Numerical Integration Filters for Maximum Likelihood Estimation of Asymmetric Stochastic Volatility Models

Hiroyuki Kawakatsu*
School of Management and Economics
25 University Square
Queen's University, Belfast
Belfast BT7 1NN
Northern Ireland
hkawakat@qub.ac.uk

January 28, 2005

Abstract

I consider (frequentist) maximum likelihood estimation of stochastic volatility models with leverage, where the correlation between the mean and volatility innovation can provide a more accurate estimate of the underlying volatility. I consider two filtering algorithms (quadrature and mixture Gaussian) for maximum likelihood estimation based on numerical integration. These algorithms extend straightforwardly to stochastic volatility models with non-Gaussian innovations. A small Monte Carlo simulation experiment shows that the mixture Gaussian filter performs remarkably well both in terms of accuracy and computation time. As an empirical application, I fit the asymmetric stochastic volatility model to the S&P 500 index daily returns with a Gaussian and skew-t innovation. The estimates from the two filtering algorithms are remarkably similar, suggesting the usefulness of the mixture Gaussian filter for practical use.

JEL classification: C13, C22.

Keywords: stochastic volatility, leverage effect, numerical integration, nonlinear filtering, mixture Gaussian.

*I thank Gauthier Lanot for insisting that I was using too many particles in the quadrature filter.

1 Introduction

Stochastic volatility (SV) models arise prominently in certain option pricing models and have been studied extensively in the literature (Ghysels, Harvey and Renault 1996, Broto and Ruiz 2004). However, due to the innovation in the unobserved volatility equation, estimation of SV models remains a difficult problem and appears to hamper its widespread use. In particular, until recently there was relatively little discussion of estimation of SV models with leverage effects, where the innovation in the volatility equation can be correlated with the innovation in the mean equation. In the presence of leverage, this correlation can provide a more accurate estimate of the underlying volatility from the information in the observed asset returns. Thus ignoring the leverage effect when in fact it is present can yield misleading estimates of the underlying volatility and result in mispricing of options. A number of researchers have recently proposed Bayesian Markov chain Monte Carlo (MCMC) methods for SV models with leverage (Jacquier, Polson and Rossi 2004, Omori, Chib, Shephard and Nakajima 2004, Yu 2004). The purpose of this paper is to consider (frequentist) maximum likelihood estimation (MLE) of SV models with leverage effects.

I consider estimation of the following univariate class of autoregressive stochastic (ARSV) models:[1]

$$y_t = \beta_0 \sigma_t u_t \tag{1a}$$

$$\log(\sigma_{t+1}^2) = \beta_1 \log(\sigma_t^2) + \beta_2 v_{t+1} \tag{1b}$$

$$\begin{bmatrix} u_t \\ v_{t+1} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right) \tag{1c}$$

y_t is the scalar observed series such as the returns from an financial asset, σ_t^2 is the unobserved volatility series, and u_t, v_{t+1} are innovations. The parameter vector to be estimated is $\theta^\top = (\beta_0, \beta_1, \beta_2, \rho)$ where we restrict $-1 < \beta_1 < 1$, $\beta_2 > 0$, and $-1 < \rho < 1$. β_0 is a scale parameter

[1] As noted recently by Yu (2004), once we allow $\rho \neq 0$, the interpretation of leverage differs depending on whether we define $\rho = cor(u_t, v_t)$ or $\rho = cor(u_t, v_{t+1})$. Following empirical evidence in Yu (2004) which favors the latter definition, I adopt this latter definition of ρ. The modification to handle the former defintion of ρ is straightforward.

that removes the constant term from the log-volatility equation (1b).

Broto and Ruiz (2004) have recently surveyed estimation methods for ARSV models without leverage effects ($\rho = 0$). They classify estimation methods into one of the following three: GMM, MLE, and indirect inference (EMM). The main drawback of GMM and EMM is that they do not directly provide estimates of the underlying volatility process. To estimate the volatility process from these methods, an additional step based on re-projection as proposed by Gallant and Tauchen (1998) needs to carried out. As the underlying volatility process is often of primary interest, I focus on MLE methods based on non-linear filtering that provide filtered estimates of the underlying volatility as a by-product of the estimation step.

