Barra, István; Hoogerheide, Lennart; Koopman, Siem Jan; Lucas, André

Working Paper

Joint Bayesian Analysis of Parameters and States in Nonlinear, Non-Gaussian State Space Models

Tinbergen Institute Discussion Paper, No. 14-118/III

Provided in Cooperation with:
Tinbergen Institute, Amsterdam and Rotterdam

Suggested Citation: Barra, István; Hoogerheide, Lennart; Koopman, Siem Jan; Lucas, André (2014) : Joint Bayesian Analysis of Parameters and States in Nonlinear, Non-Gaussian State Space Models, Tinbergen Institute Discussion Paper, No. 14-118/III, Tinbergen Institute, Amsterdam and Rotterdam

This Version is available at:
https://hdl.handle.net/10419/107830

TI 2014-118/III
Tinbergen Institute Discussion Paper

Joint Bayesian Analysis of Parameters and States in Nonlinear, Non-Gaussian State Space Models

*István Barra**
Lennart Hoogerheide
Siem Jan Koopman
André Lucas

** Duisenberg School of Finance, the Netherlands.*

Faculty of Economics and Business Administration, VU University Amsterdam, and Tinbergen Institute, the Netherlands.

Tinbergen Institute is the graduate school and research institute in economics of Erasmus University Rotterdam, the University of Amsterdam and VU University Amsterdam.

More TI discussion papers can be downloaded at http://www.tinbergen.nl

Tinbergen Institute has two locations:

Tinbergen Institute Amsterdam
Gustav Mahlerplein 117
1082 MS Amsterdam
The Netherlands
Tel.: +31(0)20 525 1600

Tinbergen Institute Rotterdam
Burg. Oudlaan 50
3062 PA Rotterdam
The Netherlands
Tel.: +31(0)10 408 8900
Fax: +31(0)10 408 9031

Duisenberg school of finance is a collaboration of the Dutch financial sector and universities, with the ambition to support innovative research and offer top quality academic education in core areas of finance.

DSF research papers can be downloaded at: http://www.dsf.nl/

Duisenberg school of finance
Gustav Mahlerplein 117
1082 MS Amsterdam
The Netherlands
Tel.: +31(0)20 525 8579

Joint Bayesian analysis of parameters and states in nonlinear, non-Gaussian state space models *

István Barra,[(a,b)] *Lennart Hoogerheide,*[(a)]
Siem Jan Koopman,[(a,c),‡] *André Lucas*[(a)]

[(a)] VU University Amsterdam and Tinbergen Institute, The Netherlands

[(b)] Duisenberg School of Finance, The Netherlands

[(c)] CREATES, Aarhus University, Denmark

August 2014

Abstract

We propose a new methodology for designing flexible proposal densities for the joint posterior density of parameters and states in a nonlinear non-Gaussian state space model. We show that a highly efficient Bayesian procedure emerges when these proposal densities are used in an independent Metropolis-Hastings algorithm. A particular feature of our approach is that smoothed estimates of the states and the marginal likelihood are obtained directly as an output of the algorithm. Our method provides a computationally efficient alternative to several recently proposed algorithms. We present extensive simulation evidence for stochastic volatility and stochastic intensity models. For our empirical study, we analyse the performance of our method for stock returns and corporate default panel data.

JEL classification: C11, C15, C22, C32, C58.

Keywords: Bayesian inference, importance sampling, Monte Carlo estimation, Metropolis-Hastings algorithm, mixture of Student's t-distributions.

*E-mail addresses of authors: István Barra, E-mail: `i.barra@vu.nl`; Lennart Hoogerheide, E-mail: `l.f.hoogerheide@vu.nl`; ‡ Corresponding author, Siem Jan Koopman, E-mail: `s.j.koopman@vu.nl`; André Lucas, E-mail: `a.lucas@vu.nl`. Address: VU University Amsterdam, FEWEB, De Boelelaan 1105, 1081 HV Amsterdam. This paper is an updated version of the paper that appeared earlier as Barra, I., Hoogerheide, L.F., Koopman, S.J., and Lucas, A. (2013) "Joint Independent Metropolis-Hastings Methods for Nonlinear Non-Gaussian State Space Models". TI Discussion Paper 13-050/III. Amsterdam: Tinbergen Institute.

1 Introduction

Empirical studies based on nonlinear non-Gaussian state space models are widespread in economics and finance. It is well known that Bayesian parameter estimation in this type of models can be difficult, and that the evaluation of the marginal likelihood function is a challenging task. One of the complications is that the joint posterior density of the parameter and state vectors is typically high-dimensional which makes it cumbersome to develop successful proposal distributions and Monte Carlo algorithms. It is standard practice to overcome this difficulty by disentangling the target density into lower dimensional densities and develop proposal densities for each of them. However, this approach leads to other problems. Although the curse of dimensionality may be resolved to some extent, it is rather demanding to design a proposal density on a case by case basis for each lower dimensional target density. Furthermore, these separately defined proposal densities may not adequately characterize the properties of the joint posterior density, possibly resulting in unsatisfactory computational performance of the method and possibly leading to biased estimates of posterior moments and marginal likelihoods.

The aim of our paper is to develop flexible proposal distributions for the joint posterior density of the parameters and states in nonlinear, non-Gaussian state space models. Our proposed Extended Mixture of t by Importance Sampling Weighted Expectation Maximization (EMitISEM) method extends the *Mixture of t by Importance Sampling weighted Expectation Maximization* MitISEM method of Hoogerheide et al. (2012) for the case when the likelihood is not available in closed form. We differentiate two categories of nonlinear, non-Gaussian state space models based on the transition density of the states. In case of a Gaussian transition density we propose a joint candidate for the posterior of the parameters and states. The proposal density in our *EMitISEM* method in this case consists of two components: (i) a mixture of Student's t-densities that targets the marginal posterior density of the parameters, and (ii) an approximating density that targets the density of the states given the observations and the parameters. The mixture of Student's t-densities is constructed by means of an extension of the MitISEM method; see Hoogerheide et al. (2012). The proposal density for the states is then based on a given set of parameters. We can take any reasonable approximating density for the states including those developed by Shephard and Pitt (1997), Durbin and Koopman (1997), Richard and Zhang (2007), Koopman et al. (2014) and McCausland (2012). We can use these proposal densities in an independent MH algorithm or in an importance sampling procedure to estimate the marginal likelihood and parameters. The resulting procedure can be almost fully automated without requiring user intervention. If the transition den-

sity is not Gaussian, we propose to replace the likelihood used in the MitISEM method by an unbiased estimator of the likelihood in which the states are integrated out using a particle filter.

We argue and show that our approach is computationally efficient and robust and can be regarded as an effective alternative to existing Markov chain Monte Carlo (MCMC) methods. Our method provides at least two advantages. First, the methodology can be fully automated. There is no need for case by case fine tuning of the algorithm whenever a different model specification with a possibly different observation density is considered. Second, the necessary computations can be implemented in a parallel manner. This implies that we can use state-of-the-art computer technology based on graphics cards to further reduce the computing time of our method.

Our work relates to two strands in the literature. First, we contribute to the recent literature on Bayesian estimation of nonlinear non-Gaussian state space models by jointly sampling parameters and state paths. McCausland (2012) suggests a proposal density based on a higher order approximation of the states given the parameter vector. Although this sampler appears to be efficient, it relies on the assumption that the state vector is univariate. Chan and Strachan (2012) propose a method that overcomes this restriction. Their proposal density for the state vectors, however, is derived from a local approximation of the smoothed density, which can lead to reduced performance in higher dimensional problems.

Second, our paper relates to the literature on the Bayesian estimation of nonlinear non-Gaussian state space models using particle filters. Andrieu et al. (2010) develop a collection of Particle Markov Chain Monte Carlo (PMCMC) methods for parameter estimation. As argued by Flury and Shephard (2011), the key idea of several PMCMC methods is that the unknown true likelihood can be replaced by an unbiased estimator of the likelihood within a Metropolis-Hastings iteration. Although PMCMC methods provide a general solution to parameter and state estimation in nonlinear non-Gaussian state space models, they require the application of a particle filter for each iteration, see for example Doucet et al. (2012). To overcome this computational burden, Lindsten and Schön (2012) propose a modified version of the particle Gibbs sampler. For the same motivation, Pitt et al. (2012) develop an adaptive version of the particle independent Metropolis-Hastings algorithm with partially adapted auxiliary particle filters. In an extensive simulation study we show that our EMitISEM method is a viable alternative to other recently developed methods including the Adaptive Independent Metropolis-Hastings method of Pitt et al. (2012) and the particle filter MCMC methods of Andrieu et al. (2010). We compare the methods in detail for two cases: the stochastic volatility

model and the stochastic intensity model. For these cases, we provide evidence that our method provides posterior draws and estimates of posterior moments in a computationally more efficient manner than the state-of-the-art alternatives that we consider. We conclude that particle filters may not necessarily be the most efficient or robust approach from a numerical perspective in all empirically relevant cases.

The remainder of this paper is organized as follows. In Section 2 we introduce the new methodology. In Section 3 we demonstrate the performance of the methodology against state-of-the-art alternatives in a Monte Carlo study designed for parameter and state estimation in stochastic volatility and stochastic intensity models. In Section 4, we empirically illustrate the methods by considering a long time series of IBM stock returns and a large panel data set of U.S. corporate defaults. Section 5 concludes.

2 The EMitISEM method for state space models

2.1 EMitISEM in case of Gaussian state equations

For a time series of observations $y_1, \ldots, y_T$, we define the nonlinear non-Gaussian state space model by the observation density and the state equation

$$y_t \sim p_y(y_t|x_t;\theta), \qquad x_t = c_t + Z_t\alpha_t, \tag{1}$$

$$\alpha_{t+1} = d_t + T_t\alpha_t + \eta_t, \quad \eta_t \sim \text{NID}\left(0, Q_t\right), \tag{2}$$

where p_y is the observation density, x_t is the latent dynamic signal, θ is the parameter vector, α_t is the state vector, for $t = 1, \ldots, T$. Moreover the η_t disturbance vectors are normally distributed and independent for $t = 1, \ldots, T$. We discuss generalizations in Section 2.3. The signal x_t is a linear function of the state vector α_t, with scalar intercept c_t and loading vector Z_t both possibly depending in a deterministic way on time and on the parameter vector θ, i.e. $c_t = c(t;\theta)$ and $Z_t = Z(t;\theta)$. The state vector α_t evolves as a linear Gaussian dynamic process given by (2) where the intercept vector $d_t = d(t;\theta)$, transition matrix $T_t = T(t;\theta)$ and variance matrix $Q_t = Q(t;\theta)$ are deterministic functions of t and θ. We assume that all vectors and matrices have appropriate dimensions. Bayesian inference for the model given by equations (1) and (2) involves the estimation of the properties of interest of the posterior density $p(\theta|y)$ of the parameter vector θ and the smoothed density of the signal $p(x|y)$, where $y = (y_1, \ldots, y_T)'$ and $x = (x_1, \ldots, x_T)'$.

In the original MitISEM procedure, Hoogerheide et al. (2012) propose to approximate the posterior density $p(\theta|y)$ (of which only a kernel is required) by considering the Student's t mixture $q_{\zeta'}(\theta|y)$ where ζ' includes mode vectors, scale matrices, degrees of

freedom and mixing weights for the Student's t-distributions in the candidate mixture $q_{\zeta'}(\theta|y)$, and minimizing the Kullback and Leibler (1951) divergence

$$\int p(\theta|y) \log p(\theta|y) \mathrm{d}\theta - \int p(\theta|y) \log q_{\zeta'}(\theta|y) \mathrm{d}\theta. \tag{3}$$

Since the first term does not depend on the proposal, an approximation of the Kullback-Leibler divergence can be minimized by maximizing

$$\frac{1}{N} \sum_{j=1}^{N} \frac{p(\theta^{(j)}|y)}{q_{\zeta}(\theta^{(j)}|y)} \log q_{\zeta'}(\theta^{(j)}|y), \tag{4}$$

where $\theta^{(j)} \sim q_{\zeta}(\theta|y)$ is a sequence of independent and identically distributed (i.i.d) draws from a previous proposal density, for $j = 1, \ldots, N$. Unfortunately, in nonlinear non-Gaussian state space models we do not know (a kernel of) the posterior density $p(\theta^{(j)}|y)$ in closed form, so that we have to modify the original MitISEM method.

