim N\left(0, \sigma^2\right),$$

via $\gamma = (1-\phi)\mu$. Note that the implied conditional prior $p(\gamma|\phi)$ follows a normal distribution with mean $b_\mu(1-\phi)$ and variance $B_\mu(1-\phi)^2$. In this subsection, we will discuss three common blocking strategies for sampling $\boldsymbol{\theta}$.

For a *one block* update of $\boldsymbol{\theta}$, we use a single MH step. The posterior arising from an auxiliary regression model with conjugate priors is used as the proposal density:

$$p_{\text{aux}}(\boldsymbol{\theta}_{\text{new}}|\mathbf{h}) = p_{\text{aux}}(\gamma_{\text{new}}, \phi_{\text{new}}|\mathbf{h}, \sigma^2_{\text{new}})p_{\text{aux}}(\sigma^2_{\text{new}}|\mathbf{h}).$$

We choose $p_{\text{aux}}(\sigma^2) \propto \sigma^{-1}$ to denote the density of an auxiliary improper conjugate $\mathcal{G}^{-1}\left(-\frac{1}{2}, 0\right)$ prior, and $p_{\text{aux}}(\gamma, \phi|\sigma^2)$ to denote the density of an auxiliary conjugate $N_2\left(\mathbf{0}, \sigma^2\mathbf{B}_0\right)$ prior with $\mathbf{B}_0 = \text{diag}(B_0^{11}, B_0^{22})$. More specifically, $p_{\text{aux}}(\gamma|\sigma) \sim N\left(0, \sigma^2 B_0^{11}\right)$ and $p_{\text{aux}}(\phi|\sigma) \sim N\left(0, \sigma^2 B_0^{22}\right)$. In order to avoid collinearity problems when σ^2 is close to zero (and thus h_t almost constant for all t), we pick slightly informative variances, i.e. $B_0^{11} = 10^{12}$ and $B_0^{22} = 10^8$. This yields

$$\gamma, \phi|\mathbf{h}, \sigma^2 \sim N_2\left(\mathbf{b}_T, \sigma^2\mathbf{B}_T\right), \tag{8}$$

with $\mathbf{B}_T = (\mathbf{X}'\mathbf{X} + \mathbf{B}_0^{-1})^{-1}$ and $\mathbf{b}_T = \mathbf{B}_T\mathbf{X}'\mathbf{h}_{[-0]}$, where $\mathbf{X}$ is the $T \times 2$ design matrix with ones in the first column and $\mathbf{h}_{[-T]}$ in the second. The marginalized auxiliary posterior distribution for σ^2 is given through $\sigma^2|\mathbf{h} \sim \mathcal{G}^{-1}(c_T, C_T)$, with $c_T = (T-1)/2$ and $C_T = \frac{1}{2}\left(\sum_{i=1}^T h_i^2 - \mathbf{b}_T'\mathbf{X}'\mathbf{h}_{[-0]}\right)$.

The acceptance probability is given through $\min(1, R)$, with

$$R = \frac{p(h_0|\boldsymbol{\theta}_{\text{new}})p(\gamma_{\text{new}}|\phi_{\text{new}})p(\phi_{\text{new}})p(\sigma^2_{\text{new}})}{p(h_0|\boldsymbol{\theta}_{\text{old}})p(\gamma_{\text{old}}|\phi_{\text{old}})p(\phi_{\text{old}})p(\sigma^2_{\text{old}})} \times \frac{p_{\text{aux}}(\phi_{\text{old}}, \gamma_{\text{old}}|\sigma^2_{\text{old}})p_{\text{aux}}(\sigma^2_{\text{old}})}{p_{\text{aux}}(\phi_{\text{new}}, \gamma_{\text{new}}|\sigma^2_{\text{new}})p_{\text{aux}}(\sigma^2_{\text{new}})}.$$

In the *two-block* sampler, we draw the first block from the full conditional distribution $\gamma, \phi|\mathbf{h}, \sigma^2$ given in (8) and accept with probability $\min(1, R)$, where

$$R = \frac{p(h_0|\gamma_{\text{new}}, \phi_{\text{new}})p(\gamma_{\text{new}}|\phi_{\text{new}})p(\phi_{\text{new}})}{p(h_0|\gamma_{\text{old}}, \phi_{\text{old}})p(\gamma_{\text{old}}|\phi_{\text{old}})p(\phi_{\text{old}})} \times \frac{p_{\text{aux}}(\gamma_{\text{old}}, \phi_{\text{old}})}{p_{\text{aux}}(\gamma_{\text{new}}, \phi_{\text{new}})}.$$

In order to construct a suitable proposal for the – now full conditional – density $p(\sigma^2|\mathbf{h}, \mu, \phi)$, we again use the auxiliary conjugate prior $p_{\text{aux}}(\sigma^2) \propto \sigma^{-1}$, under which we straightforwardly obtain

$$\sigma^2|\mathbf{h}, \mu, \phi \sim \mathcal{G}^{-1}(c_T, C_T), \tag{9}$$

where $c_T = T/2$ and $C_T = \frac{1}{2}\left(\sum_{t=1}^T((h_t - \mu) - \phi(h_{t-1} - \mu))^2 + (h_0 - \mu)^2(1 - \phi^2)\right)$. The acceptance probability simplifies to $\min(1, R)$ with

$$R = \frac{p(\sigma^2_{\text{new}})}{p(\sigma^2_{\text{old}})} \times \frac{p_{\text{aux}}(\sigma^2_{\text{old}})}{p_{\text{aux}}(\sigma^2_{\text{new}})} = \exp\left\{\frac{\sigma^2_{\text{old}} - \sigma^2_{\text{new}}}{2B_\sigma}\right\}.$$

In the *three-block* sampler, each individual parameter is drawn from the full conditional distribution $\mu|\cdot$, $\phi|\cdot$, and $\sigma^2|\cdot$, respectively. Thus, σ^2 is drawn from (9). For sampling ϕ, we obtain a proposal from

$$\phi|\mathbf{h}, \gamma, \sigma^2 \sim N\left(\frac{\sum_{t=1}^T h_{t-1}h_t - \gamma\sum_{t=0}^{T-1} h_t}{\sum_{t=0}^{T-1} h_t^2 + 1/B_0^{22}}, \frac{\sigma^2}{\sum_{t=0}^{T-1} h_t^2 + 1/B_0^{22}}\right).$$

The acceptance probability is equal to $\min(1, R)$ with

$$R = \frac{p(h_0|\phi_{\text{new}}, \mu, \sigma^2)p(\phi_{\text{new}})}{p(h_0|\phi_{\text{old}}, \mu, \sigma^2)p(\phi_{\text{old}})} \times \frac{p_{\text{aux}}(\phi_{\text{old}}|\sigma^2)}{p_{\text{aux}}(\phi_{\text{new}}|\sigma^2)}.$$

For sampling γ from the full conditional posterior distribution, we obtain a proposal from

$$\gamma|\mathbf{h}, \phi, \sigma^2 \sim N\left(\frac{\sum_{t=1}^T h_t - \phi\sum_{t=0}^{T-1} h_t}{T + 1/B_0^{11}}, \frac{\sigma^2}{T + 1/B_0^{11}}\right)$$

and an acceptance probability equaling $\min(1, R)$ with

$$R = \frac{p(h_0|\gamma_{\text{new}}, \phi, \sigma^2)p(\gamma_{\text{new}}|\phi)}{p(h_0|\gamma_{\text{old}}, \phi, \sigma^2)p(\gamma_{\text{old}}|\phi)} \times \frac{p_{\text{aux}}(\gamma_{\text{old}}|\sigma^2)}{p_{\text{aux}}(\gamma_{\text{new}}|\sigma^2)}.$$

2.5. Step (b)-NC: sampling of μ, ϕ and σ in NC

In the noncentered parameterization, only ϕ is left in the state equation. To sample this parameter, we employ a flat auxiliary prior $p_{\text{aux}}(\phi) \propto c$, yielding the proposal

$$\phi|\tilde{\mathbf{h}} \sim N\left(\frac{\sum_{t=0}^{T-1}\tilde{h}_t\tilde{h}_{t+1}}{\sum_{t=0}^{T-1}\tilde{h}_t^2}, \frac{1}{\sum_{t=0}^{T-1}\tilde{h}_t^2}\right),$$

and an acceptance probability of $\min(1, R)$, where $R = p(\tilde{h}_0|\phi^{\text{new}})p(\phi^{\text{new}})/p(\tilde{h}_0|\phi^{\text{old}})p(\phi^{\text{old}})$.