The MLE approach I consider is based on filtering via numerical integration. Kitagawa (1987) proposed numerical integration based on piecewise linear approximation of the integrand for general non-linear non-Gaussian state-space models. Fridman and Harris (1998) used Gauss-Legendre quadrature and applied it to estimate stochastic volatility models without leverage ($\rho = 0$). To implement the numerical integration approach to non-linear filtering, I first cast the ARSV model (1) into a non-linear state-space form in section 2. While the state-space representation is not unique, computational considerations dictate us to choose a representation with a scalar state variable. In particular, I use an alternative parameterization to (1) which removes the dependency of the unconditional state distribution on the unknown parameter vector. I note that, unlike the QML approach of Harvey and Shephard (1996), I do *not* linearize the measurement equation via a log-square transformation. This is partly to avoid the need to introduce $\operatorname{sgn}(y_t)$ as in Harvey and Shephard (1996).

In section 3, I describe and provide pseudo-code for the non-linear filtering algorithm based on Gaussian quadratures. The Gauss-Legendre quadrature used by Fridman and Harris (1998) requires truncation of the state-space as we need to specify the integration bounds. To avoid the need to truncate the state-space, I consider the use of Gauss-Hermite quadrature which approximates an integral over the real line. As the Gauss-Hermite quadrature is tailored for use with a specific weighting function (exponential), it appears to be a natural choice for SV models based on Gaussian innovations. Gaussian quadratures can approximate an integral quite accurately with only a few nodes if the integrand has a known functional form. However, as the filtered state density in the integrand is unknown, we need a reasonably large number of nodes as they also serve as "particles" that span the unknown state-space.

To address the computational cost of the quadrature filters, in section 4, I consider approximating the unknown state distributions with a mixture of Gaussians. These so-called Gaussian filters, of which the extended Kalman filter is a special case, have a long history in the engineering literature (Alspach and Sorenson 1972, Ito and Xiong 2000). The primary focus in the engineering literature is the filtering problem with known parameters, while the focus in this paper is the estimation of unknown parameters. The formulation in Ito and Xiong (2000) assumes additive innovations and hence needs to be modified for the ARSV model with leverage (1). I show that, once we make the Gaussian assumption, the integral for the prediction step is available in closed form and dramatically reduces the computational cost. The integrals for the updating step, however, do not appear to have a closed form even with the Gaussian assumption. The extended Kalman filter approach would approximate the integrand by a first order Taylor expansion. However, because of the Gaussian assumption, we can now efficiently evaluate the integral by the Gauss-Hermite quadrature with only a few number of nodes.

Section 5 shows that the non-linear filtering algorithms discussed in sections 3–4 extend straightforwardly to SV models with non-Gaussian innovations. A number of researchers have considered non-Gaussian SV models to model the fat-tails observed in asset returns (Liesenfeld and Jung 2000, Cappuccio, Lubian and Raggi 2004, Jacquier et al. 2004, Omori et al. 2004). In section 5 I consider the use of the skew-t distribution proposed in Azzalini and Capitanio (2003) and provide expressions for the implied moments. The use of the skew-t distribution is motivated by the fact that it nests the commonly used Student-t distribution. Moreover, using results in Azzalini and Capitanio (2003), a simple diagnostic check of the distributional assumption based on the quantile-quantile plot is available as illustrated in the empirical application in section 7.

In section 6, I conduct a small Monte Carlo simulation experiment to assess the finite sample performance of the proposed algorithms. The simulation experiments reveal convergence problems with the Gauss-Hermite filter. On the other hand, the mixture Gaussian filters with only a small number of mixture components perform remarkably well both in terms of accuracy and computational cost. Section 7 fits the SV model to the daily returns from the S&P 500 index for two sample periods, one containing the market crash in October 1987. While the skew parameter is statistically significant for the skew-t model, it still fails to capture the extreme spike observed at the October 1987 crash. Finally, section 8 concludes with some suggestions for further research.

2 State-space formulation

A general state-space model can be formulated as

$$y_t = f_t(x_t, w_t) \qquad \text{(measurement equation)} \tag{2a}$$

$$x_{t+1} = g_t(x_t, v_{t+1}) \qquad \text{(state equation)} \tag{2b}$$

where w_t, v_{t+1} are zero mean independent innovations.