The Kullback-Leibler divergence between our target density $p(x, \theta|y)$ and the joint proposal density $q_{\zeta'}(x, \theta|y)$ is given by

$$\int p(x, \theta|y) \log p(x, \theta|y) \mathrm{d}x \mathrm{d}\theta - \int p(x, \theta|y) \log q_{\zeta'}(x, \theta|y) \mathrm{d}x \mathrm{d}\theta. \tag{5}$$

Minimizing the KL divergence is therefore equivalent to maximizing

$$\int p(x, \theta|y) \log q_{\zeta'}(x, \theta|y) \mathrm{d}x \mathrm{d}\theta = \int \frac{p(x, \theta|y)}{q_{\zeta}(x, \theta|y)} q_{\zeta}(x, \theta|y) \log q_{\zeta'}(x, \theta|y) \mathrm{d}x \mathrm{d}\theta, \tag{6}$$

where $q_{\zeta}(x, \theta|y)$ is a previous proposal density. As our proposal density, we use $q_{\zeta'}(x, \theta|y)$ $= q(x|\theta, y) q_{\zeta'}(\theta|y)$, where we obtain $q(x|\theta, y)$ from an approximation to the smoothed state density. For example, the approximation from the NAIS method proposed by Koopman et al. (2014) appears to be sufficiently accurate. Substituting $q_{\zeta'}(x, \theta|y) = q(x|\theta, y) q_{\zeta'}(\theta|y)$ into the right-hand side of (6) we get

$$\int \frac{p(x, \theta|y)}{q_{\zeta}(x, \theta|y)} q_{\zeta}(x, \theta|y) \log q(x|\theta, y) \mathrm{d}x \mathrm{d}\theta + \int \frac{p(x, \theta|y)}{q_{\zeta}(x, \theta|y)} q_{\zeta}(x, \theta|y) \log q_{\zeta'}(\theta|y) \mathrm{d}x \mathrm{d}\theta. \tag{7}$$

The first term in (7) does not depend on ζ'. Hence we maximize the second term in (7), with respect to ζ' using the approximation

$$\int \frac{p(x, \theta|y)}{q_{\zeta}(x, \theta|y)} q_{\zeta}(x, \theta|y) \log q_{\zeta'}(\theta|y) \mathrm{d}x \mathrm{d}\theta \approx \frac{1}{N} \sum_{j=1}^{N} \frac{p(x^{(j)}, \theta^{(j)}|y)}{q_{\zeta}(x^{(j)}, \theta^{(j)}|y)} \log q_{\zeta'}(\theta^{(j)}|y), \tag{8}$$

where $(x^{(j)}, \theta^{(j)}) \sim q_{\zeta}(x, \theta|y)$ is an i.i.d. sequence for $j = 1, \ldots, N$.

If we compare (4) with (8), we see that the difference with the original MitISEM approach is that we replace the weight $p(\theta^{(j)}|y)/q_{\zeta}(\theta^{(j)}|y)$ by $p(x^{(j)}, \theta^{(j)}|y)$ $/q_{\zeta}(x^{(j)}, \theta^{(j)}|y)$.

This novel result implies that we can use the MitISEM algorithm as described in Appendix A, with only a slight modification. We note that the new weights

$$w^{(j)} = \frac{p(x^{(j)}, \theta^{(j)}|y)}{q_\zeta(x^{(j)}, \theta^{(j)}|y)} \propto \frac{p(y|x^{(j)}, \theta^{(j)})p(x^{(j)}|\theta^{(j)})p(\theta^{(j)})}{q(x^{(j)}|\theta^{(j)}, y)q_\zeta(\theta^{(j)}|y)} \tag{9}$$

can be replaced by

$$w^{(j)} \propto q(y|\theta^{(j)})\frac{p(y|x^{(j)}, \theta^{(j)})p(\theta^{(j)})}{q(y|x^{(j)}, \theta^{(j)})q_\zeta(\theta^{(j)}|y)}, \tag{10}$$

where we used the relations

$$q(x^{(j)}|\theta^{(j)}, y) = \frac{q(y|x^{(j)}, \theta^{(j)})q(x^{(j)}|\theta^{(j)})}{q(y|\theta^{(j)})}, \tag{11}$$

and $p(x^{(j)}|\theta^{(j)}) = q(x^{(j)}|\theta^{(j)})$, which follow from the fact that we use the same Gaussian linear state equation for the true model and the approximating Gaussian linear state space model upon which our proposal density for the signal is based; see Appendix B for the details on this approximating model and the proposal density for the signal. The formulation in (10) is more convenient than (9), as we do not have to evaluate the density $q(x^{(j)}|\theta^{(j)}, y)$.

EMitISEM algorithm

1. **Initialization**: Simulate a series of N parameter vector draws $\theta^{(0,1)}, \ldots, \theta^{(0,N)}$ from the initial Student's t distribution $q_{\zeta(0)}(\theta|y)$ with parameters denoted by $\zeta(0)$. The initial parameters consist of the mode equal to the simulated maximum likelihood estimate of θ, the scale parameter equal to minus the inverse Hessian of the log likelihood evaluated at the current parameter estimates and the degree of freedom parameter equal to 5. The evaluation of the likelihood is given in (A11). Conditionally on the draws $\theta^{(0,1)}, \ldots, \theta^{(0,N)}$ we simulate a corresponding series of N signal paths for x from $q(x|\theta^{(j)}, y)$ and denote these by $x^{(0,1)}, \ldots, x^{(0,N)}$. Finally, we evaluate the joint importance sampling (IS) weights $w^{(0,1)}, \ldots, w^{(0,N)}$ given by (10).

2. **Adaptation**: Estimate the mean and variance of the target distribution via IS using the draws $\theta^{(0,1)}, \ldots, \theta^{(0,N)}$ from $q_{\zeta(0)}(\theta|y)$. We adopt the estimated mean and variance as the mode and scale of the new proposal distribution, which is denoted by $q_{\zeta(1)}$. We simulate draws $\theta^{(1,1)}, \ldots, \theta^{(1,N)}$ from $q_{\zeta(1)}$. Finally, we simulate signal paths $x^{(1,1)}, \ldots, x^{(1,N)}$ conditionally on the parameter draws, and evaluate the joint IS weights $w^{(1,1)}, \ldots, w^{(1,N)}$ given by (10) and using $q_\zeta = q_{\zeta(1)}$.

3. **IS weighted EM algorithm**: We obtain the updated proposal $q_{\zeta(2)}$ from the IS weighted EM algorithm of MitISEM, using the latest draws $\theta^{(1,1)}, \ldots, \theta^{(1,N)}$ and

corresponding IS weights $w^{(1,1)}, \ldots, w^{(1,N)}$. Appendix A provides further details about the MitISEM algorithm. We simulate draws $\theta^{(2,1)}, \ldots, \theta^{(2,N)}$ from the updated proposal $q_{\zeta(2)}$, and signal paths $x^{(2,1)}, \ldots, x^{(2,N)}$ conditionally on these parameter draws. We compute the corresponding IS weights $w^{(2,1)}, \ldots, w^{(2,N)}$ in (10) using $q_\zeta = q_{\zeta(2)}$. Calculate the coefficient of variation of the weights $w^{(2,1)}, \ldots, w^{(2,N)}$ (i.e., the standard deviation divided by the mean) $CoV^{(2)}$, where we use the notation

$$CoV^{(k)} = \frac{\sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(w^{(k,j)}\right)^2 - \left(\frac{1}{N}\sum_{j=1}^{N} w^{(k,j)}\right)^2}}{\frac{1}{N}\sum_{j=1}^{N} w^{(k,j)}} \tag{12}$$

for the coefficient of variation of the weights at iteration k. Set $i = 3$.

4. **Iterate on the number of mixture components**:

   We consider 10% of the simulated draws from the last iteration $\theta^{(i-1,1)}, \ldots, \theta^{(i-1,N)}$ that correspond to the highest IS weights, based on the current candidate mixture density $q_{\zeta(i-1)}$. We use this smaller set of draws and weights to compute a new mode and scale matrix. The new mode and scale are used as starting parameters for the additional Student's t component in the mixture. We adopt this choice because the new Student's t component should cover a part of the parameter space that is insufficiently covered by the previous candidate, when compared to the target density as discussed in Hoogerheide et al. (2012). The starting values for the mixture probability and degrees of freedom parameter for the new Student's t component are set to 0.1 and 5, respectively. The starting values of the mixture probabilities for the older Student's t components are obtained by multiplying the latest values by 0.9. Given the last set of N simulated draws $\theta^{(i-1,1)}, \ldots, \theta^{(i-1,N)}$ and the corresponding importance weights $w^{(i-1,1)}, \ldots, w^{(i-1,N)}$, we apply the IS weighted EM algorithm to update the new mixture distribution. We simulate draws $\theta^{(i,1)}, \ldots, \theta^{(i,N)}$ from the updated proposal $q_{\zeta(i)}$, and signal paths $x^{(i,1)}, \ldots, x^{(i,N)}$ conditionally on these parameter draws. We compute the corresponding IS weights $w^{(i,1)}, \ldots, w^{(i,N)}$ in (10) using $q_\zeta = q_{\zeta(i)}$.

5. **Evaluate the IS weights**: We calculate the coefficient of variation $CoV^{(i)}$ of the IS weights from the current iteration $w^{(i,1)}, \ldots, w^{(i,N)}$. We terminate the iterations when the coefficient of variation changes by less than 5% compared to the coefficient of variation in the last iteration $CoV^{(i-1)}$; otherwise we go to Step 4.

There is a trade-off between the quality of the proposal and the speed of the estimation procedure. When more draws are used, the approximation generally becomes better, but at the cost of an increased computation time. Fortunately, the draws and corresponding weights can be recycled such that we do not require the sampling of new draws when going through the iterations to obtain the mixture components. To be able to recycle previous draws, we need to implement a slight modification when computing the coefficient of variation of the importance weights that correspond to the latest candidate. Given the draws $(x^{(2,1)}, \theta^{(2,1)}), \ldots, (x^{(2,N)}, \theta^{(2,N)})$ from the proposal $q_{\zeta(2)}(x, \theta|y) = q(x|\theta, y)q_{\zeta(2)}(\theta|y)$ with only one Student's t component, we can evaluate the coefficient of variation of the weights in iteration $i > 2$ based on the new proposal $q_{\zeta(i)}(x, \theta|y)$ by using the mean of the weights as given by

$$\int \frac{p(x,\theta|y)}{q_{\zeta(i)}(x,\theta|y)} q_{\zeta(i)}(x,\theta|y)\mathrm{d}\theta\mathrm{d}x = \int \frac{p(x,\theta|y)}{q_{\zeta(2)}(x,\theta|y)} q_{\zeta(2)}(x,\theta|y)\mathrm{d}\theta\mathrm{d}x, \tag{13}$$

and the variance of the weights as given by

$$\begin{aligned} \int \left(\frac{p(x,\theta|y)}{q_{\zeta(i)}(x,\theta|y)}\right)^2 q_{\zeta(i)}(x,\theta|y)\mathrm{d}\theta\mathrm{d}x &= \int \frac{p(x,\theta|y)^2}{q_{\zeta(i)}(x,\theta|y)}\mathrm{d}\theta\mathrm{d}x \\ &= \int \frac{p(x,\theta|y)}{q_{\zeta(i)}(x,\theta|y)} \frac{p(x,\theta|y)}{q_{\zeta(2)}(x,\theta|y)} q_{\zeta(2)}(x,\theta|y)\mathrm{d}\theta\mathrm{d}x. \end{aligned} \tag{14}$$

Based on these results the mean and variance of the importance weights corresponding to the importance density at iteration $i > 2$ can then be estimated via

$$\text{mean}^{(i)} = \frac{1}{N}\sum_{j=1}^{N} \frac{p(x^{(j)}, \theta^{(j)}|y)}{q_{\zeta(2)}(x^{(j)}, \theta^{(j)}|y)} \tag{15}$$

and

$$\text{var}^{(i)} = \frac{1}{N}\sum_{j=1}^{N} \left[\frac{p(x^{(j)}, \theta^{(j)}|y)}{q_{\zeta(i)}(x^{(j)}, \theta^{(j)}|y)} \frac{p(x^{(j)}, \theta^{(j)}|y)}{q_{\zeta(2)}(x^{(j)}, \theta^{(j)}|y)}\right] - \left[\text{mean}^{(i)}\right]^2, \tag{16}$$

respectively, where $(x^{(j)}, \theta^{(j)}) \sim q_{\zeta(2)}(x, \theta|y)$ is an i.i.d. sequence for $j = 1, \ldots, N$. We emphasize that $q_{\zeta(i)}(x^{(j)}, \theta^{(j)}|y)$ in (16) can be easily evaluated via $q(x|\theta, y)q_{\zeta(i)}(\theta|y)$ as $q(x|\theta, y)$ is the same as in $q_{\zeta(2)}(x, \theta|y)$. This modification of the procedure leads to our modified algorithm, from which we realize a substantial gain in speed; the iterations of our modified algorithm are as follows:

EMitISEM modified algorithm

1. **Initialization**: Same as Step 1 of the EMitISEM algorithm.