For sampling μ and σ, one can straightforwardly rewrite the conditional observation equation (7) as a regression model with homoskedastic errors, i.e.

$$\breve{\mathbf{y}} = \mathbf{X}\begin{bmatrix}\mu\\ \sigma\end{bmatrix} + \boldsymbol{\epsilon}, \tag{10}$$

where $\boldsymbol{\epsilon} \sim N_K(\mathbf{0}, \mathbf{I}_K)$, and

$$\breve{\mathbf{y}} = \begin{bmatrix}(\tilde{y}_1 - m_{r_1})/s_{r_1}\\ \vdots \\ (\tilde{y}_T - m_{r_T})/s_{r_T}\end{bmatrix}, \qquad \mathbf{X} = \begin{bmatrix}\tilde{h}_1/s_{r_1} & 1/s_{r_1}\\ \vdots & \vdots \\ \tilde{h}_T/s_{r_T} & 1/s_{r_T}\end{bmatrix}.$$

The joint posterior distribution is again bivariate Gaussian with variance–covariance matrix $\mathbf{B}_T = (\mathbf{B}_0^{-1} + \mathbf{X}'\mathbf{X})^{-1}$ and mean $\mathbf{b}_T = \mathbf{B}_T(\mathbf{B}_0^{-1}\mathbf{b}_0 + \mathbf{X}'\breve{\mathbf{y}})$, where $\mathbf{b}_0 = (b_\mu, 0)'$ and $\mathbf{B}_0 = \text{diag}(B_\mu, B_\sigma)$ denote mean and variance of the joint prior density $p(\mu, \sigma)$, respectively.

Alternatively, one could sample both parameters from the full conditional posteriors (*three-block* sampling), yielding $\mu|\mathbf{y}, \tilde{\mathbf{h}}, \mathbf{r}, \sigma \sim N\left(b_{T,\mu}, B_{T,\mu}\right)$ with

$$b_{T,\mu} = B_{T,\mu}\left(\sum_{t=1}^{T}\frac{\tilde{y}_t - m_{r_t} - \sigma\tilde{h}_t}{s_{r_t}^2} + \frac{b_\mu}{B_\mu}\right), \qquad B_{T,\mu} = 1\Big/\left(\sum_{t=1}^{T}1/s_{r_t}^2 + \frac{1}{B_\mu}\right),$$

and $\sigma|\mathbf{y}, \tilde{\mathbf{h}}, \mathbf{r}, \mu \sim N\left(b_{T,\sigma}, B_{T,\sigma}\right)$ with

$$b_{T,\sigma} = B_{T,\sigma}\sum_{t=1}^{T}\frac{\tilde{h}_t(\tilde{y}_t - m_{r_t} - \mu)}{s_{r_t}^2}, \qquad B_{T,\sigma} = 1\Big/\left(\sum_{t=1}^{T}\frac{\tilde{h}_t^2}{s_{r_t}^2} + \frac{1}{B_\sigma}\right).$$

2.6. Step (c): sampling the indicators $\mathbf{r}$

We proceed exactly as Omori et al. (2007). Observing that $\tilde{y}_t - h_t = \epsilon_t^*$ with $\epsilon_t^* \sim N\left(m_{r_t}, s_{r_t}^2\right)$, one easily obtains the posterior probabilities $\mathbb{P}(r_t = k|\cdot)$ for $k \in \{1, \ldots, 10\}$ and $t \in \{1, \ldots, T\}$ according to

$$\mathbb{P}(r_t = k|\cdot) \propto \mathbb{P}(r_t = k)\frac{1}{s_k}\exp\left\{-\frac{(\epsilon_t^* - m_k)^2}{2s_k^2}\right\},$$

where $\mathbb{P}(r_t = k)$ denotes the mixture weights of the kth component. In our implementation, we do the calculations on a log-scale and normalize with respect to the maximum as required. The actual drawing is then conducted via inverse transform sampling. Note that due to $T \times 10$ exponential function calls, this step is computationally rather expensive but can easily be parallelized.

3. Interweaving C and NC by ASIS

To provide some intuition about the sampling efficiency in C, let $\phi = 0$ for a moment. This implies that the state equation (2) reduces to $h_t \sim N(\mu, \sigma^2)$ i.i.d. for all $t \in \{1, \ldots, T\}$. In this setting, $\mathbf{h}$ becomes more informative about μ when the conditional variance σ^2 gets smaller. Thus, *more* information is missing when treating $\mathbf{h}$ as latent data. Consequently, when sampling μ under the assumption that $\phi = 0$ and σ^2 is small, we expect C to be inefficient. On the other hand, if ϕ approaches 1, the latent process converges towards a random walk and $\mathbf{h}$ will be very uninformative about μ. Thus, only little information is lost when treating $\mathbf{h}$ as latent data and C has better chances to work fine. In NC, no major troubles are to be expected if $\phi = 0$, since the state equation (4) reduces to $\tilde{h}_t \sim N(0, 1)$ i.i.d. for all $t \in \{1, \ldots, T\}$, which is obviously independent of the value of σ. Thus, sampling μ and σ in the linearized equation (10) reduces to simple linear regression

with independent regressors. If, however, ϕ goes towards 1, we are prone to running into spurious regression problems. Certainly these arguments rely on massive oversimplification (e.g. by not taking into account the impact of the mixture approximation or spillover effects by inefficient proposal densities and different blocking strategies) and can only provide a faint idea of what is going on in the general case.

Nevertheless, due to the fact that in the context of the model at hand, the latent variables $\mathbf{h}$ in C form a sufficient statistic for μ and σ, while the transformed volatilities $\tilde{\mathbf{h}}$ in NC form an ancillary statistic for these parameters, there is hope that interweaving C and NC helps to increase sampling efficiency. Yu and Meng (2011) propose an *ancillary-sufficiency interweaving strategy (ASIS)* which, in certain situations, converges geometrically even when C and/or NC fail to do so. They explain this "seemingly magical property" by relating to Basu's theorem (Basu, 1955) on the independence of complete sufficient and ancillary statistics and show in a quite general context that the geometric convergence rate of the sampler interweaving $\mathbf{h}$ and $\tilde{\mathbf{h}}$ is always bound by $R(r_C r_{NC})^{1/2}$, where R is the maximal correlation between $\mathbf{h}$ and $\tilde{\mathbf{h}}$ in their joint posterior distribution $p(\mathbf{h}, \tilde{\mathbf{h}}|\mathbf{y})$ and r_C, r_{NC} denote the geometric rate of convergence of C and NC, respectively. This means that the rate of convergence of the interwoven sampler is mainly governed by the individual convergence rates and the posterior correlation R, implying that ancillary-sufficiency pairs of latent variables are likely to be good candidates for reducing sampling inefficiency. It is worth noting that the original ASIS notation Y_{obs} for the observed data directly transforms to $\mathbf{y}$ for the model at hand, while Y_{mis} – denoting the "missing" part of the data – equals $\mathbf{h}$.

The idea of interweaving is surprisingly simple. It is based on sampling the parameters in question – in our case μ, σ (and ϕ) – twice: once utilizing C and again utilizing NC. Ad hoc, it is not clear whether one should start C and redraw NC ("baseline C") or vice versa ("baseline NC"). We will discuss both strategies and assess their performance individually. Algorithm 2 describes the former, i.e. both the latent volatilities and the indicators are sampled *once with baseline C*, while the parameters μ, ϕ, σ are sampled *once in each parameterization* within each iteration of the sampler. It is termed "GIS-C", where the first three letters are borrowed from Yu and Meng (2011) and stand for *global interweaving strategy*. C simply denotes the fact that we use the centered baseline.

Algorithm 2 (*GIS-C*)**.** Choose appropriate starting values and repeat the following steps:

(a) Draw $\mathbf{h}$ (C).
(b) Draw μ, ϕ, σ (C).
(b*) Move to NC by the simple deterministic transformation $\tilde{h}_t = \frac{h_t - \mu}{\sigma}$ for all t.
(b**) Redraw μ, ϕ, σ (NC).
(b***) Move back to C by calculating $h_t = \mu + \sigma \tilde{h}_t$ for all t.
(c) Draw the indicators $\mathbf{r}$ (C).

The individual sampling steps are implemented exactly as described in Sections 2.3–2.6. Note that since ϕ is not involved in the reparameterization, in step (b**), one might as well redraw μ and σ only; the difference concerning sampling efficiency is however negligible. Also note that although additional sampling steps are introduced as (b*) to (b***), overall sampling time is only affected minimally because these steps are very cheap in terms of computation cost.

The sampler with noncentered baseline is of course very similar. As before, for each iteration the parameters μ, ϕ, and σ are sampled twice (once in C and once in NC), while the latent volatilities and the indicators are sampled in NC only.