To cast the stochastic volatility model (1) into state-space form (2), we define the state variable $x_t = \log(\sigma_t^2)$ and apply a transformation so that the innovations in the state and measurement equations are independent. There are two ways to do this. First, if we define $w_t = \frac{1}{\sqrt{1-\rho^2}}(u_t - \rho v_{t+1})$, we have the state-space form

$$y_t = \beta_0 e^{\frac{x_t}{2}} \left(\tfrac{\rho}{\beta_2}(x_{t+1} - \beta_1 x_t) + \sqrt{1-\rho^2} w_t\right) \tag{3a}$$

$$x_{t+1} = \beta_1 x_t + \beta_2 v_{t+1} \tag{3b}$$

$$\begin{bmatrix} w_t \\ v_{t+1} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}\right) \tag{3c}$$

Second, if we define $w_{t+1} = \frac{1}{\sqrt{1-\rho^2}}(v_{t+1} - \rho u_t)$, we have the state-space form

$$y_t = \beta_0 e^{\frac{x_t}{2}} u_t \tag{4a}$$

$$x_{t+1} = \beta_1 x_t + \frac{\beta_2 \rho}{\beta_0} y_t e^{-\frac{x_t}{2}} + \beta_2 \sqrt{1-\rho^2} w_{t+1} \tag{4b}$$

$$\begin{bmatrix} u_t \\ w_{t+1} \end{bmatrix} \sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \tag{4c}$$

The first formulation (3) fits into the general state-space form (2) by defining the state vector as $\alpha_t^\top = (x_{t+1}, x_t)$ and a linear state equation

$$\alpha_t = \begin{pmatrix} \beta_1 & 0 \\ 1 & 0 \end{pmatrix} \alpha_{t-1} + \begin{pmatrix} \beta_2 \\ 0 \end{pmatrix} v_{t+1} \tag{5}$$

The difficulty with this formulation for the numerical integration approach is that the state vector has dimension two. Because the state innovation in (5) is uni-dimensional, the prediction step can be approximated by numerical integration over a univariate grid (Bølviken and Storvik 2001). However, the updating (or correction) step needs to be done over a two-dimensional grid to integrate out the state vector and can be quite computationally intensive.

For this reason, I use the state-space formulation (4) with a uni-dimensional state vector for the numerical integration approach. For this parameterization, however, the unconditional distribution of the state is $x_t \sim N(0, \frac{\beta_2^2}{1-\beta_1^2})$ and depends on the parameters β_1, β_2. The numerical integration method outlined below selects a finite discretized grid to cover the state-space. As the unconditional distribution of the state may provide guidance to the choice of this grid, its dependence on the unknown parameters is a nuisance. To remove the dependence of the unconditional state distribution on the parameters, I use the following parameterization instead.

$$
\begin{aligned}
y_t &= \sigma_t u_t && \text{(6a)}\\
\log(\sigma_t^2) &= \alpha_0 + \alpha_1 x_t && \text{(6b)}\\
x_{t+1} &= \phi x_t + \sqrt{1-\phi^2} v_{t+1}, \quad x_0 \sim N(0,1) && \text{(6c)}\\
\begin{bmatrix} u_t \\ v_{t+1} \end{bmatrix} &\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & \rho \\ \rho & 1 \end{bmatrix} \right) && \text{(6d)}
\end{aligned}
$$

with state-space representation

$$
\begin{aligned}
y_t &= e^{\frac{1}{2}(\alpha_0+\alpha_1 x_t)} u_t && \text{(7a)}\\
x_{t+1} &= \phi x_t + \rho\sqrt{1-\phi^2} e^{-\frac{1}{2}(\alpha_0+\alpha_1 x_t)} y_t + \sqrt{(1-\phi^2)(1-\rho^2)} w_{t+1} && \text{(7b)}\\
\begin{bmatrix} u_t \\ w_{t+1} \end{bmatrix} &\sim N\left(\begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \right) && \text{(7c)}
\end{aligned}
$$

Note that under this parameterization, the unconditional state distribution is standard normal and does not depend on the parameter vector $\theta^\top = (\alpha_0, \alpha_1, \phi, \rho)$ where we restrict $|\phi| < 1$ and $|\rho| < 1$.[2]