2. **Adaptation**: Same as Step 2 of the EMitISEM algorithm.

3. **IS weighted EM algorithm**: Same as Step 3 of the EMitISEM algorithm.

4. **Iterate on the number of mixture components**: We now consider 10% of the simulated draws $\theta^{(2,1)}, \ldots, \theta^{(2,N)}$ that correspond to the highest IS weights, based on the current candidate mixture density $q_{\zeta(i-1)}$. We use this smaller set of draws and weights to compute a new mode and scale matrix. The new mode and scale are used as starting parameters for the additional Student's t component in the mixture. The starting values for the mixture probability and degrees of freedom parameter for the new Student's t component are set to 0.1 and 5, respectively. The starting values of the mixture probabilities for the older Student's t components are obtained by multiplying the latest values by 0.9. Given the last set of N simulated draws $\theta^{(2,1)}, \ldots, \theta^{(2,N)}$ and the corresponding importance weights $w^{(2,1)}, \ldots, w^{(2,N)}$, we apply the IS weighted EM algorithm to update the new mixture distribution to obtain $q_{\zeta(i)}$.

5. **Evaluate the IS weights**: We estimate the coefficient of variation $CoV^{(i)}$ using formulas (15) and (16) . We terminate the iterations when the coefficient of variation changes by less than 5% compared to the coefficient of variation in the last iteration $CoV^{(i-1)}$; otherwise we go to Step 4.

2.2 EMitISEM in case of non-Gaussian state equations

In the previous section we have shown how we can use our approach to estimate the parameters and states of a nonlinear, non-Gaussian state space model with linear Gaussian state equation. This assumption can be somewhat restrictive. Here we present a brief sketch of how we can estimate the model parameters for models with a nonlinear, non-Gaussian state equation, that is

$$y_t \sim p_y(y_t|\alpha_t;\theta), \tag{17}$$

$$\alpha_t \sim p_\alpha(\alpha_t|\alpha_{t-1};\theta). \tag{18}$$

To obtain a Gaussian approximation of the state can be problematic because the smoothed state density can be severely different from the Gaussian density. When the transition density (18) differs from the normal, the approximation is less accurate. However we can still use the fact that using an unbiased estimator of the likelihood $\hat{p}(y|\theta)$ instead of the true likelihood $p(y|\theta)$ in (4) yields a consistent estimator of the Kullback-Leibler divergence between $p(\theta|y)$ and $q_{\zeta'}(\theta|y)$. Hence as long as we can compute $\hat{p}(y|\theta)$,

we can use the original MitISEM approach but with the replacement of the likelihood by its unbiased estimator in the calculation of the weight.

$$\hat{w}^{(j)} = \frac{\hat{p}(\theta^{(j)}|y)}{q_{\zeta}(\theta^{(j)}|y)}. \tag{19}$$

Fortunately, an unbiased estimator of the likelihood $\hat{p}(y|\theta)$ can be calculated with a particle filter in the model given by equation (17) and (18).

EMitISEM algorithm for non-Gaussian state

1. **Initialization**: Simulate a series of N parameter vector draws $\theta^{(0,1)}, \ldots, \theta^{(0,N)}$ from the initial Student's t distribution $q_{\zeta(0)}(\theta|y)$ with its mode equal to the likelihood estimates of θ using the smooth particle filter to evaluate the likelihood by Pitt (2002) and with its scale equal to minus the inverse Hessian of the log likelihood evaluated at the current parameter estimates. Conditionally on the draws $\theta^{(0,1)}, \ldots, \theta^{(0,N)}$ we evaluate the likelihood by running N particle filters in parallel. Finally, we evaluate the estimates of the importance sampling weights $\hat{w}^{(0,1)}, \ldots, \hat{w}^{(0,N)}$ given by (19).

2. **Adaptation**: Same as Step 2 of the EMitISEM algorithm, but with the estimated weights $\hat{w}^{(1,1)}, \ldots, \hat{w}^{(1,N)}$ form (19) replacing the weights from (10).

3. **IS weighted EM algorithm**: Same as Step 3 of the EMitISEM algorithm, but with the estimated weights $\hat{w}^{(1,1)}, \ldots, \hat{w}^{(1,N)}$ form (19) replacing the weights from (10).

4. **Iterate on the number of mixture components**: Same as Step 4 of the EMitISEM algorithm, but with the estimated weights $\hat{w}^{(1,1)}, \ldots, \hat{w}^{(1,N)}$ form (19) replacing the weights from (10).

5. **Evaluate the IS weights**: Same as Step 5 of the EMitISEM algorithm, but with the estimated weights $\hat{w}^{(1,1)}, \ldots, \hat{w}^{(1,N)}$ form (19) replacing the weights from (10).

2.3 Discussion and relation to other methods

In this section we discuss the possible computational gains from our procedure and relate it to previously proposed methods.

2.3.1 Gaussian transition density

The proposal density constructed via EMitISEM can be used in importance sampling or in an independent Metropolis-Hastings procedure. Using the EMitISEM proposal in

an independent Metropolis-Hastings iteration and the method proposed by Pitt et al. (2012) are similar in spirit. The independent Metropolis-Hastings algorithm is the core of both algorithms. However, there are several clear differences between the two estimation routines. First, our method is not adaptive, which means that it is easier to parallellize as the proposal density is constant throughout the MCMC phase. Candidate draws are fully independent both in the training phase and in the MCMC phase. Second, in the Gaussian transition density case we sample one state path at each iteration; we do not need to integrate out the state. We also emphasize that our method in this case directly provides the smoothed state estimates; we do not require additional algorithms for this task. Finally, we use a mixture of Student's t-distributions instead of a mixture of normals to approximate the posterior distribution of the parameters.

Since the most intensive part of the computations in our algorithm is the generation of the signal paths x conditional on the parameters and observations, it is interesting to compare the number of signal draws that are required in our algorithm and in the PMCMC based methods. Our modified EMitISEM procedure requires $3 \times N + I$ draws where N is the size of the training sample and I is the number of iterations in the MCMC phase. In contrast, the PMCMC methods require $I \times S$ signal draws where S is the number of draws used to estimate the likelihood. In the two examples in our simulation and empirical studies, we use I =25,000, N =10,000 and $S = 50$. It implies that PMCMC requires around 20 times more signal paths than our modified EMitISEM method, whereas our modified EMitISEM method yields a better accuracy in less computation time in these two examples.

The choice of the mixture of Student's t-densities as a proposal for the posterior of the parameters has several theoretical and practical advantages over other choices. First, under certain regularity conditions any density can be approximated by a mixture of Student's t-densities if we use a sufficient number of mixture components as shown by Zeevi and Meir (1997). Second, sampling from a mixture of Student's t-densities is fast. Third, the fat tails make the Student's t-distribution (with small enough degrees of freedom) a robust importance sampler. We are less prone to importance weights with an infinite variance, so posterior estimates are more reliable and more efficient. Finally, the construction of Student's t-distributions in the mixture and the mixing weights can be carried out efficiently by means of the extension of the MitISEM procedure.

The exposition above concentrated on the case of a univariate signal. However, the method can be extended when the observation and signal are vectors by using the approach of Scharth (2012). This extended version of the NAIS method is able to treat the signal vector via the use of quasi-random numbers for the numerical evaluation of the variance

of the log weights and subsequently for its minimization.

A promising feature of EMitISEM is that the evaluation of marginal likelihoods can take place in a straightforward manner via importance sampling. Given the proposal density obtained in the training phase we can approximate the marginal likelihood by

$$p(y) = \int \frac{p(y|x,\theta)p(x|\theta)p(\theta)}{q_\zeta(x,\theta|y)} q_\zeta(x,\theta|y)\mathrm{d}\theta \approx \frac{1}{N}\sum_{j=1}^{N} w^{(j)}, \tag{20}$$

where $w^{(j)}$ is defined as the right-hand side of (10), and N is the number of draws used to evaluate the marginal likelihood.

We are faced with a possible limitation of our method when the time dimension T increases. In this case the variance of the importance weights also increases and the Monte Carlo approximation (8) may become less reliable. However, we point out that our simulated and empirical data sets have a considerable size, consisting of 1,260 trading days (for the estimation of a stochastic volatility model) or 40 years of defaults (for the estimation of a stochastic intensity model). For both datasets the EMitISEM works smoothly and without particular difficulties.

2.3.2 Non-Gaussian transition density

Our EMitISEM proposal density can be used as a proposal in an independent particle Metropolis-Hastings iteration. This method is conceptually close to the method proposed by Duan and Fülöp (2013). In a similar set-up, they suggest to integrate out the states using a particle filter. Their move step utilizes an independent particle Metropolis-Hastings step. However, they approximate the posterior density of the parameters by a Sequential Monte Carlo sampler, which forms a bridge between the prior and the posterior density. Our method instead takes advantage of the fact that we can estimate the likelihood, such that we have a reasonable initial candidate for the posterior of the parameters early on in the procedure.

3 Simulation study

3.1 EMitISEM proposal for Independent Metropolis-Hastings

We base our analysis on an independent Metropolis-Hastings sampler; see Metropolis et al. (1953) and Hastings (1970) for the original contributions. We draw from the joint posterior density of the parameters and states $p(x,\theta|y)$. Our procedure consists of two phases: the training phase and the Markov chain Monte Carlo (MCMC) phase.

In the training phase we use the EMitISEM for the construction of a proposal density that approximates the joint posterior $p(x, \theta|y)$. We construct the approximation from proposal densities $q(x|\theta, y)$ and $q_{\zeta'}(\theta|y)$, where $q(x|\theta, y)$ is the conditional proposal density of x given θ, and $q_{\zeta'}(\theta|y)$ is the marginal proposal density for θ and obtained using EMitISEM. We take $q_{\zeta'}(\theta|y)$ as a mixture of Student's t-densities and use it as an approximation of $p(\theta|y)$, We take $q(x|\theta, y)$ as a Gaussian density from the numerically accelerated importance sampling (NAIS) method of Koopman et al. (2014). NAIS is numerically more efficient than alternative approximations such as the ones proposed by Richard and Zhang (2007), Shephard and Pitt (1997), or Durbin and Koopman (1997).

In the Markov chain Monte Carlo phase we use the candidate as the proposal density in an independent Metropolis-Hastings algorithm to draw from the joint posterior density $p(x, \theta|y)$. We sample the joint candidate draws $(\theta^{(j)}, x^{(j)})$ by first sampling

$$\theta^{(j)} \sim q_{\zeta'}(\theta|y), \tag{21}$$

and then, conditioning on $\theta^{(j)}$, sampling

$$x^{(j)} \sim q(x|\theta^{(j)}, y). \tag{22}$$

Let $(\theta^{(i-1)}, x^{(i-1)})$ and (θ^{+}, x^{+}) denote the previous accepted draw of the Markov chain and the new candidate draw, respectively. We set $(\theta^{(i)}, x^{(i)}) = (\theta^{+}, x^{+})$ with probability

$$\alpha = \min \left\{ \frac{p\left(\theta^{+}, x^{+}|y\right) q(x^{(i-1)}|\theta^{(i-1)}, y) q_{\zeta'}(\theta^{(i-1)}|y)}{p\left(\theta^{(i-1)}, x^{(i-1)}|y\right) q(x^{+}|\theta^{+}, y) q_{\zeta'}(\theta^{+}|y)}, 1 \right\}, \tag{23}$$

and $(\theta^{(i)}, x^{(i)}) = (\theta^{(i-1)}, x^{(i-1)})$ otherwise.