Algorithm 3 (*GIS-NC*)**.** Choose appropriate starting values and repeat the following steps:

(a) Draw $\tilde{\mathbf{h}}$ (NC).
(b) Draw μ, ϕ, σ (NC).
(b*) Move to C by the simple deterministic transformation $h_t = \mu + \sigma \tilde{h}_t$ for all t.
(b**) Redraw μ, ϕ, σ (C).
(b***) Move back to NC by transforming back: $\tilde{h}_t = \frac{h_t - \mu}{\sigma}$ for all t.
(c) Draw the indicators $\mathbf{r}$ (NC).

To conclude, note that the strategy of interweaving is intrinsically different to *alternating* the parameterizations, for instance by (randomly) choosing one parameterization and running a complete MCMC cycle within that parameterization. Also, it is distinct from *compromising* between two parameterizations, e.g. by partial noncentering.

4. Simulation results

In order to assess simulation efficiency of our algorithms, we simulate data from the model specified in Eqs. (1) and (2). For the sake of simplicity and readability, μ_{true} is set to -10 for all runs. Results not reported here show that this choice is of minor influence. The parameters ϕ_{true} and σ_{true} vary on a $\{0, 0.5, 0.8, 0.9, 0.95, 0.96, 0.97, 0.98, 0.99\} \times \{0.5, 0.4, 0.3, 0.2, 0.1\}$ grid, resulting in 45 distinct parameter settings. This choice includes previously investigated and empirically plausible values; see e.g. Jacquier et al. (1994), Kim et al. (1998), Liesenfeld and Richard (2006) and Strickland et al. (2008). Moreover, the range is chosen to also include more extreme values that frequently arise when univariate SV is applied to capture conditional heteroskedasticity in latent variables such as factors or residuals of regression-type problems.

We repeat this exercise for 500 data sets and apply four sampling schemes (C, NC, GIS-C, GIS-NC) by using $M = 100\,000$ MCMC draws after a burn-in of 10 000 for each data set. Time series length is fixed to $T = 5000$, which corresponds to just above 20 years of daily data. Overall, this results in 90 000 chains of length 110 000, or a total of around 50 trillion latent instantaneous volatility draws. Nevertheless, due to parallel implementation of native C code on our local computer cluster using 500 cores, sampling can easily be done overnight. Throughout all simulations we use priors with means equaling the true values, more specifically $b_\mu = \mu_{\text{true}}$, $B_\mu = 10$, $a_0 = 40$, $b_0 = 80/(1 + \phi_{\text{true}}) - 40$, $B_\sigma = \sigma^2_{\text{true}}$, and starting values are set to true values to avoid values outside the stationary distribution after the burn-in period.

Computation of parallel MCMC chains for each parameter constellation was conducted on a cluster of workstations consisting of 44 IBM dx360M3 nodes with a total of 544 cores running R 2.15.1 (R Development Core Team, 2012) and OpenMPI 1.4.3 (Gabriel et al., 2004). For high-level-parallelization and parallel random number generation according to L'Ecuyer et al. (2002), the R packages `parallel` (part of R) and `snow` (Tierney et al., 2011) were used. Ex-post analysis and timing was done on a Laptop with a 2.67 GHz Intel i5 M560 CPU running the same R version. For the actual sampling, the R package `stochvol` (Kastner, 2013), available on CRAN, was created. The core implementation is written in C, interfaced to R via `Rcpp` (Eddelbuettel and François, 2011). Inefficiency factors and effective sample sizes were computed with the R package `coda` (Plummer et al., 2006).

The mean time for running 1000 simulation draws varies between 2.3 s for C and 2.4 s for GIS-NC on a Laptop with a 2.67 GHz Intel i5 M560 CPU using one core. Note that these numbers are fairly constant for all true parameter values and grow linearly with T. As an example, the time to run 1000 simulations for $T = 500$ varies between 0.23 and 0.24 s.

4.1. To center or not to center?

Simulation efficiency of the two raw parameterizations mainly depends on the values of the parameters ϕ (persistence) and σ (volatility of volatility). To illustrate the latter, Fig. 1 shows autocorrelations of an exemplary parameter setup with small volatility of volatility $\sigma_{\text{true}} = 0.1$ for a single time series that has been randomly selected from the pool of all 500 time series. Here, C "fails" in the sense that the draws from $p(\mu|\mathbf{y})$ and $p(\sigma|\mathbf{y})$ exhibit large autocorrelation, while NC performs substantially better. This observation is in line with findings of Pitt and Shephard (1999) and Frühwirth-Schnatter (2004), who observe that simulation efficiency in the centered parameterization decreases with decreasing σ_{true} for linear Gaussian state space models.

On the other hand, Fig. 2 portraits a parameter setup with larger volatility of volatility $\sigma_{\text{true}} = 0.5$, while persistence ϕ_{true} and level μ_{true} are the same as before. Here, we see that draws from C show little autocorrelation, while MCMC chains obtained from NC do not mix well.

4.2. Sampling efficiency

For assessing simulation efficiency, the inefficiency factor (IF) is employed as a benchmark. It is an estimator for the integrated autocorrelation time τ of a stochastic process given through $\tau = 1 + 2\sum_{s=1}^{\infty} \rho(s)$, where $\rho(s)$ is the autocorrelation function for lag s. We estimate τ through the spectral density of the Markov chain, i.e. $\text{IF} = \gamma_0/s^2$, where γ_0 denotes the estimated spectral density evaluated at zero and s^2 denotes the sample variance of the MCMC draws. The inefficiency factor is directly proportional to the squared Monte Carlo standard error MCSE^2 through the relationship $\text{MCSE}^2 = (s^2/M) \times \text{IF}$. In other words, 100 000 draws from a Markov chain with an IF of 100 have roughly the same MCSE as 1000 draws from an independent sample. Consequently, the effective sample size ESS is given by M/IF. Clearly, the aim is to provide samplers with small IFs, thus large ESSs, at smallest possible computational cost.

Even for artificially created data sets of length $T = 5000$ or larger, estimation results may depend substantially on the actual realization of the underlying process. Also, other factors – most importantly the initial seed for drawing pseudo random variables in the individual MCMC steps – can influence both sample statistics from the posterior distribution as well as sample statistics for evaluating simulation efficiency. To compensate for this fact, we repeat each simulation with 500 independently generated artificial data sets. The boxplots provided in Figs. 3 and 4 illustrate the variation of IFs for the same parameter constellations as above. While the overall picture about non-centering remains the same, we can now observe some substantial deviation from the median for certain realizations. Furthermore, the plots show that both interweaving strategies GIS-C and GIS-NC help to avoid woe by working well no matter which raw parameterization fails.

4.3. Efficiency overview

In order to gain insight into the entire parameter range of interest, Tables 1, 3 and 5 provide a summary of median inefficiency factors across all 45 parameter constellations.

Median IFs obtained from draws from $p(\mu|\mathbf{y})$ in Table 1 confirm clearly that the centered parameterization is quite capable of efficiently estimating the level μ of the latent process throughout a wide parameter range, no matter which blocking strategy is used. Only a combination of both small σ_{true} and small ϕ_{true} leads to large inefficiency. As was to be expected, this is exactly the area where the non-centered parameterization performs comparably well; median IFs are small

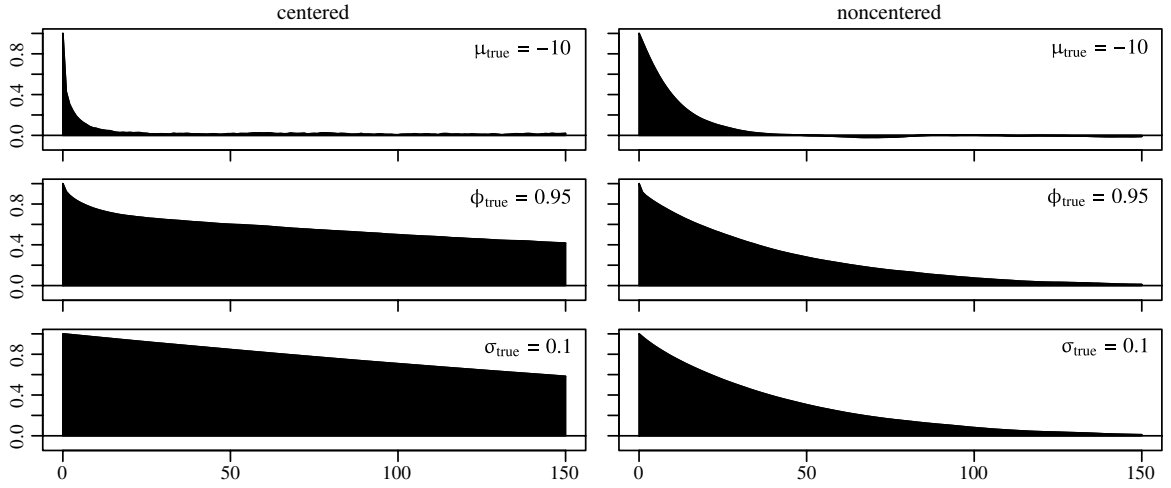

Fig. 1. Sample autocorrelations of 100 000 MCMC draws obtained from C (left hand side) and NC (right hand side) for a *small volatility of volatility* setup.