Formulation (7) does not quite fit into the standard state-space form (2) because of the presence of the measurement y_t in the state equation. However, a simple modification to the standard filtering recursions can accomodate formulation (7). Denote $p(\cdot)$ for generic density functions and $y_{s:t} = \{y_s, y_{s+1}, \dots, y_t\}$. Then the standard filtering recursions for (2) can be written as

$$
\begin{aligned}
p(x_t|y_{1:t-1}) &= \int p(x_t|x_{t-1}) p(x_{t-1}|y_{1:t-1}) dx_{t-1} && \text{(prediction step)}\\
p(x_t|y_{1:t}) &= \tfrac{1}{c_t} p(y_t|x_t) p(x_t|y_{1:t-1}) && \text{(updating step)}\\
c_t &= \int p(y_t|x_t) p(x_t|y_{1:t-1}) dx_t
\end{aligned}
$$

[2] The mapping between the parameters in (1) and those in (6) are given by $\beta_0 = e^{\frac{\alpha_0}{2}}$, $\beta_1 = \phi$, $\beta_2 = \alpha_1\sqrt{1-\phi^2}$ and $\alpha_0 = 2\log\beta_0$, $\alpha_1 = \frac{\beta_2}{\sqrt{1-\beta_1^2}}$, $\phi = \beta_1$.

The prediction step for the standard formulation (2) is based on the Markov property $p(x_t|x_{t-1}, y_{1:t-1}) = p(x_t|x_{t-1})$. However, for formulation (7), we have $p(x_t|x_{t-1}, y_{1:t-1}) = p(x_t|x_{t-1}, y_{t-1})$ instead. Thus the filtering recursions for (7) need to be modified as follows

$$p(x_t|y_{1:t-1}) = \int p(x_t|x_{t-1}, y_{t-1})p(x_{t-1}|y_{1:t-1})dx_{t-1} \tag{8a}$$

$$p(x_t|y_{1:t}) = \tfrac{1}{c_t}p(y_t|x_t)p(x_t|y_{1:t-1}) \tag{8b}$$

$$c_t = \int p(y_t|x_t)p(x_t|y_{1:t-1})dx_t \tag{8c}$$

Recursion (8) forms the basis of the numerical integration approach to likelihood evaluation for nonlinear state space models advocated in this paper. From the recursion (8), the log-likelihood value for a given parameter vector θ can be obtained as

$$\ell(\theta) = \sum_{t=1}^{T} \log(c_t) \tag{9}$$

The difficulty in using the recursions (8) to evaluate the likelihood is that the prediction $p(x_t|y_{1:t-1})$ and filtered $p(x_t|y_{1:t})$ densities are typically not known. Below I reconsider two approaches to approximate these unknown densities. The first approach is to approximate these densities nonparametrically by a set of m (deterministic) particles, or points, in the state-space with associated weights or "probabilities". The key to this approach is to tradeoff accuracy (large m) with computational efficiency (small m). The second approach is to approximate the unknown densities with a flexible parametric function. As discussed below, I reconsider the use of Gaussian mixtures as an approximating density. The key to this approach is to how to handle the additional parameters in this approximating function.

3 Quadrature filters

The first approach I consider is to evaluate the integrals in (8) by numerical integration. The feasibility of numerical integration for low dimension state vectors was shown by Kitagawa (1987). Bølviken and Storvik (2001) call this as filtering based on deterministic particles. Kitagawa (1987) evaluates the integral with Newton-Cotes quadrature where the integrand is approximated by a piecewise linear function on an arbitrarily chosen equi-spaced grid on a closed interval. Fridman and Harris (1998) and Bølviken and Storvik (2001) choose the grid based on the Gaussian quadrature. Fridman and Harris (1998) analyze stochastic volatility models only for the case $\rho = 0$ with possibly non-Gaussian distributions.[3]