We carry out a detailed simulation experiment to demonstrate the performance of our estimation procedure against two alternative procedures. We estimate parameters for a stochastic volatility model and for a stochastic intensity model using simulated data sets. The stochastic volatility model is well known and provides an important benchmark model with many challenges for parameter estimation, see the discussions in Shephard (2005). Similar challenges emerge for the stochastic intensity model, but this model also illustrates a new and interesting application of a nonlinear non-Gaussian state space model for which the full conditional density of the parameter vector is not known in closed form. The stochastic intensity models are particularly useful in portfolio credit risk modeling, see Koopman et al. (2008), Duffie et al. (2009) and Azizpour et al. (2010) for interesting illustrations of the problem. The details of the stochastic volatility and stochastic intensity models are given in Sections 3.3 and 3.4, respectively.

We compare the performance of our proposed EMitISEM method with state-of-the-art alternatives rather than with some feeble benchmark procedures. In particular, we

compare the performance of parameter estimation using the new EMitISEM method versus using two competing methods of Pitt et al. (2012). The first competing method is the adaptive random walk Metropolis-Hastings (ARWMH) algorithm and is an extension of the method of Roberts and Rosenthal (2009). Our second benchmark method is the adaptive independent Metropolis-Hastings (AIMH) algorithm, where the proposal is a mixture of normals. It is an extension of the method of Giordani and Kohn (2010). These two recently developed and advanced methods provide fast and efficient solutions to parameter estimation for nonlinear non-Gaussian state space models by taking advantage of the powerful framework provided in Andrieu et al. (2010). To further enhance the numerical efficiency of the benchmark methodologies, we use a modified version of these methods by introducing the numerically accelerated importance sampling (NAIS) method of Koopman et al. (2014) for integrating out the signal vector. For the purpose of likelihood estimation, the NAIS method is used as an alternative to the partially adapted auxiliary particle filter of Pitt et al. (2012).

We use NAIS as a state sampler for the following three reasons: (i) it can provide an approximation to the state smoothing density that minimizes the variance of the log importance weights; (ii) the approximating Gaussian linear state space model can be constructed in a computationally efficient way by taking advantage of standard Kalman filter methods and deterministic integration methods for one-dimensional integrals; (iii) the simulated signal paths can be efficiently computed via the simulation smoothers of de Jong and Shephard (1995) or Durbin and Koopman (2002). We have found that NAIS yields estimates of the likelihood with lower variance in less computing time than methods based on the particle filter. The findings are discussed in Section 3.2. In Appendix B we provide the details of NAIS. For its use in the ARWMH and AIMH methods, we use 50 simulated paths of the signal for likelihood estimation. We notice that EMitISEM requires one simulated signal path from NAIS only at each iteration. Further implementation details of the competitive benchmark methods are discussed in Appendices C and D.

We estimate the parameters for 56 data sets on an 8-core computer. The data sets are generated with parameter values that are close to those estimated from the empirical data sets of Section 4. For each simulated data set, we re-estimate parameters by using the EMitISEM method, its modified version (which we denote by EMitISEM mod.), the ARWMH method and the AIMH method. For the modified EMitISEM method, the candidate draws are recycled after the first MitISEM update in the training phase. After 5,000 burn-in draws we perform 20,000 iterations of the algorithms. We calculate medians and interquartile ranges (over the 56 simulated data sets) of the parameter estimates, acceptance rates, and inefficiencies. In order to assess the quality of the simulation methods

we compute the inefficiency factor (IF), which is defined as the variance of the parameter estimate divided by the variance in case the sampling scheme would generate independent posterior draws. The IF statistic is discussed, amongst others, by Pitt et al. (2012). In our case, we define the inefficiency factor as

$$IF = 1 + 2 \sum_{j=1}^{\max(L,1000)} r_j, \tag{24}$$

where r_j is the j-th order sample serial correlation of the 20,000 parameter draws, and where L is the lowest order j for which r_j is not significant.

3.2 Likelihood estimation: NAIS versus particle filters

We have argued that ARWMH and AIMH methods can be implemented using both PMCMC and NAIS algorithms for drawing the signal vectors. To assess the difference between the two implementations, we use the particle filter and NAIS methods to evaluate the likelihood function. A review of different particle filtering methods is provided in for instance Doucet et al. (2001). For the case of a stochastic volatility model, we obtain more efficient likelihood estimates when using NAIS in comparison to using particle filters. The NAIS importance sampling estimates of the likelihood function have lower variance and need less computing time than the particle filter likelihood estimates.

We simulate 56 data sets using the same data generation process for the stochastic volatility model of Section 3.3. We estimate the likelihood value at the "true" parameter values 100 times for each simulated data set using the bootstrap filter of Gordon et al. (1993), the auxiliary particle filter of Pitt and Shephard (1999) and the NAIS method of Koopman et al. (2014). We compute 100 likelihood estimates for each data set and we calculate the variance of the estimates together with the mean computing time for each data set. We report the median variance and computing times over the 56 data sets. Table 1 presents the results. For all considered time series lengths, the median variance of the NAIS estimate is lower than the median variance of the particle filter estimates for all numbers of particles considered. Moreover, the estimation using importance sampling takes much less time than the estimation using particle filters. We therefore use the NAIS in all algorithms to facilitate fair comparisons.

3.3 Stochastic volatility model

Many macroeconomic and financial time series exhibit volatility clustering, which results in autocorrelated time varying variances and volatilities. To capture autocorrelation in

Table 1: Comparison of the variance of the particle filter and importance sampling estimate of the likelihood in the SV model based on 56 simulated data sets. We compare the likelihood estimates from the bootstrap filter (BF), the auxiliary particle filter (APF) and the numerically accelerated importance sampling (NAIS). The number of particles/number of draws is denoted by M and the length of the simulated data set is denoted by T. For different M and T we report the median of the variances over the 56 data sets, where these variances are calculated from 100 runs per data set.

Method		Variance		Time	
	M	T =1,000	T =2,000	T =1,000	T =2,000
BF	250	1.079	1.505	5.970	37.901
	500	0.743	1.071	18.435	88.828
	1000	0.541	0.735	44.018	173.432
APF	250	0.988	1.420	6.362	37.434
	500	0.674	0.963	21.025	88.764
	1000	0.490	0.687	42.663	195.456
NAIS	50	0.140	0.233	0.283	0.527

volatilities we adopt the stochastic volatility (SV) model. A basic specification of the stochastic volatility model is given by

$$y_t = \exp(x_t/2)\varepsilon_t, \qquad \varepsilon_t \sim \text{NID}(0,1), \tag{25}$$

$$x_t - \delta = \phi(x_{t-1} - \delta) + \eta_t, \qquad \eta_t \sim \text{NID}\left(0, \sigma_\eta^2\right), \tag{26}$$

where x_t is the unobserved log-volatility process, which follows an autoregressive process of order 1, ε_t is a standardized error term, δ is the overall mean of x_t, $0 < \phi < 1$ is the persistence parameter, and $\sigma_\eta^2 > 0$ is the innovation variance of the log-volatility process. The three unknown parameters δ, ϕ, and σ_η^2 need to be estimated. More discussions on the SV model and its extensions can be found in Kim et al. (1998) and Shephard (2005).

The 56 data sets are generated from the basic SV model (25) and (26) with parameters set equal to $\delta = 0.48$, $\phi = 0.97$, and $\sigma_\eta^2 = 0.049$, which correspond closely to the estimates obtained in our empirical study in Section 4. We simulate time series of length 1,250 and use the following prior specifications (see Omori et al. (2007)) for the parameters

$$\delta \sim \text{N}(0,1), \qquad \frac{\phi+1}{2} \sim \text{Beta}(20, 1.5), \qquad \frac{1}{\sigma_\eta^2} \sim \text{Gamma}\left(\frac{5}{2}, \frac{0.05}{2}\right).$$

Table 2 presents the medians and interquartile ranges of acceptance rates, parameter estimates and inefficiency factors. We also present the estimated posterior means of the parameters to indicate that the different simulation methods provide similar results.

The results suggest that none of the methods considered produce biased estimates due

Table 2: Performance for the stochastic volatility model: We compare the Adaptive Random Walk Metropolis-Hastings method (ARWMH), the Adaptive Independent Metropolis-Hastings method (AIMH) , the EMitISEM with 10,000 parameter draws (EMitISEM 10k), the modified EMitISEM with 10,000 parameter draws (EMitISEM mod. 10k) and the modified EMitISEM with 2,000 parameter draws (EMitISEM mod. 2k) in terms of time (in sec), acceptance rates(AR), parameter estimates, and inefficiency factors (24). The table presents the medians and interquartile ranges (within parentheses) over 56 simulated data sets. The estimates are based on 20,000 draws after a burn-in sample of 5,000 observations. We use 50 simulated draws to evaluate the likelihood in the ARWMH and AIMH algorithms.

Algorithm	Time	AR	Estimate			Inefficiency		
			δ	ϕ	σ_η^2	δ	ϕ	σ_η^2
ARWMH	8401	0.305	0.445	0.971	0.049	14.538	14.312	13.265
	(485)	(0.017)	(0.195)	(0.014)	(0.016)	(2.588)	(2.151)	(2.079)
AIMH	8020	0.604	0.445	0.971	0.049	2.905	2.979	2.900
	(277)	(0.092)	(0.204)	(0.014)	(0.016)	(0.687)	(1.719)	(1.390)
EMitISEM 10k	3526	0.507	0.450	0.970	0.048	4.833	5.070	5.561
	(1062)	(0.065)	(0.195)	(0.014)	(0.016)	(1.106)	(2.139)	(2.210)
EMitISEM mod.10k	2406	0.509	0.442	0.971	0.049	5.113	5.056	5.572
	(131)	(0.071)	(0.195)	(0.014)	(0.016)	(2.072)	(1.934)	(2.550)
EMitISEM mod. 2k	1417	0.501	0.444	0.971	0.049	6.354	5.471	4.970
	(75)	(0.061)	(0.201)	(0.013)	(0.017)	(3.398)	(3.324)	(2.818)

to the omission of relevant parts of the parameter space. Therefore we focus on comparing alternative methods in terms of accuracy and computing time.

The interquartile ranges of the estimated posterior means are similar among the different methods. This finding does not imply that for example ARWMH is performing as well as the other methods, because the interquartile ranges are mainly driven by the differences between the 56 simulated data sets and in particular by the differences between the 56 true posterior means. The reported inefficiency factors (IF) in the last three columns of the table suggest that the ARWMH method is close to 3-4 times less efficient than the competing methods. The medians of the inefficiency factor and the acceptance rates show that the AIMH is successful in approximating the posterior density of the parameters. The interquartile range of the inefficiency factors are indicative of the robustness of the methods across different data sets. The EMitISEM methods perform slightly worse compared to the AIMH approach for both acceptance rates and inefficiencies. However, the EMitISEM methods requires substantially less computing time. The modified version of EMitISEM provides a slightly lower quality proposal than the standard EMitISEM, but comes with a clear reduction in computing time. Finally, the results for modified EMitISEM with a training sample of only 2,000 draws (instead of the default of 10,000 draws) suggest that we are able to obtain further efficiency gains at the cost of a moderate

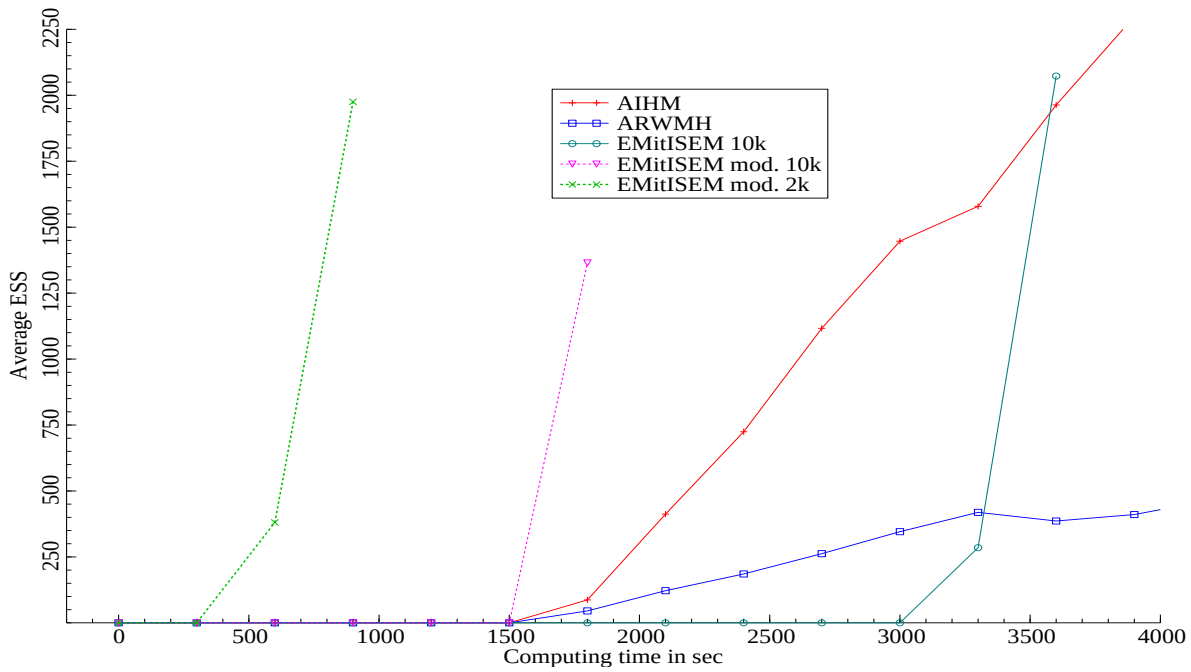

Figure 1: Performance for the stochastic volatility model: average effective sample size per computing time (in seconds). The average (over the three parameters) of the $ESS(s)$ is computed for one simulated data set for the different estimation procedures. The average $ESS(s)$ is calculated every five minutes, after which linear interpolation gives a crude approximation of the average $ESS(s)$ as a function of computing time.

loss of robustness of the procedure.