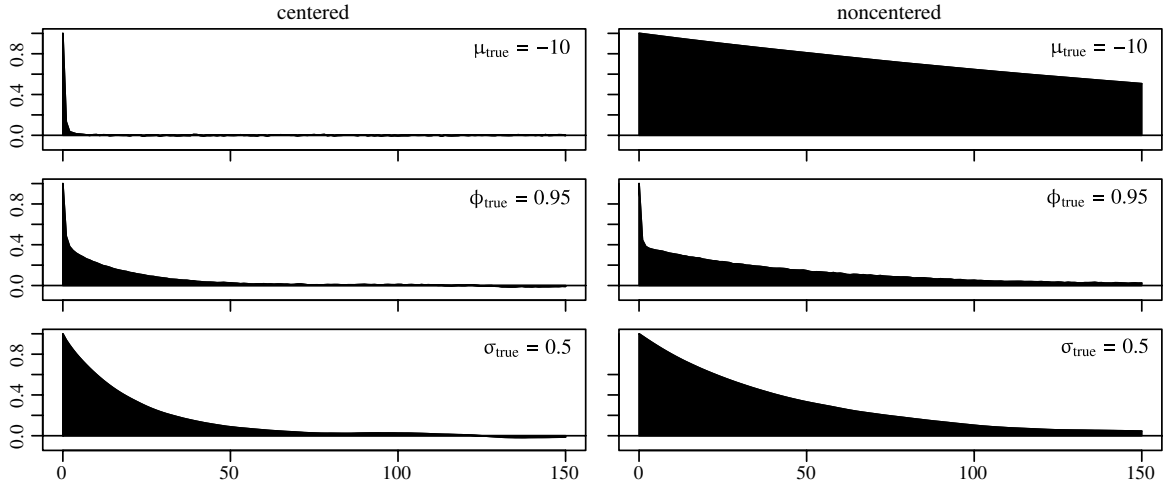

Fig. 2. Sample autocorrelations of 100 000 MCMC draws obtained from C (left hand side) and NC (right hand side) for a *large volatility of volatility* setup.

to moderately large. On the other end of the scale – where we find both highly persistent and highly varying latent variables – NC becomes close to useless with very large IFs of 1000 and above for both blocking strategies.

The lower three panels of Table 1 show the performance of the interwoven samplers with different baselines, GIS-C and GIS-NC. It stands out that in terms of simulation efficiency, both variants are always better than or en par with the ideal parameterization, while there are practically no differences between the sampler with baseline C and the one with baseline NC. Comparing CPU time of the raw samplers with their interwoven counterparts reveals the computational cost of interweaving, which amounts to merely around 2% in our setup. Thus, even when taking into account the extra cost, interweaving is hardly ever a bad choice. Also note that GIS-C is practically as fast as NC.

Table 2 shows a direct comparison of the interwoven sampler with the raw parameterizations in terms of increase in effective sample size. All numbers are positive, showing that interweaving is more efficient than the ideal parameterization, but sometimes only slightly. Note that in comparison to the suboptimal parameterization, GIS is always at least twice as effective.

Next, we turn to assessing simulation efficiency for the persistence parameter ϕ, summarized in Table 3. Because ϕ is not involved in the reparameterization, the differences between C and NC (and consequently also between the raw and the interwoven samplers) are much less pronounced. For a summary of efficiency gains, see Table 4. It stands out that one- and two-block samplers show very similar IFs, whereas the three-block sampler deteriorates due to massive overconditioning for moderate and small ϕ_{true} or σ_{true}. Note however that again the interwoven sampler is exempt from these defects due to the fact that NC performs solidly. Results not reported here show that for shorter time series with $T = 500$, sampling

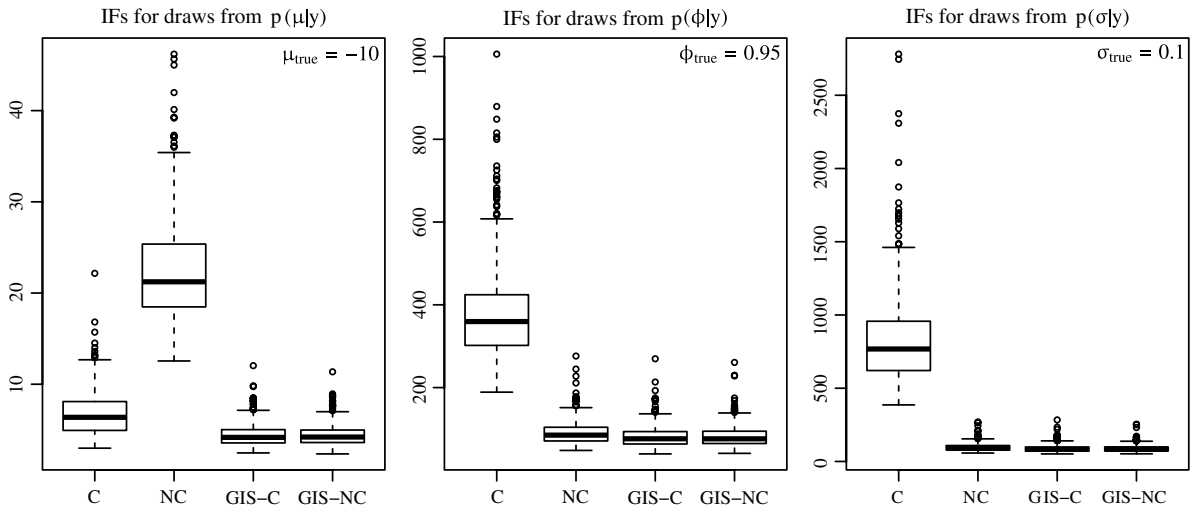

Fig. 3. Boxplots of 500 repeated measurements of inefficiency factors of 100 000 draws from the marginal densities. The underlying latent volatility process exhibits *small volatility of volatility*.

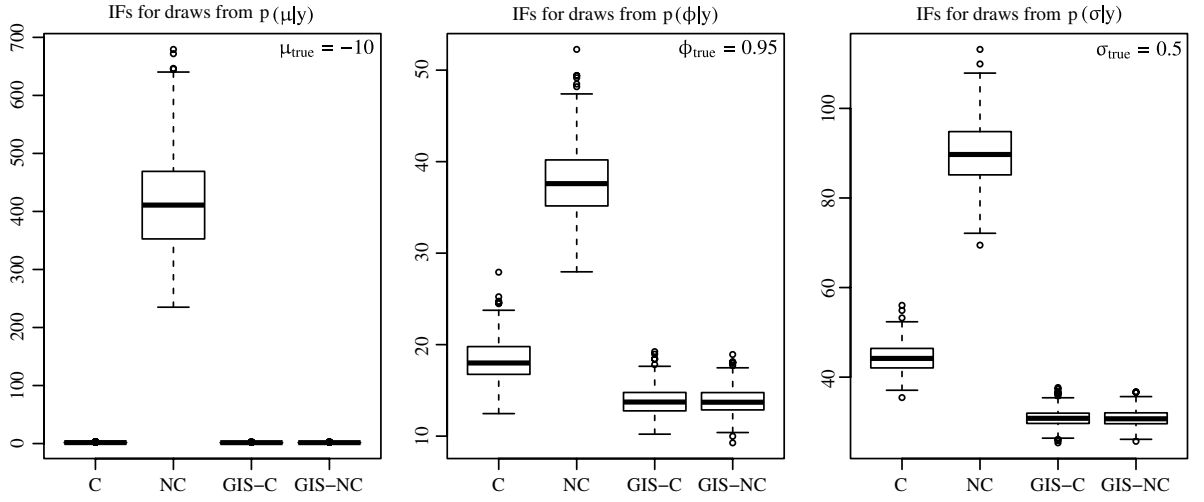

Fig. 4. Boxplots of 500 repeated measurements of inefficiency factors of 100 000 draws from the marginal densities. The underlying latent volatility process exhibits *large volatility of volatility*.

inefficiency is uniformly smaller for all parameterizations. The interwoven 2-block samplers for instance show IFs of 30 or below for all underlying true parameter values.