The Gaussian quadrature integration rule is based on the approximation (Judd 1998, p.258)

$$\int_a^b f(x)\omega(x)dx \approx \sum_{i=1}^m w_i f(x_i) \tag{10}$$

where $\omega(x)$ is a non-negative weighting function and x_i, w_i are the quadrature nodes and weights, respectively. Note that the nodes x_i and weights w_i do not depend on the function $f(\cdot)$. Both Fridman and Harris (1998) and Bølviken and Storvik (2001) use Gauss-Legendre quadrature for weighting function $\omega(x) = 1$. The algorithm presented below uses two arrays p_1, p_0, each of size m, which store the deterministic "particles" that approximate the prediction $p(x_t|y_{1:t-1})$ and filtered $p(x_t|y_{1:t})$ densities, respectively. The idea is to use the nodes x_i to represent the support of the state distribution and to propagate the particles through the recursion (8).

$$\begin{aligned} p_1(x_i) &= \sum_{j=1}^m p(x_i|x_j, y_{t-1})\widetilde{p}_0(x_j) && i = 1, \dots, m \\ c_t &= \sum_{j=1}^m w_j p(y_t|x_j)\frac{p_1(x_j)}{\omega(x_j)} \\ \widetilde{p}_0(x_i) &= \frac{1}{c_t} p(y_t|x_i) w_i \frac{p_1(x_i)}{\omega(x_i)} && i = 1, \dots, m \end{aligned}$$

where $\widetilde{p}_0(x_j) = w_j p_0(x_j)/\omega(x_j)$. Note that x_j with weights $\widetilde{p}_0(x_j)$ for $j = 1, \dots, m$ is the particle approximation of the filtered distribution $p(x_t|y_{1:t})$.

[3] Fridman and Harris (1998) only mention the possibility that their method can be modified to accomodate the leverage case $\rho \neq 0$.

Algorithm 1 : Gaussian quadrature numerical integration

```
input: $y$ ($T \times 1$), $\theta$ ($n_p \times 1$), $m$ (scalar), $a$ (scalar), $b$ (scalar)
output: $\ell$ (scalar) (9)

initialization:
    $x$, $w \leftarrow$ Gaussian quadrature nodes and weights ($m \times 1$) based on $a$ and $b$
    $p_0[i] \leftarrow w_i p_0(x_i)/\omega(x_i)$ for $i = 1, \ldots, m$ (pre-sample state distribution)
    $\ell \leftarrow 0$

for $t = 1$ to $T$ do
    $p_1[i] \leftarrow \sum_{j=1}^{m} p(x_i|x_j, y_{t-1})\, p_0[j]$, $i = 1, \ldots, m$ (8a)
    $p_0[i] \leftarrow p(y_t|x_i)\, w_i\, p_1[i]/\omega(x_i)$, $i = 1, \ldots, m$
    $c_t \leftarrow \sum_{j=1}^{m} p_0[j]$ (8c)
    $p_0[i] \leftarrow p_0[i]/c_t$, $i = 1, \ldots, m$ (8b)
    $\ell \leftarrow \ell + \log(c_t)$ (9)
end for
```

The algorithm reveals that the computational bottleneck is the prediction step (8a) which is of order $O(m^2)$ for each observation t. That is, for each of m points x_t, we need to evaluate the integral using m points of x_{t-1}. This is quite frustrating as Gaussian quadratures are quite accurate even with a small number of nodes m. The reason for the $O(m^2)$ cost is that the integrand does not have a functional form but is represented by a set of m points with associated weights. That is, the particles in the quadrature filter are serving two purposes, one to cover the support of the state-space and two to evaluate the integral.

To deal with this computational bottleneck, we somehow need to separate these two roles of the particles. One way to do this is to assume a flexible parameteric form for the integrand and is discussed below in section 4. Alternatively, I note that the main cost in the prediction step arises from having to the evaluate the conditional density $p(x_i|x_j, y_{t-1})$ (which may be non-Gaussian) at m^2 points (x_i, x_j). One way to reduce this computational cost is to pre-calculate the conditional density $p(x_i|x_j, y_{t-1})$. For $\rho = 0$, this would only cost an additional $m \times m$ array for $p(x_i|x_j)$. For $\rho \neq 0$, however, pre-computing and holding the $m^2 T$ array for $p(x_i|x_j, y_{t-1})$ would become costly as the sample size T gets large. An ad hoc method to reduce the cost of evaluating the conditional density is to check the weights p_0 and to evaluate $p(x_i|x_j, y_{t-1})$ only if p_0 is not close to zero. This approximation would work well if $p(x_{t-1}|y_{1:t-1})$ were close to zero for most points in the grid for x_{t-1} *and* $p(x_t|x_{t-1}, y_{t-1})$ were close to zero at x_{t-1} points where $p(x_{t-1}|y_{1:t-1})$ is close to zero.