To obtain further insight in the efficiency of the estimation procedures, we look at the trade-off between the inefficiency factor and computing time. We obtain a crude approximation of the effective sample size as a function of computing time. During the estimation process, after each five-minute period, we approximate the effective sample size by $ESS(s) = N(s)/IF(s)$, where $N(s)$ and $IF(s)$ are the number of draws (after the discarded burn-in sample) and the inefficiency factor in period s, respectively. We report the average $ESS(s)$ for the three parameters for a randomly chosen simulated data set in Figure 1. We obtain similar patterns for the other simulated data sets.

For the AIMH and ARWMH algorithms it takes five periods of five minutes (1,500 seconds) to draw the burn-in sample. After 1,500 seconds, the average $ESS(s)$ value starts to increase. The average $ESS(s)$ for AIMH increases more steeply than for ARWMH since the inefficiencies are higher for ARWMH, while the computing times are similar because both methods integrate out the state vector at each iteration. The $ESS(s)$ slopes for EMitISEM are much steeper than for AIMH and ARWMH. This confirms the efficiency and good acceptance rates of EMitISEM. A smaller size of the training sample or the use of the modified version of the EMitISEM algorithm clearly lead to further efficiency gains.

3.4 Stochastic intensity model

For our second simulation experiment, we consider a stylized version of the point processes model with stochastic intensity as used in our second empirical application in Section 4.2. Koopman et al. (2008) and Duffie et al. (2009) consider the stochastic intensity model for studying the systematic dynamics of U.S. corporate defaults and credit rating migrations.

n this section we only provide the necessary details of the model needed for the simulations. The full model set-up is deferred till Section 4.2. For the simulation, we use the following version of the model. We consider a pool of K firms and a jump process $y_k(t)$ for each firm $k = 1, \ldots, K$ with common jump intensity $\lambda(t)$ as given by

$$\lambda(t) = \exp\left[\omega + \beta' c(t) + \gamma x(t)\right], \tag{27}$$

where ω is the base log-intensity, β is a vector of regression parameters, $c(t)$ is a vector of covariates and γ is a scale factor for the unobserved signal $x(t)$. The cumulative jump process over all firms is given by

$$y(t) = \sum_{k=1}^{K} y_k(t). \tag{28}$$

The signal $x(t)$ is often referred to as an unobserved frailty factor. We follow standard practice and model it as a zero mean Ornstein-Uhlenbeck (OU) process, standardized to have unit variance at $t = 1$,

$$\mathrm{d}x(t) = -\rho x(t)\,\mathrm{d}t + \sqrt{2\rho}\,\mathrm{d}W(t), \tag{29}$$

where $\rho > 0$ is a persistence parameter and $W(t)$ is a standard Brownian motion. The set of covariates we use in the simulation is the same as in the empirical section, namely the (i) one year difference of the S&P500 index, (ii) term spread between the 10-year and 1-year Treasury Bond (with constant maturity rates), (iii) secondary market rate on 3 month Treasury Bills, and (iv) year-to-year percentage change of US industrial production (final output), all at the monthly frequency over the period from January 1, 1970 to March 4, 2010; compare Duffie et al. (2007), Lando and Nielsen (2010) and Azizpour et al. (2010). The covariates are obtained from the FRED and CRSP databases. We set the parameters to $\omega = -4.75$, $\beta = (-0.85, 0.01, -0.055, -5.1)$, $\gamma = 1.15$ and $\rho = 0.12$, which are close to the empirical estimates from Section 4.

We simulate data for K =3,000 firms over the period January 1, 1970 to March 4, 2010. As K is kept fixed, a firm can jump repeatedly over the sample. If a jump is interpreted as default, this implies that the firm is directly re-started after default at the same pre-default intensity. In the empirical application, we depart from this construction

and allow for an absorbing default state as well as for firms that enter the sample or leave the sample for other reasons than default.

The simulations are conditional on the four covariates and are sampled by using a discretization of the continuous time processes $y(t)$, $x(t)$ and $\lambda(t)$, where the discretization takes steps of $1/32$ part of a day, i.e., 45 minutes. Over each 45 minutes slot, we use a Bernoulli approximation to generate defaults. We generate 56 data sets in this way. To the estimation process, we use (weakly informative) uniform priors on relatively wide intervals: $[-8, -2]$ for ω, $[0.01, 3]$ for γ, $[0.01, 1]$ for ρ, and $[-20, 20]$ for each of the four elements of β.

We consider the ith event time t_i and define the indicator variable D_{ki} to be one, $D_{ki} = 1$, if firm k jumps to default at the ith event time t_i, and zero otherwise. The number of jumps at event time t_i over all firms is given by $D_i = \sum_{k=1}^{K} D_{ki}$. The discrete time approximation of the jump process $y(t)$ leads to the following dynamic model in event time,

$$\begin{aligned} p(y_i|x_i, \theta) &= \exp\left[D_i \log \lambda_i - \lambda_i K \Delta_i\right], \\ x_i &= e^{-\rho\Delta_i} x_{i-1} + \eta_i, \qquad \eta_i \sim \mathrm{N}\left(0, 1 - e^{-2\rho\Delta_i}\right), \end{aligned} \tag{30}$$

where $p(y_i|x_i, \theta)$ is the density of $y_i = y(t_i)$ conditional on signal $x_i = x(t_i)$ and parameter vector θ, with $\lambda_i = \lambda(t_i)$ and $\Delta_i = t_i - t_{i-1}$. Further details of the model are presented at the empirical application in Section 4.2.

Table 3 presents the means and interquartile ranges of the parameter estimates for the 56 simulated data sets. The different simulation methods provide similar results, which suggests that none of the methods provides biased estimates. Table 4 presents the means and interquartile ranges of acceptance rates and inefficiency factors. We find that the ARWMH algorithm is clearly outperformed by the other methods. Moreover, for the stochastic intensity model, the AIMH algorithm performs generally less favourable compared to EMitISEM. Further, the AIMH method appears to be less robust for certain simulated data sets. The median inefficiencies are higher and also the interquartile ranges of the inefficiencies are larger compared to EMitISEM. A possible explanation is that AIMH uses a mixture of normal distributions, where the thin tails of the normal distribution imply that the AIMH algorithm can sometimes get stuck for a considerable amount of time when a rare candidate draw is simulated from the tails and accepted, after which many consecutive candidate draws are rejected and the same accepted draw is repeated many times. In contrast, the EMitISEM method uses a mixture of Student's t-densities to approximate the posterior distribution of the parameters. The fat tails of the Student's t-density prevent that the MH method repeats a draw from one of the tails

Table 3: Performance for the stochastic intensity model: parameter estimates for different Metropolis-Hastings algorithms. The table presents the medians and interquartile ranges (within parentheses) over 56 simulated data sets. The estimates are based on 20,000 draws after a burn-in sample of 5,000 observations. We use 50 simulated draws to evaluate the likelihood in the ARWMH and AIMH algorithms.

	Estimates						
	ω	β_1	β_2	β_3	β_4	γ	ρ
True values	-4.75	-0.85	0.01	-0.055	-5.1	1.15	0.12
Algorithm:							
ARWMH	-4.559	-0.909	0.021	-0.044	-4.611	1.307	0.154
	(1.212)	(0.498)	(0.202)	(0.107)	(2.280)	(0.400)	(0.111)
AIMH	-4.621	-0.895	0.026	-0.043	-4.546	1.256	0.149
	(1.247)	(0.501)	(0.206)	(0.103)	(2.330)	(0.459)	(0.111)
EMitISEM 10k	-4.624	-0.896	0.025	-0.044	-4.591	1.286	0.151
	(1.183)	(0.483)	(0.209)	(0.108)	(2.182)	(0.382)	(0.106)
EMitISEM mod.10k	-4.580	-0.899	0.024	-0.044	-4.562	1.283	0.150
	(1.220)	(0.482)	(0.211)	(0.105)	(2.219)	(0.389)	(0.107)
EMitISEM mod. 5k	-4.599	-0.897	0.023	-0.044	-4.559	1.302	0.145
	(1.189)	(0.477)	(0.206)	(0.106)	(2.115)	(0.394)	(0.106)

for a long sequence of iterations. The performance of the modified versions of EMitISEM are again comparable to the standard version. According to the results in Table 4, the size of the training sample can be reduced to obtain higher efficiency gains.

Figure 2 presents the average $ESS(s)$ per computing time for the different methods and for a randomly chosen data set. We can see that the relative performance of the alternative methods for the stochastic intensity model is similar to that for the stochastic volatility model. After the burn-in draws are computed in the first 3,000 seconds, the average $ESS(s)$ for the AIMH and ARWMH methods starts to increase relatively slowly compared to that of EMitISEM. It shows that the EMitISEM methods are computationally more efficient. They outperform the alternative methods both in terms of computing speed and in terms of the fit of the proposal density. Furthermore, Figure 2 shows for this particular data set that the average $ESS(s)$ of AIMH only increases steadily after 6,300 seconds. The decrease and standstill of the estimated average $ESS(s)$ between 5,100 and 6,300 seconds is caused by having a long sequence of draws at a particular parameter value in a remote part of the posterior distribution that had not yet been explored by earlier draws (that were simulated in the first 5,100 seconds).

We conclude from the results presented for both the stochastic volatility and stochastic intensity model that using the EMitISEM algorithm to construct a proposal density

Table 4: Performance for the stochastic intensity model: We compare the Adaptive Random Walk Metropolis-Hastings method (ARWMH), the Adaptive Independent Metropolis-Hastings method (AIMH) , the EMitISEM with 10,000 parameter draws (EMitISEM 10k), the modified EMitISEM with 10,000 parameter draws (EMitISEM mod. 10k) and the modified EMitISEM with 2,000 parameter draws (EMitISEM mod. 2k) in terms of time (in seconds), acceptance rates(AR), parameter estimates, and inefficiency factors (24). The table presents the medians and interquartile ranges (within parentheses) over 56 simulated data sets. The estimates are based on 20,000 draws after a burn-in sample of 5,000 observations. We use 50 simulated draws to evaluate the likelihood in the ARWMH and AIMH algorithms.