Finally, we investigate sampling efficiency for σ. Table 5 summarizes median IFs for draws from $p(\sigma|\mathbf{y})$. We observe a similar overall picture to the one presented in Table 1: C performs poorly when σ_{true} and ϕ_{true} are small, and NC performs poorly when σ_{true} and ϕ_{true} are large, while interweaving strategies perform well for all underlying parameter values. This result partially contrasts the conclusions of Strickland et al. (2008), who associate better mixing with larger $|\phi|$ for all parameterizations and recommend the non-centered parameterization in any setup. It should be noted, however, that these authors use a different sampling algorithm that does not rely on Gaussian mixture approximation. Moreover, the parameter range investigated in their paper does not span the range of parameters examined in our paper. For a summary of percentage gains in terms of effective sample size, see Table 6.

5. Application to exchange rate data

We apply our estimation methodology to daily Euro exchange rates. The data stems from the European Central Bank's Statistical Data Warehouse and comprises 3140 observations of 23 currencies ranging from January 3, 2000 to April 4, 2012.

Table 1
Inefficiency factors for 100 000 draws from $p(\mu|\mathbf{y})$ in various parameterizations using different blocking strategies. Time series length $T = 5000$, the values reported are medians of 500 repetitions and T_{CPU} denotes the median time to complete 1000 iterations.

Shading: 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500+

$p(\mu\|\mathbf{y})$	σ_{true} \ ϕ_{true}	0	0.5	0.8	0.9	0.95	0.96	0.97	0.98	0.99
C (1-block)	0.1	591	227	62	18	6	5	3	2	3
	0.2	245	90	23	9	3	2	2	2	3
$T_{\text{CPU}} = 2.30$	0.3	112	48	15	5	2	2	2	2	4
	0.4	70	34	11	3	2	2	2	2	5
	0.5	52	26	8	3	2	2	2	2	5
C (2-block)	0.1	641	223	64	18	6	5	3	2	3
	0.2	253	89	22	9	3	2	2	2	3
$T_{\text{CPU}} = 2.31$	0.3	113	47	15	5	2	2	2	2	4
	0.4	72	33	11	3	2	2	2	2	5
	0.5	52	26	8	2	2	2	2	2	5
C (3-block)	0.1	609	244	67	20	6	5	4	2	2
	0.2	250	89	23	8	3	2	2	1	2
$T_{\text{CPU}} = 2.31$	0.3	112	47	14	5	2	2	1	1	2
	0.4	69	33	11	4	2	1	1	2	2
	0.5	51	25	8	3	2	1	1	2	2
NC (2-block)	0.1	9	10	12	13	21	30	50	113	487
	0.2	24	21	15	22	70	108	190	419	1729
$T_{\text{CPU}} = 2.34$	0.3	23	17	17	41	148	234	407	922	3743
	0.4	18	16	23	70	265	412	726	1707	6790
	0.5	17	17	32	105	411	660	1149	2534	9421
NC (3-block)	0.1	9	10	13	13	21	31	50	116	516
	0.2	24	21	14	22	71	109	191	437	1883
$T_{\text{CPU}} = 2.35$	0.3	23	17	17	41	149	240	416	935	3843
	0.4	18	16	23	70	268	410	735	1722	7152
	0.5	17	17	32	106	409	661	1151	2589	10355
GIS-C (2-block)	0.1	9	9	11	8	4	3	3	2	3
	0.2	23	20	11	5	2	2	2	2	3
$T_{\text{CPU}} = 2.36$	0.3	22	15	9	4	2	2	2	2	3
	0.4	17	13	7	3	2	2	2	2	4
	0.5	15	12	5	2	2	2	2	2	4
GIS-C (3-block)	0.1	9	9	12	8	4	3	2	2	2
	0.2	23	20	11	5	2	2	2	1	1
$T_{\text{CPU}} = 2.37$	0.3	22	15	9	3	2	1	1	1	2
	0.4	17	13	7	3	1	1	1	1	2
	0.5	15	12	5	2	1	1	1	2	2
GIS-NC (2-block)	0.1	9	9	11	8	4	3	3	2	3
	0.2	23	20	11	5	2	2	2	2	3
$T_{\text{CPU}} = 2.40$	0.3	22	16	9	3	2	2	2	2	3
	0.4	17	13	7	3	2	2	2	2	4
	0.5	15	12	5	2	2	2	2	2	4

Table 2
Percentage gains in effective sample size for the 2-block sampler. Top table: $\text{ESS}_{\text{GIS-C}}$ vs. $\max(\text{ESS}_{\text{C}}, \text{ESS}_{\text{NC}})$. Bottom table: $\text{ESS}_{\text{GIS-C}}$ vs. $\min(\text{ESS}_{\text{C}}, \text{ESS}_{\text{NC}})$.

Shading: 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100+ (top)

Shading: 0 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000+ (bottom)

$p(\mu\|\mathbf{y})$	σ_{true} \ ϕ_{true}	0	0.5	0.8	0.9	0.95	0.96	0.97	0.98	0.99
GIS-C vs. better	0.1	2	2	9	49	53	34	21	14	11
	0.2	4	4	36	61	18	13	9	6	15
	0.3	2	13	79	39	11	9	6	6	21
	0.4	8	25	62	23	7	5	5	7	28
	0.5	13	41	47	15	5	6	5	8	32
GIS-C vs. worse	0.1	7404	2264	459	113	409	797	1825	5290	16547
	0.2	999	335	107	298	2924	5415	10731	24612	64832
	0.3	415	210	102	1070	8051	14082	25470	54664	114292
	0.4	329	154	250	2588	16212	26504	45144	88863	164695
	0.5	248	114	509	4754	26553	43176	72398	121917	240626

Table 3
Inefficiency factors for 100 000 draws from $p(\phi|\mathbf{y})$ in various parameterizations using different blocking strategies. Time series length $T = 5000$, the values reported are medians of 500 repetitions and T_{CPU} denotes the median time to complete 1000 iterations.

Shading: 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500+

$p(\phi\|\mathbf{y})$	σ_{true} \ ϕ_{true}	0	0.5	0.8	0.9	0.95	0.96	0.97	0.98	0.99
C (1-block)	0.1	138	203	238	317	367	339	262	170	80
	0.2	136	195	289	241	113	89	66	43	26
$T_{\text{CPU}} = 2.30$	0.3	129	172	209	104	49	40	30	22	15
	0.4	113	146	115	57	28	23	19	14	13
	0.5	96	112	73	36	19	16	13	11	11
C (2-block)	0.1	139	201	234	318	360	319	241	152	69
	0.2	137	194	288	232	106	82	59	38	22
$T_{\text{CPU}} = 2.31$	0.3	129	171	203	100	46	36	27	19	13
	0.4	113	148	112	54	26	22	17	12	11
	0.5	96	113	71	35	18	15	12	9	10
C (3-block)	0.1	29900	29980	35443	33223	13461	9054	4862	2347	719
	0.2	29607	30722	27821	8991	1799	1142	685	363	154
$T_{\text{CPU}} = 2.31$	0.3	26975	26012	11051	2212	564	372	241	141	69
	0.4	23672	20661	4382	882	249	181	122	76	42
	0.5	18800	13056	1990	449	146	105	74	50	30
NC (2-block)	0.1	138	201	226	169	85	74	61	53	58
	0.2	136	193	144	80	52	48	45	45	58
$T_{\text{CPU}} = 2.34$	0.3	130	159	92	56	43	42	41	44	53
	0.4	113	121	67	47	39	40	40	43	48
	0.5	96	91	55	42	38	37	38	40	42
NC (3-block)	0.1	139	202	223	168	85	73	60	51	57
	0.2	137	192	144	79	52	48	45	45	58
$T_{\text{CPU}} = 2.35$	0.3	129	161	92	56	43	42	41	43	52
	0.4	113	120	67	47	39	39	40	43	50
	0.5	96	91	56	42	38	38	38	40	45
GIS-C (2-block)	0.1	128	186	206	157	76	66	51	39	26
	0.2	125	180	134	70	39	33	28	21	14
$T_{\text{CPU}} = 2.36$	0.3	119	149	82	43	25	21	17	13	9
	0.4	105	113	55	31	18	15	12	9	7
	0.5	88	84	41	23	14	12	9	7	6
GIS-C (3-block)	0.1	139	201	223	168	79	67	52	40	28
	0.2	137	194	139	72	40	34	28	22	14
$T_{\text{CPU}} = 2.37$	0.3	130	160	84	44	26	22	18	14	9
	0.4	114	120	56	31	18	16	13	10	7
	0.5	95	87	42	24	14	12	10	8	6
GIS-NC (2-block)	0.1	127	187	208	155	76	66	51	39	27
	0.2	126	180	134	70	39	33	27	21	14
$T_{\text{CPU}} = 2.40$	0.3	119	150	82	43	25	22	17	13	9
	0.4	105	113	55	31	18	15	12	9	7
	0.5	89	84	41	23	14	11	9	7	6

Table 4
Percentage gains in effective sample size for the 2-block sampler. Top table: $\text{ESS}_{\text{GIS-C}}$ vs. $\max(\text{ESS}_{\text{C}}, \text{ESS}_{\text{NC}})$. Bottom table: $\text{ESS}_{\text{GIS-C}}$ vs. $\min(\text{ESS}_{\text{C}}, \text{ESS}_{\text{NC}})$.