There remains the issue of the choice of the Gaussian quadrature. Both Fridman and Harris (1998) and Bølviken and Storvik (2001) use Gauss-Legendre quadrature with weighting function $\omega(x) = 1$. The problem with Gauss-Legendre quadrature is that one needs to select the integration bounds $[a, b]$ in (10). For non-stationary models, one might need to adjust these bounds and hence the quadrature nodes and weights by observation. For stationary models with a fixed set of nodes and weights, it is natural to choose the integration bounds so that they cover most of the support of the unconditional state distribution. For the stochastic volatility model with parameterization (6), the unconditional state distribution is $p(x_0) \sim N(0, 1)$. Thus one could set the bounds centered about the unconditional mean of zero with $b = -a$ where a has the interpretation of the standard deviations covered by the interval.

In this paper, I consider the use of Gauss-Hermite quadrature which takes the form (Judd 1998, p.261)

$$\int_{-\infty}^{\infty} f(x)e^{-x^2/2}dx \approx \sum_{i=1}^{m} w_i f(x_i) \tag{11}$$

As can be seen, the Gauss-Hermite quadratute approximates an integral over the entire real line with weighting function $\omega(x) = e^{-x^2/2}$. This quadrature rule is ideal for the Gaussian density and certain members of the exponential family. The problem, of course, is that there are many distributions that do not have a weighting of the form $\omega(x) = e^{-x^2/2}$ such as the t-distribution. For these densities, one needs to use the approximation of the form $\sum_{i=1}^{m} w_i f(x_i) e^{x_i^2/2}$ instead. The effect of the choice of the quadrature rule on the performance of the numerical integration filter is examined by a Monte Carlo experiment in section 6.

4 Mixture Gaussian filters

The quadrature filters in section 3 will consistently evaluate the likelihood (9) as the number of nodes $m \to \infty$. However, because the prediction step is of order $O(m^2)$ the computational cost increases rapidly with m. In this section, I consider approximation methods based on the parameterization of the unknown prediction $p(x_t|y_{1:t-1})$ and filtered $p(x_t|y_{1:t})$ densities.

The key to the parameteric approach is to choose a flexible functional form. The tradeoff is between flexibility and ease of estimation. One should note that the unknown densities must be estimated for each observation t. A natural candidate for the parametric form is a low order polynomial. However, "the Gram-Charlier and Edgeworth expansions have been proposed and investigated. . . . , it has the distinct disadvantage that, when truncated, the resulting series approximation is not itself a valid density function." (Alspach and Sorenson 1972).

In this section, I follow Alspach and Sorenson (1972) and consider approximations based on mixtures of Gaussians. The mixture Gaussian filters considered by Alspach and Sorenson (1972) and Kitagawa and Gersch (1996) are based on the linearization of the nonlinear state-space model. Here I consider mixture Gaussian filters based on numerical integration without the linear approximation. This filter was recently analyzed by Ito and Xiong (2000). The analysis in Ito and Xiong (2000) assumes additive innovations in the measurement and state equations and hence does not directly apply to the stochastic volatility model (7). Therefore, instead of the generalized Kalman filter algorithm of Ito and Xiong (2000), I construct an approximate filter based directly on the recursions (8).

In addition to solving the truncation problem of polynomial approximations, the use of mixture Gaussian approximation for the proposed filter has the distinct advantage that numerical integration can be done efficiently with the use of Gauss-Hermite quadrature. Unlike the quadrature filter in section 3, the nodes of the Gauss-Hermite quadrature need not be large as we now have a functional approximation to the unknown densities.

4.1 Component Gaussian filter

The idea of the Gaussian mixture filter is to use the recursions (8) to update each Gaussian component in parallel. In this subsection I discuss filtering recursions for each Gaussian component under the assumption that the prediction $p(x_t|y_{1:t-1})$ and filtered $p(x_t|y_{1:t})$ state distributions can both be approximated by a Gaussian distribution. As the Gaussian distribution is fully characterized by its first two moments, once we make the Gaussian assumption, all we need to do is to update the first two moments.