Algorithm	Time	Acc.	Inefficiency						
	(in sec)	Rate	ω	β_1	β_2	β_3	β_4	γ	ρ
ARWMH	10053	0.247	41.833	31.735	29.212	29.428	32.037	39.873	33.523
	(7380)	(0.015)	(13.621)	(6.883)	(4.619)	(5.650)	(7.931)	(21.481)	(1.469)
AIMH	8953	0.411	25.256	8.308	10.812	9.439	8.592	26.513	10.777
	(6406)	(0.150)	(181.847)	(22.293)	(12.817)	(13.771)	(15.356)	(135.075)	(39.141)
EMitISEM	5858	0.658	3.431	2.586	2.575	2.624	2.642	3.406	2.893
10k	(3684)	(0.031)	(0.769)	(0.374)	(0.380)	(0.327)	(0.559)	(0.879)	(0.666)
EMitISEM	3124	0.633	4.863	2.900	2.884	2.899	3.024	4.934	3.355
mod. 10k	(2386)	(0.041)	(3.950)	(0.735)	(0.804)	(0.946)	(0.728)	(5.605)	(1.210)
EMitISEM	2089	0.618	5.877	3.265	3.169	3.279	3.303	5.926	4.117
mod. 5k	(1616)	(0.037)	(4.823)	(0.851)	(1.012)	(0.985)	(1.062)	(7.377)	(2.038)

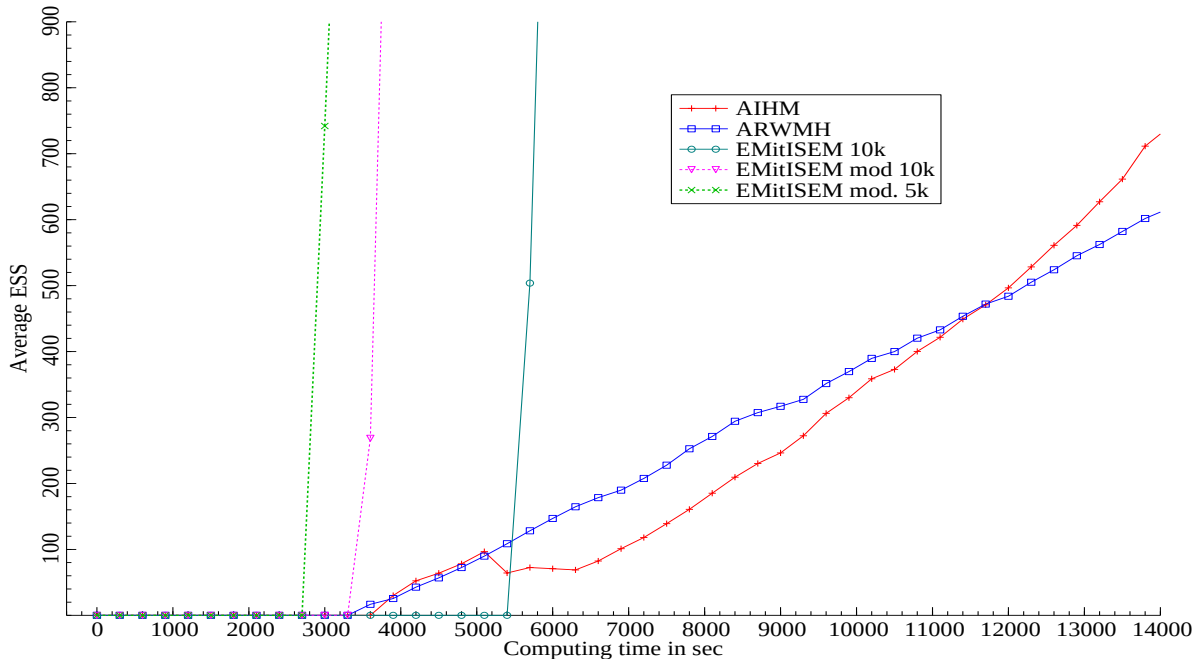

Figure 2: Performance for the stochastic intensity model: average effective sample size per computing time (in seconds). The average (over the seven parameters) of the $ESS(s)$ is computed for one simulated data set for the different estimation procedures. The average $ESS(s)$ is calculated every five minutes, after which linear interpolation gives a crude approximation of the average $ESS(s)$ as a function of computing time.

for independent Metropolis-Hastings iterations offers an efficient alternative to PMCMC. Substantial improvements can be obtained with respect to computational speed, accuracy, numerical efficiency, and robustness.

4 Two empirical studies

In this section we illustrate the performance of the EMitISEM proposal in an Metropolis-Hastings procedure as part of two empirical studies. First, we estimate the parameters and the signal in a stochastic volatility model for a time series of daily IBM stock returns. Second, we estimate the stochastic intensity model using a large panel of U.S. corporate defaults.

4.1 Stochastic volatility model

We consider the stochastic volatility model for IBM stock returns over the period January 3, 2007 to December 30, 2011. The data are obtained from CRSP. The sample period includes the financial crisis at the end of 2008 and the subsequent recession and contains 1,260 observations. The excess kurtosis of 7.004 and skewness of 0.095 indicate that the density of returns is heavy-tailed and nearly symmetrically distributed. The sample autocorrelation functions of the returns and squared returns are shown in Figure 3. The autocorrelations of the squared returns clearly provide evidence that the returns exhibit volatility clustering.

The estimated parameters and the inefficiencies of the chains of EMitISEM draws are reported in Table 5. The estimated parameter values are typical of what is found in similar empirical studies. The unconditional mean of the log volatility process is estimated close to 0.48. The estimated autoregressive coefficient in the state equation is 0.97. These results imply a smooth, highly persistent log-volatility process.

The inefficiency of the posterior draws is 5 for the mean δ and persistence coefficient ϕ, and 10 for the variance σ_η^2. The acceptance rate for EMitISEM is 50.85%. Figure 5 displays the IBM return series along with the smoothed signal estimates. The EMitISEM method provides these smoothed estimates and the corresponding credible interval as a byproduct of the algorithm: they are computed directly from the accepted MH draws of the signal paths. We observe increased volatility levels at the end of 2008, the first half of 2009, and at the end of 2011.

The computation of marginal likelihoods is straightforward in the EMitISEM ap-

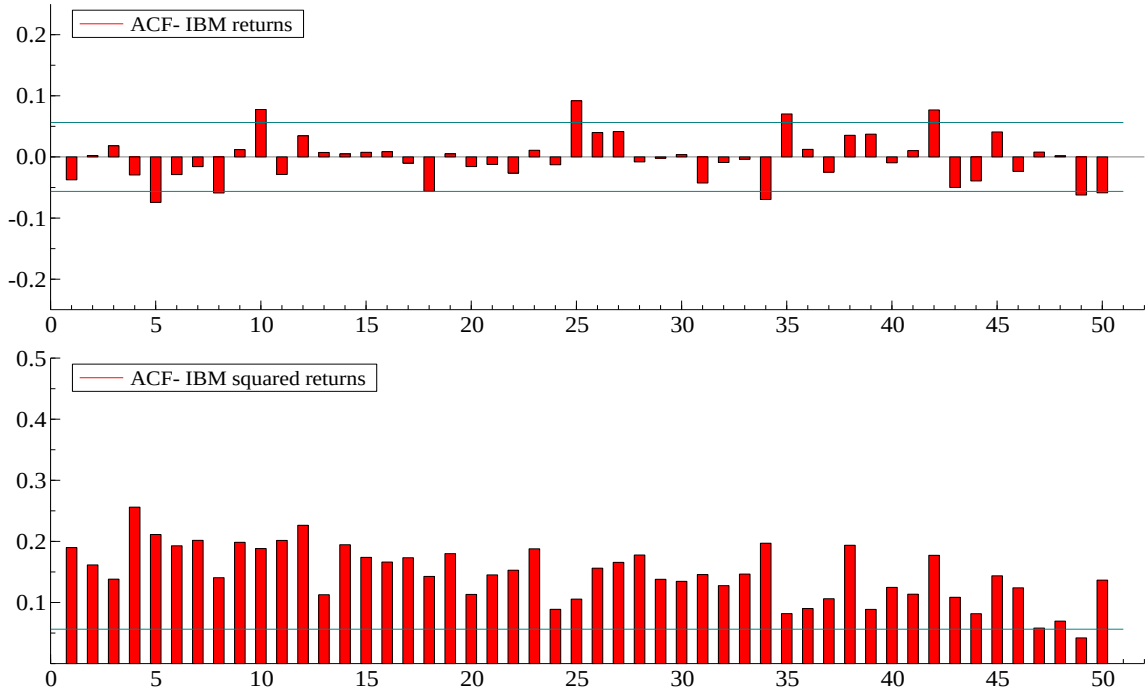

Figure 3: Sample autocorrelation functions for the IBM returns (top panel) and for the squared IBM returns (bottom panel) from January 3, 2007 to December 30, 2011.

proach. We obtain the marginal likelihood estimate as the average of the importance sampling weights, using the result in equation (20). These weights have already been computed during the algorithm to construct the MH acceptance probabilities. As an alternative to the independent Metropolis-Hastings method, we could use the importance sampling approach to estimate the model parameters or smoothed log volatility process $x(t)$ using the candidate draws and corresponding importance weights obtained during the EMitISEM algorithm. Given that the candidate draws are independent, we would not have to discard burn-in draws. The independence of the draws allows us to compute reliable numerical standard errors for the estimated posterior means. However, when using importance sampling the computation of the posterior density of a parameter or the computation of credible intervals for the parameters or states would require additional work. This stems from the fact that the importance sampling method yields a series of weighted candidate draws instead of draws from the posterior itself. In any case, the results from importance sampling and from the independent MH algorithm are typically rather close if based on the same set of candidate draws.

Table 5: Stochastic Volatility model: parameter estimates for IBM returns from January 3, 2007 to December 30, 2011. The 90% credible intervals are within parentheses. The inefficiency factor is computed as (24).

Parameter	Estimate	Inefficiency
δ	0.479	4.979
	(0.065 , 0.875)	
ϕ	0.973	5.849
	(0.956, 0.988)	
σ_η^2	0.049	9.639
	(0.029 , 0.076)	

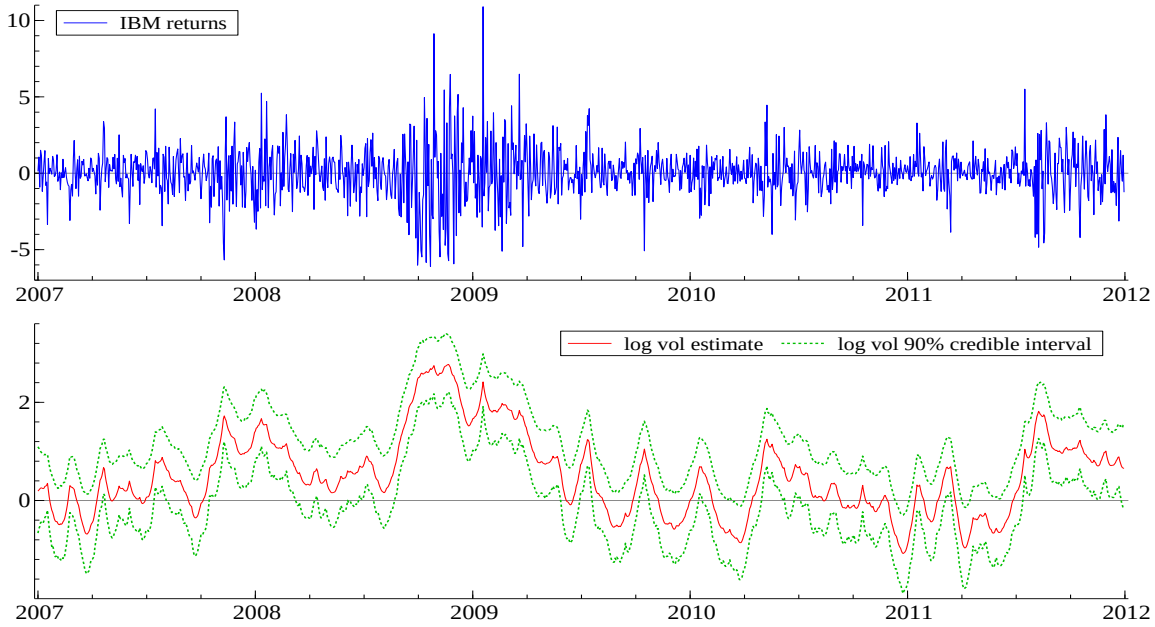

Figure 4: IBM returns from January 3, 2007 to December 30, 2011 (top panel) and the smoothed estimate of the log volatility process with its 90% credible interval (bottom panel).

4.2 Stochastic intensity model

For our second empirical illustration we consider the stochastic intensity model for a large panel data set of U.S. corporate defaults obtained from Moody's. The core of the model is the same as in Section 3.4, with a slight change to account for sample extension and for attrition due to other reasons than default. The dummy variable D_{ki} is defined as before, with $D_{ki} = 1$ if firm k jumps into default at time t_i, and $D_{ki} = 0$ otherwise. We introduce the new dummy variables R_{ki}, with $R_{ki} = 1$ if firm k is at risk of defaulting at time $t_i - \varepsilon$, for $\varepsilon > 0$ arbitrarily small, and $R_{ki} = 0$ otherwise. An event time t_i occurs when one of the control variates changes its value (e.g., at the end of the month or quarter), when a

firm is added to the sample, or when a firm leaves the sample, either due to default or due to other reasons. We denote the default intensity of firm k at time t_i as $\lambda_{ki} = \lambda_k(t_i)$.