Shading: 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100+ (top)

Shading: 0 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000+ (bottom)

$p(\phi\|\mathbf{y})$	σ_{true} \ ϕ_{true}	0	0.5	0.8	0.9	0.95	0.96	0.97	0.98	0.99
GIS-C vs.	0.1	8	8	10	7	12	12	20	36	119
	0.2	9	7	8	14	34	44	64	82	57
better	0.3	9	7	12	31	71	69	52	42	47
	0.4	7	7	22	52	46	41	34	32	65
	0.5	8	8	35	50	31	28	28	28	81
GIS-C vs.	0.1	9	9	14	102	371	381	376	293	162
	0.2	9	8	115	231	173	146	114	115	315
worse	0.3	9	15	147	131	81	93	136	233	480
	0.4	8	31	104	76	118	158	220	358	600
	0.5	8	34	72	79	174	225	317	441	636

Table 5
Inefficiency factors for 100 000 draws from $p(\sigma|\mathbf{y})$ in various parameterizations using different blocking strategies. Time series length $T = 5000$, the values reported are medians of 500 repetitions and T_{CPU} denotes the median time to complete 1000 iterations.

Shading: 0 | 50 | 100 | 150 | 200 | 250 | 300 | 350 | 400 | 450 | 500+

$p(\sigma\|\mathbf{y})$	σ_{true} \ ϕ_{true}	0	0.5	0.8	0.9	0.95	0.96	0.97	0.98	0.99
C (1-block) $T_{\text{CPU}} = 2.30$	0.1	5403	5125	3634	1907	829	679	497	348	247
	0.2	3067	1938	809	398	196	167	140	117	107
	0.3	687	505	328	164	98	88	78	71	73
	0.4	246	269	169	96	64	60	56	53	60
	0.5	140	169	109	66	49	46	43	43	50
C (2-block) $T_{\text{CPU}} = 2.31$	0.1	5440	4899	3608	1845	768	604	431	300	194
	0.2	3001	1813	779	374	181	151	124	100	80
	0.3	663	490	316	155	90	79	68	59	50
	0.4	238	260	163	91	58	54	48	43	38
	0.5	135	166	105	63	44	41	37	34	31
C (3-block) $T_{\text{CPU}} = 2.31$	0.1	5347	5083	3274	1265	701	617	495	388	220
	0.2	3089	1774	559	565	342	280	212	150	92
	0.3	675	398	470	357	189	147	112	78	55
	0.4	239	226	380	242	115	92	69	51	41
	0.5	133	210	309	166	77	61	47	39	32
NC (2-block) $T_{\text{CPU}} = 2.34$	0.1	57	64	91	130	90	83	73	71	87
	0.2	99	95	121	87	70	70	75	89	137
	0.3	61	77	94	70	71	77	89	114	189
	0.4	38	75	75	66	79	90	106	144	257
	0.5	31	68	67	67	90	103	126	173	316
NC (3-block) $T_{\text{CPU}} = 2.35$	0.1	57	64	90	130	90	83	73	72	92
	0.2	99	96	123	87	70	71	76	91	145
	0.3	60	77	95	70	72	78	90	115	199
	0.4	38	75	75	66	81	91	108	146	273
	0.5	31	68	67	67	90	105	129	176	335
GIS-C (2-block) $T_{\text{CPU}} = 2.36$	0.1	56	64	89	122	82	75	64	58	61
	0.2	97	93	114	76	53	50	48	48	51
	0.3	58	72	82	53	42	41	40	39	40
	0.4	35	69	58	42	35	35	34	33	33
	0.5	28	59	47	35	31	30	29	29	28
GIS-C (3-block) $T_{\text{CPU}} = 2.37$	0.1	56	64	89	125	85	76	65	60	63
	0.2	97	93	117	76	54	51	49	49	52
	0.3	57	73	83	53	43	41	40	40	41
	0.4	34	70	59	43	35	35	34	33	33
	0.5	28	60	48	36	31	30	29	29	28
GIS-NC (2-block) $T_{\text{CPU}} = 2.40$	0.1	56	64	88	123	83	74	64	58	61
	0.2	96	93	113	75	53	50	48	48	50
	0.3	58	72	81	52	42	41	40	39	40
	0.4	35	69	59	42	35	35	34	33	33
	0.5	28	58	47	35	31	30	29	29	28

Table 6
Percentage gains in effective sample size for the 2-block sampler. Top table: $\text{ESS}_{\text{GIS-C}}$ vs. $\max(\text{ESS}_{\text{C}}, \text{ESS}_{\text{NC}})$. Bottom table: $\text{ESS}_{\text{GIS-C}}$ vs. $\min(\text{ESS}_{\text{C}}, \text{ESS}_{\text{NC}})$.

Shading: 0 | 10 | 20 | 30 | 40 | 50 | 60 | 70 | 80 | 90 | 100+ (top)

Shading: 0 | 100 | 200 | 300 | 400 | 500 | 600 | 700 | 800 | 900 | 1000+ (bottom)

$p(\sigma\|\mathbf{y})$	σ_{true} \ ϕ_{true}	0	0.5	0.8	0.9	0.95	0.96	0.97	0.98	0.99
GIS-C vs. better	0.1	1	0	2	6	10	11	15	21	43
	0.2	2	2	6	15	33	39	55	85	58
	0.3	5	7	15	33	69	89	73	49	26
	0.4	8	9	28	57	66	55	42	29	15
	0.5	13	16	41	76	43	36	27	19	10
GIS-C vs. worse	0.1	9550	7593	3947	1409	834	708	574	414	218
	0.2	2999	1842	584	393	242	200	157	108	170
	0.3	1041	583	286	195	114	93	124	190	374
	0.4	588	277	180	115	126	160	215	332	677
	0.5	382	183	123	87	191	246	332	506	1031

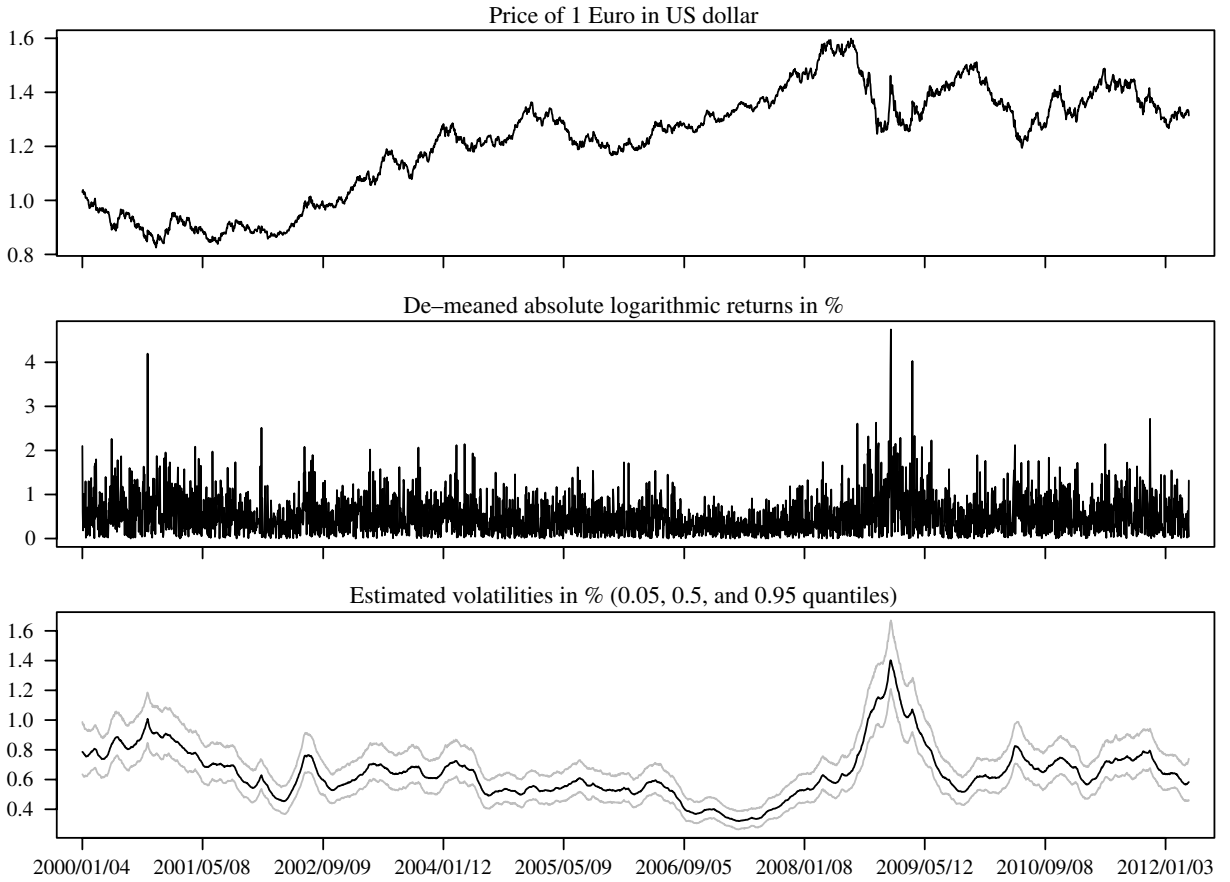

Fig. 5. Indirectly quoted EUR/US$ exchange rates (top) with de-meaned absolute log-returns (middle) and estimated instantaneous volatilities (bottom) based on 1000 000 draws via GIS-C.