The main benefit of making the Gaussian assumption is that the first two moments from the prediction step (8a) may be evaluated analytically. This is the case for the stochastic volatility model (7) with Gaussian innovations. If we assume that $p(x_{t-1}|y_{1:t-1}) \sim N(\mu_{t-1}, s_{t-1}^2)$, then the prediction density has first two moments

$$\begin{aligned}
E[x_t|y_{1:t-1}] &= \int xp(x|y_{1:t-1})dx \\
&= \int \left(\int xp(x|x_{t-1}, y_{t-1})dx \right) p(x_{t-1}|y_{1:t-1})dx_{t-1} \\
&= \int E[x_t|x_{t-1}, y_{t-1}]p(x_{t-1}|y_{1:t-1})dx_{t-1} \\
&= \int (\phi x_{t-1} + \rho\sqrt{1-\phi^2}y_{t-1}e^{-\frac{1}{2}(\alpha_0+\alpha_1 x_{t-1})})p(x_{t-1}|y_{1:t-1})dx_{t-1} \\
&= \phi\mu_{t-1} + \rho\sqrt{1-\phi^2}y_{t-1}e^{-\frac{\alpha_0}{2}-\frac{\alpha_1}{2}\mu_{t-1}+\frac{\alpha_1^2}{8}s_{t-1}^2} \\
E[x_t^2|y_{1:t-1}] &= \int x^2p(x|y_{1:t-1})dx \\
&= \int \left(\int x^2p(x|x_{t-1}, y_{t-1})dx \right) p(x_{t-1}|y_{1:t-1})dx_{t-1} \\
&= \int E[x_t^2|x_{t-1}, y_{t-1}]p(x_{t-1}|y_{1:t-1})dx_{t-1} \\
&= \int (\phi^2x_{t-1}^2 + \rho^2(1-\phi^2)y_{t-1}^2e^{-\alpha_0-\alpha_1x_{t-1}} + (1-\phi^2)(1-\rho^2) \\
&\quad + 2\phi\rho\sqrt{1-\phi^2}y_{t-1}x_{t-1}e^{-\frac{1}{2}(\alpha_0+\alpha_1x_{t-1})})p(x_{t-1}|y_{1:t-1})dx_{t-1} \\
&= \phi^2(\mu_{t-1}^2 + s_{t-1}^2) + \rho^2(1-\phi^2)y_{t-1}^2e^{-\alpha_0-\alpha_1\mu_{t-1}+\frac{\alpha_1^2}{2}s_{t-1}^2} + (1-\phi^2)(1-\rho^2) \\
&\quad + 2\phi\rho\sqrt{1-\phi^2}y_{t-1}(\mu_{t-1} - \frac{\alpha_1}{2}s_{t-1}^2)e^{-\frac{\alpha_0}{2}-\frac{\alpha_1}{2}\mu_{t-1}+\frac{\alpha_1^2}{8}s_{t-1}^2}
\end{aligned}$$

Thus the Gaussian approximation to the prediction density is $N(\mu_{t|t-1}, s_{t|t-1}^2)$ where

$$\mu_{t|t-1} = \phi\mu_{t-1} + \rho\sqrt{1-\phi^2}y_{t-1}e^{-\frac{\alpha_0}{2}-\frac{\alpha_1}{2}\mu_{t-1}+\frac{\alpha_1^2}{8}s_{t-1}^2} \tag{12a}$$

$$\begin{aligned}
s_{t|t-1}^2 &= \phi^2(\mu_{t-1}^2 + s_{t-1}^2) + \rho^2(1-\phi^2)y_{t-1}^2e^{-\alpha_0-\alpha_1\mu_{t-1}+\frac{\alpha_1^2}{2}s_{t-1}^2} + (1-\phi^2)(1-\rho^2) \\
&\quad + 2\phi\rho\sqrt{1-\phi^2}y_{t-1}(\mu_{t-1} - \frac{\alpha_1}{2}s_{t-1}^2)e^{-\frac{\alpha_0}{2}-\frac{\alpha_1}{2}\mu_{t-1}+\frac{\alpha_1^2}{8}s_{t-1}^2} - \mu_{t|t-1}^2
\end{aligned} \tag{12b}$$