The conditional density of the observations given the complete paths of the covariates $c_i = c(t_i), i = 1, \ldots, T$ and the complete path of the unobserved process $x_i = x(t_i), i = 1, \ldots, T$ is given by

$$p(y|x, \theta) = \prod_{i=1}^{T} p(y_i|x_i, \theta) = \prod_{i=1}^{T} \prod_{k=1}^{K} \exp\left[D_{ki} \log \lambda_{ki} - R_{ki}\lambda_{ki}\Delta_i\right], \tag{31}$$

for $y = (y_1, \ldots, y_T)'$ and $x = (x_1', \ldots, x_T')'$. Firm k only contributes to the likelihood function when it is at risk of defaulting, that is when $R_{ki} = R_k(t_i) = 1$. The state equation remains as in Section 3.4, for further details, see Koopman et al. (2008).

Our data set contains 1,627 defaults from 12,881 U.S. firms observed daily over the period January 1, 1970 to March 4, 2010. The number of firms in the portfolio increases over time from about 1,000 firms at the beginning of the sample to around 5,000 firms around 2010. Defaults originating from parent-subsidiary relationships are excluded: if there are multiple defaults with the same parental ID, we only keep the oldest firm as this is likely to be the parent firm. Event times and durations are measured in business days. As in Koopman et al. (2008), we winsorize the number of defaults per day to one to account for outliers and other data irregularities.

The top panel of Figure 5 shows the number of defaults per day. The concentration of the vertical lines clearly presents evidence of default clustering over time. In particular, we find high levels of defaults during 1989-1991 at the end of the savings and loan crisis and during the subsequent recession, in 2001 after the burst of the dot-com bubble, and after the 2008 financial crisis and during the subsequent recession. The increasing numbers of defaults in the last two decades do not immediately imply an increase in the frailty process $x(t)$, because also the number of firms increased substantially in the later part of the sample. Furthermore, the covariates may also partly explain the movements in the data. The covariates are S&P500 returns, Treasury Bond spreads, 3 month Treasury Bill yields, and yearly changes in U.S. industrial production, see also Section 3.4.

The parameter estimates and the inefficiency factors for the chains of the EMitISEM draws are presented in Table 6. The 90% credibility intervals of the coefficients β_2 (for the term spread) and β_3 (for the U.S. Treasury Bill rate) include zero, which indicates that these parameters are not significantly different from zero. The signs of the parameters are consistent with what we expect. Both lower returns on the S&P500 index and lower percentage changes in industrial production imply a higher default intensity. The mean reversion parameter ρ is estimated as 0.12. At the yearly frequency, this implies an autoregressive coefficient $e^{-\rho} \approx 0.9$, such that the frailty process has a high persistence.

Table 6: Stochastic intensity model: parameter estimates for U.S. corporate defaults from January 1, 1970 to March 4, 2010. The 90% credible intervals are within parentheses. The inefficiency factor is calculated as (24).

	Parameter	Estimate	Inefficiency
ω	Constant	-4.750	2.755
		(-6.357 , -3.287)	
β_1	S&P500 1 year return	-0.862	2.097
		(-1.346 , -0.375)	
β_2	Term spread	0.013	3.495
		(-0.164, 0.191)	
β_3	3M TBill	-0.054	2.489
		(-0.167 , 0.058)	
β_4	Change in indust. prod.	-5.115	2.450
		(-7.798 , -2.292)	
γ	Loading on frailty	1.148	2.809
		(0.548 , 2.294)	
ρ	Mean reversion	0.120	2.688
		(0.014 , 0.351)	

The inefficiency factors have values around 3 and the acceptance rate is 71.04%. The bottom panel of Figure 5 displays the smoothed estimate of the frailty process together with the 90% credible interval. We emphasize that the credible interval includes all uncertainties due to the observation noise, the randomness of the frailty process, and the uncertainty about the parameter vector θ. This contrasts with the confidence bands around the estimated frailty process which are shown in most of the literature as part of a classical analysis where the parameter uncertainty is typically ignored. The estimated frailty process represents the credit cycle dynamics in excess of the dynamics caused by the observable controls in $c(t_i)$. We clearly recognize the local peaks of the 1991 recession, the burst of the dot-com bubble, and the aftermath of the financial crisis of 2008.

5 Conclusion

We have introduced a new Extended Mixture of t by Importance Sampling Weighted Expectation Maximization (EMitISEM) algorithm to construct a flexible proposal for the joint posterior density of the parameters and the states in non-linear, non Gaussian state space models. We conclude that using the EMitISEM proposal in an independent Metropolis-Hastings procedure is a computationally efficient alternative to competing MCMC methods such as the adaptive particle independent Metropolis-Hastings method.

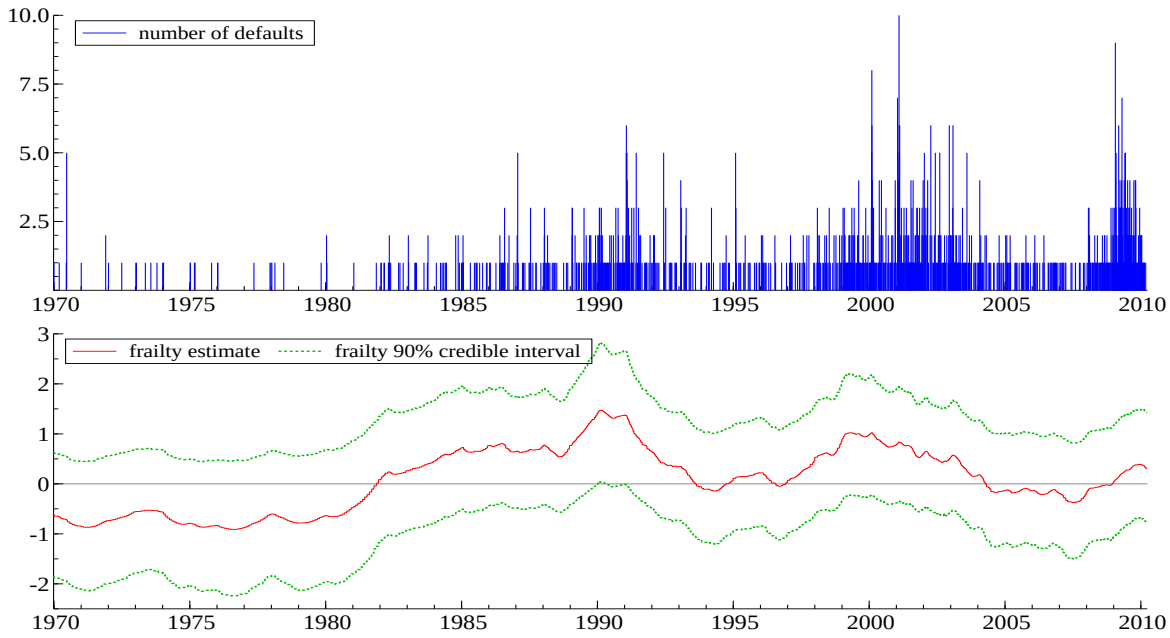

Figure 5: Default data from 1st January 1970 to 4th March 2010 (top panel) and the smoothed estimate of the frailty process with its 90% credible interval (bottom panel).

In a Monte Carlo study, we have shown that our method outperforms competing methods in terms of efficiency and computation time. An interesting extension for future research is to explore the sampling of state paths with the backward smoothing algorithm described in Lindsten and Schön (2012).

Acknowledgements

Preliminary versions of this paper were presented at ESOBE 2012 in Vienna, ERCIM 2012 in Oviedo, and the 3rd Humboldt-Copenhagen Conference on Financial Econometrics in Berlin in 2013. We are indebted to several participants at these meetings for constructive comments. István Barra and André Lucas thank the Dutch National Science Foundation (NWO) for financial support under VICI grant 453-09-005. Lennart Hoogerheide was supported by VENI grant 451-09-028 of the Dutch National Science Foundation (NWO). Koopman acknowledges support from CREATES, Center for Research in Econometric Analysis of Time Series (DNRF78) at Aarhus University, Denmark, and funded by the Danish National Research Foundation.

References

Andrieu, C., Doucet, A., Holenstein, R., 2010. Particle Markov chain Monte Carlo methods. Journal of Royal Statistical Society Series B 72, 269–342.

Azizpour, S., Giesecke, K., Schwenkler, G., 2010. Exploring the sources of default clustering, Working Paper, Stanford University.

Chan, J., Strachan, R., 2012. Estimation in non-linear non-Gaussian state space models with precision-based methods, CAMA Working Paper.

de Jong, P., Shephard, N., 1995. The simulation smoother for time series models. Biometrika 82, 339–350.

Doucet, A., de Freitas, N., Gordon, N. (Eds.), 2001. Sequential Monte Carlo Methods in Practice. Springer Verlag, New York.

Doucet, A., Pitt, M., Kohn, R., 2012. Efficient implementation of MCMC when using an unbiased likelihood estimator, Working Paper.

Duan, J.-C., Fülöp, A., 2013. Density-tempered marginalized sequential monte carlo sampler, Research Paper.

Duffie, D., Eckner, A., Horel, G., Saita, L., 2009. Frailty correlated default. Journal of Finance 64, 2089–2123.

Duffie, D., Saita, L., Wang, K., 2007. Multi-period corporate default prediction with stochastic covariates. Journal of Financial Economics 83 (3), 635–665.

Durbin, J., Koopman, S. J., 1997. Monte Carlo maximum likelihood estimation for non-Gaussian state space models. Biometrika 84, 669–684.

Durbin, J., Koopman, S. J., 2002. A simple and efficient simulation smoother for state space time series models. Biometrika 89, 603–616.

Flury, T., Shephard, N., 2011. Bayesian inference based only on simulated likelihoood. Econometric Theory 27, 933–956.

Giordani, P., Kohn, R., 2010. Adaptive independent Metropolis-Hastings by fast estimation of mixtures of normals. Journal of Computational and Graphical Statistics 19, 243–259.

Gordon, N., Salmond, D. J., Smith, A. F. M., 1993. A novel approach to non-linear and non-Gaussian Bayesian state estimation. IEE-Proceedings F 140, 107–113.

Hastings, W. K., 1970. Monte Carlo sampling methods using Markov chains and their applications. Biometrika 57, 97–109.

Hoogerheide, L. F., Opschoor, A., van Dijk, H. K., 2012. A class of adaptive importance sampling weighted EM algorithms for efficient and robust posterior and predictive simulation. Journal of Econometrics 171 (2), 101–120.

Kim, S., Shephard, N., Chib, S., 1998. Stochastic volatility: Likelihood inference and comparison wih ARCH models. Review of Economic Studies 65, 361–393.

Koopman, S. J., Lucas, A., Monteiro, A., 2008. The multi-state latent factor intensity model for credit rating transition. Journal of Econometrics 142, 399–424.

Koopman, S. J., Lucas, A., Scharth, M., 2014. Numerically accelerated importance sampling for nonlinear non-Gaussian state space models. Journal of Business & Economic Statistics 32, Forthcoming.

Kullback, S., Leibler, R. A., 1951. On information and sufficiency. The Annals of Mathematical Statistics 22, 79–86.

Lando, D., Nielsen, M. S., 2010. Correlation in corporate defaults: contagion or conditional independence? Journal of Financial Intermediation 19, 355–372.

Lindsten, F., Schön, T. B., 2012. On the use of backward simulation in particle Markov chain Monte Carlo methods, Research Paper.

McCausland, W. J., 2012. The HESSIAN method: Highly efficient simulation smoothing, in a nutshell. Journal of Econometrics 168 (2), 189–206.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., Teller, E., 1953. Equations of state calculations by fast computing machines. Journal of Chemical Physics 21, 1087–1092.

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., 2002. Smoothed particle filters for likelihood evalution and maximization, Warwick Economic Research Papers.

Pitt, M. K., Shephard, N., 1999. Filtering via simulation: auxiliary particle filter. Journal of the American Statistical Association 94, 590–599.

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

Richard, J., Zhang, W., 2007. Efficient high-dimensional importance sampling. Journal of Econometrics 141, 1385–1411.

Roberts, G. O., Rosenthal, J. S., 2009. Examples of adaptive MCMC. Journal of Computational and Graphical Statistics 18, 349–367.

Scharth, M., 2012. Essays on Monte Carlo methods for state space models. Ph.D. thesis, VU University Amsterdam.

Shephard, N., 2005. Stochastic Volatility: Selected Readings. Oxford University Press.