In choosing the prior for ϕ we follow Kim et al. (1998), i.e. $(\phi + 1)/2 \sim \mathcal{B}(20, 1.5)$, and for the other parameters we pick rather vague priors: $\mu \sim N(-10, 100)$ and $\sigma^2 \sim \mathcal{G}\left(\frac{1}{2}, \frac{1}{2}\right)$. After a burn-in of 10 000, we use 1000 000 draws from the respective distributions in each parameterization for posterior inference.

To exemplify, Fig. 5 shows exchange rates of EUR/US$ along with absolute de-meaned log-returns, which are then used to estimate the time-varying volatilities displayed below. The transformed latent process and the absolute log-returns exhibit a similar overall pattern. Nevertheless, the volatility path is much smoother, which is due to the highly persistent autoregressive process (the posterior mean of ϕ is 0.993, the posterior mean of σ is 0.07). Marginal posterior density estimates and two-way scatterplots can be found in Fig. 6. Note that only $p(\mu|\mathbf{y})$ is symmetric, while both $p(\phi|\mathbf{y})$ and $p(\sigma|\mathbf{y})$ are skewed. Moreover, the parameter draws are (sometimes nonlinearly) correlated.

Results for all 23 examined exchange rates are displayed in Table 7. It stands out that for currencies which are closely tied to the Euro, posterior parameter means differ substantially to those found above. Most notably, the Danish krone exhibits very low overall level of volatility ($\mu_{\text{mean}} = -18$), paired with moderate persistence ($\phi_{\text{mean}} = 0.916$) and moderately high volatility of volatility ($\sigma_{\text{mean}} = 0.38$). Looking at the inefficiency factors for the raw parameterizations, one observes striking superiority of C in terms of sampling efficiency of μ, while NC usually performs better in terms of sampling efficiency of σ. Again, interweaving overcomes these problems by showing lowest IFs uniformly for all parameters and all time series. Even though not reported here in detail due to space constraints, the choice of the baseline (GIS-C vs. GIS-NC) is negligible.

6. Concluding remarks

Previous studies have shown that simple reparameterizations often turn out to have substantial impact on MCMC simulation efficiency in state-space specifications. This paper contributes to the literature by exploring the influence of choosing between two selected parameterizations for Bayesian estimation of SV models. Moreover, it provides evidence that inefficiency factors obtained from simulation experiments can heavily depend on the realization of the data generating process. Through the findings of this paper it becomes clear that employing an ancillarity-sufficiency interweaving strategy (ASIS) introduced by Yu and Meng (2011) helps to overcome shortcomings of either the centered or the non-centered

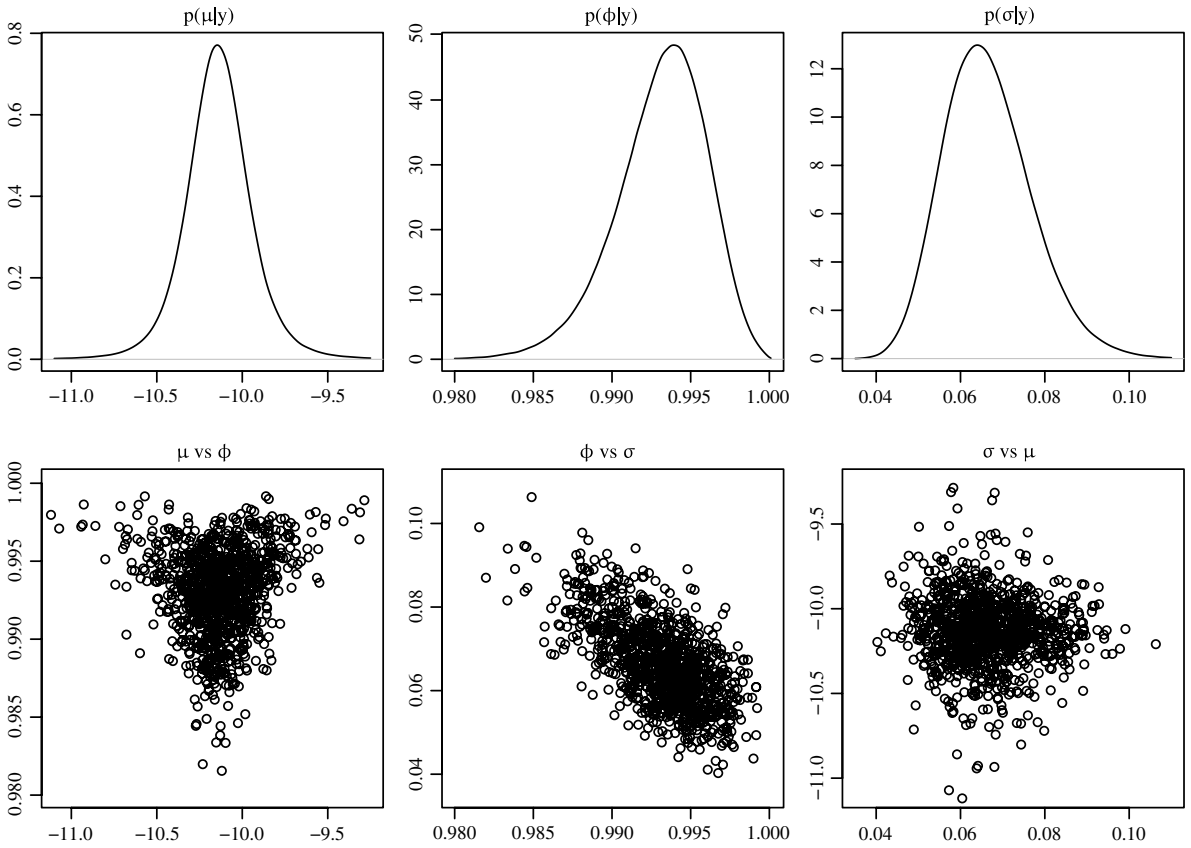

Fig. 6. Marginal posterior density estimates and bivariate scatterplots of 1000 thinned draws for the EUR/US$ exchange rate data, based on 1000 000 samples obtained via GIS-C.

Table 7
Posterior means and inefficiency factors for various estimation methods of the SV model, applied to EUR exchange rate data.

	Posterior means			IF_C			IF_{NC}			$IF_{GIS\text{-}C}$		
	μ	ϕ	σ	μ	ϕ	σ	μ	ϕ	σ	μ	ϕ	σ
Australian dollar	−10.3	0.976	0.17	3	167	256	216	120	149	2	68	97
Canadian dollar	−10.1	0.987	0.09	3	350	539	211	128	159	3	89	120
Swiss franc	−12.0	0.985	0.21	3	55	129	807	99	162	3	33	73
Czech koruna	−11.5	0.953	0.28	3	140	200	151	121	154	3	72	96
Danish krone	−18.0	0.916	0.38	5	102	145	85	90	115	4	57	72
UK pound sterling	−10.8	0.992	0.10	2	95	233	802	120	150	2	39	87
Hong Kong dollar	−10.2	0.993	0.07	2	128	309	526	89	96	2	36	75
Indonesian rupiah	−9.9	0.966	0.23	4	234	313	216	201	242	3	114	142
Japanese yen	−10.0	0.989	0.12	3	110	227	513	98	140	3	47	91
Korean won	−10.0	0.987	0.14	2	92	192	501	96	134	2	40	79
Mexican peso	−9.8	0.977	0.16	2	137	221	220	101	128	2	60	86
Malaysian ringgit	−10.3	0.990	0.08	2	220	390	404	104	124	2	59	90
Norwegian krone	−11.1	0.970	0.18	2	139	217	150	83	107	2	53	76
New Zealand dollar	−10.0	0.963	0.17	4	344	432	105	153	177	3	114	135
Philippine peso	−10.1	0.981	0.12	3	376	529	209	189	220	2	123	160
Polish zloty	−10.4	0.975	0.19	2	96	171	261	85	117	2	43	69
Romanian leu	−11.1	0.970	0.31	2	51	100	446	87	137	2	32	60
Russian ruble	−10.6	0.988	0.15	4	75	172	891	121	151	3	38	82
Swedish krona	−11.3	0.991	0.11	1	52	156	752	75	99	1	23	60
Singapore dollar	−10.6	0.995	0.07	4	138	348	998	126	132	4	47	100
Thai bhat	−10.2	0.980	0.13	3	202	314	207	102	125	3	64	90
Turkish lira	−9.8	0.966	0.27	2	72	127	259	84	127	2	42	69
US dollar	−10.1	0.993	0.07	2	126	308	504	87	99	2	37	74

parameterization by outperforming those in terms of sampling efficiency with respect to all parameters at very little extra computational cost, whereas the baseline of the interweaving strategy is of minor influence.