Shephard, N., Pitt, M. K., 1997. Likelihood analysis of non-Gaussian measurement time series. Biometrika 84, 653–667.

Zeevi, A. J., Meir, R., 1997. Density estimation through convex combination of densities; approximation and estimation bounds. Neural Networks 10, 99–106.

A The Expectation and Maximization steps in MitISEM

We want to maximize the weighted log-density

$$\frac{1}{N}\sum_{j=1}^{N} w^{(j)} \log q_{\zeta'}(\theta^{(j)}|y), \tag{A1}$$

where the weight $w^{(j)}$ is the ratio of the target density kernel and the candidate density from which the d-dimensional vector draws $\theta^{(j)}$ have been simulated, with weight $w^{(j)}$ and where $q_{\zeta'}(\theta^{(j)}|y)$ is a mixture of H Student's t-densities. The target density kernel is either the marginal posterior density kernel of θ or the joint posterior density kernel of θ and the signal x; in the latter case the candidate density is the joint candidate density for $\theta^{(j)}$ and $x^{(j)}$ where $x^{(j)}$ has been simulated conditionally on $\theta^{(j)}$. We can write the mixture of Student's t-densities using a latent variable representation where $z^{(j)}$ is a latent H dimensional vector consisting of $H-1$ zeros and one element $z_h^j = 1$ that indicates that the draw $\theta^{(j)}$ belongs to the h-th Student's t-distribution. The mixing weight is $Pr[z_h^j = 1] = \eta_h$, and

$$\theta^{(j)} \sim \mathrm{N}(\mu, \Sigma), \qquad \mu = \sum_{i=1}^{H} z_h^j \mu_h, \quad \Sigma = \sum_{i=1}^{H} z_h^j \kappa_h^j \Sigma_h, \tag{A2}$$

where μ_h and Σ_h are the mode vector and scale matrix of the h-th Student's t-distribution, and where the random variable κ_h^j has an Inverse-Gamma distribution

$$\kappa_h^j \sim IG(\nu_h/2, \nu_h/2),$$

where ν_h is the degrees of freedom parameter of the h-th Student's t-distribution.

The Expectation-Maximization (EM) algorithm proceeds with iterations $L = 1, 2, \ldots$, which consist of an expectation and maximization step, until it has converged to a (local) optimum. In the expectation step of iteration L the conditional expectations of the expressions involving the latent variables z^j and κ^j that occur in the log-density, given the draws $\theta^{(j)}$ and $\zeta = \zeta^{(L-1)} = \{\mu_h, \Sigma_h, \nu_h, \eta_h;\ h = 1, \ldots, H\}$, the parameters from the previous EM iteration $(L-1)$, are the following:

$$\tilde{z}_h^j \equiv \mathrm{E}\left[z_h^j \middle| \theta^{(j)}, \zeta = \zeta^{(L-1)}\right] = \frac{t(\theta^{(j)}|\mu_h, \Sigma_h, \nu_h)\eta_h}{\sum_{i=1}^H t(\theta^{(j)}|\mu_i, \Sigma_i, \nu_i)\eta_i}, \tag{A3}$$

where $t(\cdot|\mu, \Sigma, \nu)$ is a Student's t-density with mode μ, scale matrix Σ and degree of freedom ν,

$$\widetilde{z/\kappa}_h^j \equiv \mathrm{E}\left[\frac{z_h^j}{\kappa_h^j}\middle| \theta^{(j)}, \zeta = \zeta^{(L-1)}\right] = \tilde{z}_h^j \frac{d + \nu_h}{\rho_h^j + \nu_h}, \tag{A4}$$

$$\begin{aligned} \xi_h^i &\equiv \mathrm{E}\left[\log \kappa_h^j | \theta^{(j)}, \zeta = \zeta^{(L-1)}\right] = \\ &= \left[\log\left(\frac{\rho_h^j + \nu_h}{2}\right) - \psi\left(\frac{d + \nu_h}{2}\right)\right]\tilde{z}_h^j + \left[\log\left(\frac{\nu_h}{2}\right) - \psi\left(\frac{\nu_h}{2}\right)\right](1 - \tilde{z}_h^j), \end{aligned} \tag{A5}$$

$$\delta_h^i \equiv \mathrm{E}\left[\frac{1}{\kappa_h^j}\middle| \theta^{(j)}, \zeta\right] = \widetilde{z/\kappa}_h^j + (1 - \tilde{z}_h^j), \tag{A6}$$

with $\rho_h^j \equiv (\theta^{(j)} - \mu_h)'\Sigma_h^{-1}(\theta^{(j)} - \mu_h)$, and $\psi(.)$ is the digamma function.

The maximization step of iteration L consists of the following updates

$$\mu_h^{(L)} = \left[\sum_{j=1}^N w^{(j)} \widetilde{z/\kappa}_h^j\right]^{-1} \left[\sum_{j=1}^N w^{(j)} \theta^{(j)} \widetilde{z/\kappa}_h^j\right], \tag{A7}$$

$$\Sigma_h^{(L)} = \frac{\sum_{j=1}^N w^{(j)}(\theta^{(j)} - \mu_h^{(L)})(\theta^{(j)} - \mu_h^{(L)})' \widetilde{z/\kappa}_h^j}{\sum_{j=1}^N w^{(j)} \tilde{z}_h^j}, \tag{A8}$$

$$\eta_h^{(L)} = \frac{\sum_{j=1}^N w^{(j)} \tilde{z}_h^j}{\sum_{j=1}^N w^{(j)}}. \tag{A9}$$

Finally, $\nu_h^{(L)}$ is obtained by solving the first-order condition

$$-\psi(\nu_h/2) + \log(\nu_h/2) + 1 - \frac{\sum_{j=1}^N w^{(j)} \xi_h^j}{\sum_{j=1}^N w^{(j)}} - \frac{\sum_{j=1}^N w^{(j)} \delta_h^j}{\sum_{j=1}^N w^{(j)}} = 0 \tag{A10}$$

for ν_h. For more details we refer to Hoogerheide et al. (2012).

B NAIS

We can express the likelihood of the state space model given by (1) and (2) as

$$L(y|\theta) = \int \frac{p(x, y|\theta)}{q(y|x, \theta)} q(y|x, \theta) \mathrm{d}x = q(y|\theta) \int \omega(x, y|\theta) q(y|x, \theta) \mathrm{d}x, \tag{A11}$$

where $x = (x_1, \ldots, x_T)'$, with $x_t = c_t + Z_t\alpha_t$ being the signal at time t for $t = 1, \ldots, T$, and where

$$\omega(x, y|\theta) \equiv p(y|x, \theta)/q(y|x, \theta). \tag{A12}$$

The Gaussian importance or proposal density can be written as

$$q(y_t|x_t, \theta) = \exp\left\{a_t + b_t'x_t - \frac{1}{2}x_t'C_tx_t\right\} \tag{A13}$$

where a_t, b_t and C_t depend on the observations y and the parameters in θ for $t = 1, \ldots, T$. The importance density at time t is effectively determined by b_t and C_t as the constant a_t is chosen such that the density integrates to one. We can represent the Gaussian importance density as the smoothed density in the linear Gaussian state space model with its observation equation given by

$$y_t^* = x_t + \varepsilon_t, \quad \varepsilon \sim \mathrm{N}(0, C_t^{-1}), \quad t = 1, \ldots, T, \tag{A14}$$

where $y_t^* = C_t^{-1}b_t$ for $t = 1, \ldots, T$ and the transition density given in equation (2).

To formulate an effective importance density we choose its parameters, as collected in $\chi = \{b_1, \ldots, b_T, C_1, \ldots, C_T\}$, by minimizing a conveniently chosen metric associated with the importance sample variation, that is

$$\min_{\chi_t} \int \lambda^2(x_t, y_t|\theta)\omega(x_t, y_t|\theta)q(x_t|y^*, \theta)\mathrm{d}x_t, \tag{A15}$$

for every t, where

$$\lambda(x_t, y_t|\theta) = \log p(y_t|x_t, \theta) - \log q(y_t^*|x_t, \theta) - \lambda_{0t}. \tag{A16}$$

We can rewrite the minimization as

$$\min_{\chi_t} \sum_{j=1}^{M} \lambda^2(\tilde{x}_{tj}, y_t|\theta)\omega_{tj}, \quad \omega_{tj} = q(\tilde{x}_{tj}|y^*, \theta)\omega(\tilde{x}_{tj}, y_t|\theta)h(z_j)e^{z_j^2}, \tag{A17}$$

with $\tilde{x}_{tj} = \hat{x}_t + V_t^{1/2}z_j$, for $j = 1, \ldots, M$, and

$$q(\tilde{x}_{tj}|y^*, \theta) = \exp\left\{-\frac{1}{2}z_j^2\right\}/\sqrt{2\pi}, \quad t = 1, \ldots, T, \tag{A18}$$

where $\hat{x}_t$ is the smoothed signal, V_t is smoothed signal variance and z_j are the abscissa designated by the Gauss-Hermite quadrature. The minimization is carried out iteratively. First, for a given χ we obtain $\hat{x}_t$ and V_t for $t = 1, \ldots, T$ from the linear Gaussian state space model in (A14). Second, we obtain the optimal $\chi_t = \{C_t, b_t\}$ for $t = 1, \ldots, T$ by a weighted least squares computation with "dependent" variable $\log p(y_t|\tilde{x}_{tj}, \theta)$ and "explanatory variables" $\tilde{x}_{tj}$ and $\tilde{x}_{tj}^2$. We iterate these steps until convergence of χ. For a more detailed discussion, we refer to Koopman et al. (2014).

C Adaptive random-walk Metropolis-Hastings

Roberts and Rosenthal (2009) propose an adaptive random walk Metropolis-Hastings algorithm, with a proposal of the following form

$$q_n(\theta;\theta_{n-1}) = \omega_{1n}\phi_d(\theta;\theta_{n-1},\kappa_1\Sigma_1) + \omega_{2n}\phi_d(\theta;\theta_{n-1},\kappa_2\Sigma_{2n}), \tag{A19}$$

where $\phi_d(\theta;\hat{\theta},\Sigma)$ is a d dimensional multivariate normal density with mean $\hat{\theta}$ and covariance matrix Σ. We set $\omega_{1n} = 1$ until $n > n_0$, and $\omega_{1n} = 0.05$ afterwards. The scalars $\kappa_1 = 0.1^2/d$ and $\kappa_2 = 2.38^2/d$ and $\Sigma_1 = I_d$ are constant throughout the procedure, while Σ_{2n} covariance matrix is estimated using the first $n-1$ iterates.

D Adaptive mixture of normals

Giordani and Kohn (2010) and Pitt et al. (2012) suggest an adaptive mixture of normals proposal, which has the form

$$q_n(\theta) = \sum_{k=1}^{4} \omega_{kn} q_{kn}(\theta) \quad \omega_{kn} \leq 0, \quad \text{for} \quad k = 1,\ldots,4 \quad \text{and} \quad \sum_{k=1}^{4} \omega_{kn} = 1, \tag{A20}$$

at iteration n. The adaptation has two stages. We start the first stage with setting $\omega_{1n} = 0.8$, $\omega_{2n} = 0.2$ and we use a Gaussian density for q_{1n} with mean equal to the simulated maximum likelihood estimates and variance equal to minus the inverse Hessian at the mean. Moreover we set q_{2n} as a heavy tailed version of q_{1n} by setting the covariance matrix 15 times the covariance matrix of q_{1n}. After $5d$ accepted draws (where d is equal to the dimension of θ) we set q_{3n} and q_{4n} and we change the component weights in (A20). q_{3n} is obtained as a mixture of normals using k-means clustering on the previous draws. q_{4n} is the fat tailed version of q_{3n}, it has the same means and mixture probabilities as q_{3n} but the covariance matrices are multiplied by 20. The new weights are the following $\omega_{1n} = 0.15$, $\omega_{2n} = 0.05$, $\omega_{3n} = 0.7$, $\omega_{4n} = 0.1$. In the rest of the first stage we update q_{3n} at predetermined updating times or after rejecting 10 candidate draws in a row. We always set q_{4n} to be the fat tailed version of q_{3n}. The first stage ends if the minimal acceptance rate (i.e., the conditional acceptance probability in the MH algorithm) in the last 1,000 draws is above 0.02. After the first stage we set $q_{1n} = q_{3n}$, i.e., the last version of the mixture of normals, and q_{2n} is again the fat tailed version of the new q_{1n}. In the second stage we only update at predetermined updating times.