The concept of interweaving different parameterizations of state-space models is clearly very general, and there is good reason to hope for similar magic when applying ASIS to extension of the basic SV model such as more general innovation distributions (e.g. Liesenfeld and Jung, 2000; Delatola and Griffin, 2011), asymmetry (e.g. Yu, 2005; Omori et al., 2007) or both (e.g. Chib et al., 2002; Wang et al., 2011; Tsiotas, 2012; Ishihara and Omori, 2012; Nakajima and Omori, 2012). Preliminary results for an SV model with leverage, where a centered parameterization from Yu (2005) is compared with a non-centered version based on transforming h_t into $\tilde{h}_t = (h_t - \mu)/\sigma$ as in the present paper, show that this hope is in fact an actual possibility. A thorough investigation of this issue is however beyond the scope of this article.

Acknowledgments

The authors would like to thank the editor and two referees for their perspicacious comments on an earlier draft of this paper, and Stefan Theußl for helpful advice concerning coding and implementation.

References

Anderson, E., Bai, Z., Bischof, C., Blackford, S., Demmel, J., Dongarra, J., Croz, J.D., Greenbaum, A., Hammarling, S., McKenney, A., Sorensen, D., 1999. LAPACK Users' Guide, third ed. Society for Industrial and Applied Mathematics, Philadelphia, PA.
Basu, D., 1955. On statistics independent of a complete sufficient statistic. Sankhyā: The Indian Journal of Statistics (1933–1960) 15, 377–380.
Bos, C.S., 2012. Relating stochastic volatility estimation methods. In: Handbook of Volatility Models and their Applications. John Wiley & Sons, Inc., pp. 147–174.
Carter, C.K., Kohn, R., 1994. On Gibbs sampling for state space models. Biometrika 81, 541–553.
Chib, S., Nardari, F., Shephard, N., 2002. Markov chain Monte Carlo methods for stochastic volatility models. Journal of Econometrics 108, 281–316.
Delatola, E.-I., Griffin, J.E., 2011. Bayesian nonparametric modelling of the return distribution with stochastic volatility. Bayesian Analysis 6, 901–926.
Durbin, J., Koopman, S.J., 2002. A simple and efficient simulation smoother for state space time series analysis. Biometrika 89, 603–615.
Eddelbuettel, D., François, R., 2011. Rcpp: seamless R and C++ integration. Journal of Statistical Software 40, 1–18.
Frühwirth-Schnatter, S., 1994. Data augmentation and dynamic linear models. Journal of Time Series Analysis 15, 183–202.
Frühwirth-Schnatter, S., 2004. Efficient Bayesian parameter estimation. In: Harvey, A., Koopman, S.J., Shephard, N. (Eds.), State Space and Unobserved Component Models: Theory and Applications. Cambridge University Press, Cambridge, pp. 123–151.
Frühwirth-Schnatter, S., Sögner, L., 2008. Bayesian estimation of the Heston stochastic volatility model. Communications in Dependability and Quality Management 11, 5–25.
Frühwirth-Schnatter, S., Wagner, H., 2010. Stochastic model specification search for Gaussian and partially non-Gaussian state space models. Journal of Econometrics 154, 85–100.
Gabriel, E., Fagg, G.E., Bosilca, G., Angskun, T., Dongarra, J.J., Squyres, J.M., Sahay, V., Kambadur, P., Barrett, B., Lumsdaine, A., Castain, R.H., Daniel, D.J., Graham, R.L., Woodall, T.S., 2004. Open MPI: goals, concept, and design of a next generation MPI implementation. In: Proceedings, 11th European PVM/MPI Users' Group Meeting. Budapest, Hungary, pp. 97–104.
Gelfand, A., Sahu, S., Carlin, B., 1995. Efficient parametrisations for normal linear mixed models. Biometrika 82, 479–488.
Hull, J., White, A., 1987. The pricing of options on assets with stochastic volatilities. The Journal of Finance 42, 281–300.
Ishihara, T., 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.
Jacquier, E., Polson, N.G., Rossi, P.E., 1994. Bayesian analysis of stochastic volatility models. Journal of Business & Economic Statistics 12, 371–417.
Kastner, G., 2013. stochvol: Efficient Bayesian inference for stochastic volatility (SV) models. R Package Version 0.5-0.
Kim, S., Shephard, N., Chib, S., 1998. Stochastic volatility: likelihood inference and comparison with ARCH models. Review of Economic Studies 65, 361–393.
L'Ecuyer, P., Simard, R., Chen, E.J., Kelton, W.D., 2002. An object-oriented random-number package with many long streams and substreams. Operations Research 50, 1073–1075.
Liesenfeld, R., Jung, R.C., 2000. Stochastic volatility models: conditional normality versus heavy-tailed distributions. Journal of Applied Econometrics 15, 137–160.
Liesenfeld, R., Richard, J.-F., 2006. Classical and Bayesian analysis of univariate and multivariate stochastic volatility models. Econometric Reviews 25, 335–360.
McCausland, W.J., Miller, S., Pelletier, D., 2011. Simulation smoothing for state space models: a computational efficiency analysis. Computational Statistics and Data Analysis 55, 199–212.
Nakajima, J., Omori, Y., 2012. Stochastic volatility model with leverage and asymmetrically heavy-tailed error using GH skew Student's-distribution. Computational Statistics and Data Analysis 56, 3690–3704.
Omori, Y., Chib, S., Shephard, N., Nakajima, J., 2007. Stochastic volatility with leverage: fast and efficient likelihood inference. Journal of Econometrics 140, 425–449.
Pitt, M.K., 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.
Plummer, M., Best, N., Cowles, K., Vines, K., 2006. CODA: convergence diagnosis and output analysis for MCMC. R News 6, 7–11.
R Development Core Team, 2012. R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria, ISBN: 3-900051-07-0.
Roberts, G.O., Papaspiliopoulos, O., Dellaportas, P., 2004. Bayesian inference for non-Gaussian Ornstein–Uhlenbeck stochastic volatility processes. Journal of the Royal Statistical Society: Series B 66, 369–393.
Rue, H., 2001. Fast sampling of Gaussian Markov random fields. Journal of the Royal Statistical Society: Series B 63, 325–338.
Shephard, N., 1994. Partial non-Gaussian state space. Biometrika 81, 115–131.
Shephard, N., Kim, S., 1994. [Bayesian analysis of stochastic volatility models]: Comment. Journal of Business & Economic Statistics 12, 406–410.
Shephard, N., Pitt, M.K., 1997. Likelihood analysis of non-Gaussian measurement time series. Biometrika 84, 653–667.
Strickland, C.M., Martin, G.M., Forbes, C.S., 2008. Parameterisation and efficient MCMC estimation of non-Gaussian state space models. Computational Statistics and Data Analysis 52, 2911–2930.
Taylor, S.J., 1982. Financial returns modelled by the product of two stochastic processes—a study of daily sugar prices 1691–1679. In: Anderson, O.D. (Ed.), Time Series Analysis: Theory and Practice 1. North-Holland, pp. 203–226.
Tierney, L., Rossini, A.J., Li, N., Sevcikova, H., 2011. snow: Simple network of workstations. R Package Version 0.3-8.
Tsiotas, G., 2012. On generalised asymmetric stochastic volatility models. Computational Statistics and Data Analysis 56, 151–172.
Wang, J.J., Chan, J.S., Choy, S.B., 2011. Stochastic volatility models with leverage and heavy-tailed distributions: a Bayesian approach using scale mixtures. Computational Statistics and Data Analysis 55, 852–862.
Yu, J., 2005. On leverage in a stochastic volatility model. Journal of Econometrics 127, 165–178.
Yu, Y., Meng, X.-L., 2011. To center or not to center: that is not the question—an ancillarity-suffiency interweaving strategy (ASIS) for boosting MCMC efficiency. Journal of Computational and Graphical Statistics 20, 531–570.