Now consider the integral (8c) in the updating step. Under the Gaussian assumptions,

$p(y_t|x_t) \sim N(0, e^{\alpha_0+\alpha_1 x_t})$ and we need to evaluate

$$c_t = \int \frac{1}{2\pi s_{t|t-1}} \exp\bigl(-\frac{1}{2}(\alpha_0 + \alpha_1 x) - \frac{y_t^2}{2} e^{-\alpha_0-\alpha_1 x} - \frac{(x - \mu_{t|t-1})^2}{2s_{t|t-1}^2}\bigr) dx$$

This integral does not appear to have a closed form solution.[4] Therefore, I use the Gauss-Hermite quadrature approximation to compute the first two moments from the updating step (8b)–(8c). Let z_i, w_i be m nodes and weights for the Gauss-Hermite quadrature such that

$$\int_{-\infty}^{\infty} f(z) e^{-z^2/2} dz \approx \sum_{i=1}^{m} f(z_i) w_i$$

The first two moments from the updating step are then approximated as

$$c_t = \sum_{i=1}^{m} f_0(z_i) w_i$$

$$\mu_t = \tfrac{1}{c_t} \sum_{i=1}^{m} f_1(z_i) w_i \tag{13a}$$

$$s_t^2 = \tfrac{1}{c_t} \sum_{i=1}^{m} f_2(z_i) w_i - \mu_t^2 \tag{13b}$$

where

$$f_j(z_i) = \frac{x_i^j}{2\pi} \exp\bigl(-\frac{1}{2}(\alpha_0 + \alpha_1 x_i) - \frac{y_t^2}{2} e^{-\alpha_0-\alpha_1 x_i}\bigr), \quad x_i = \mu_{t|t-1} + s_{t|t-1} z_i$$

I note that compared to the quadrature filter considered in section 3, the number of nodes m in the approximation for c_t can be quite small. This is because the nodes no longer represent particles to span the support of the state-space. Once we have a functional form, the integral can be evaluated accurately with only a small number of nodes because an m-point Gaussian quadrature integrates a polynomial of order $2m - 1$ *exactly* (Judd 1998, p.258).

[4] Mathematica 5.1 does not return a closed form solution.

4.2 Mixture Gaussian filter

The Gaussian filter given by the recursions (12a)–(12b) and (13a)–(13b) is based on the assumption that the prediction $p(x_t|y_{1:t-1})$ and filtered $p(x_t|y_{1:t})$ state distributions are both Gaussian. This assumption is questionable, especially for models with non-Gaussian innovations. Therefore, I consider approximating the unknown prediction and filtered distributions with a mixture of Gaussian distributions. Assume then that

$$p(x_t|y_{1:t-1}) \approx \sum_{i=1}^{n} a_{t|t-1}^{(i)} p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}), \quad a_{t|t-1}^{(i)} \geq 0, \quad \sum_{i=1}^{m} a_{t|t-1}^{(i)} = 1 \tag{14a}$$

$$p(x_t|y_{1:t}) \approx \sum_{i=1}^{n} a_{t}^{(i)} p_N(x_t|\mu_{t}^{(i)}, s_{t}^{2(i)}), \quad a_{t}^{(i)} \geq 0, \quad \sum_{i=1}^{m} a_{t}^{(i)} = 1 \tag{14b}$$

where $p_N(x|\mu, s^2)$ denotes the Gaussian density function with mean μ and variance s^2. $a_{t|t-1}^{(i)}$ and $a_t^{(i)}$ are the non-negative mixture weights.

It is intuitively clear that a mixture of Gaussians converges to any density function as the number of components $n \to \infty$ and $s^{2(i)} \to 0$ (Alspach and Sorenson 1972). As $n \to \infty$ and $s^{2(i)} \to 0$ we are back to the particle filter discussed in section 3. Note that I am assuming that the number of mixture components n is the same for both the prediction and filtered distributions and that n does not change over time.

The problem then is to find the mixture approximations $(a_{t|t-1}^{(i)}, \mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)})$ and $(a_t^{(i)}, \mu_t^{(i)}, s_t^{2(i)})$ for each t. One approach is to use method of moments or least squares. This may be feasible if we are only interested in filtering the state variable given the model parameters. However, for estimating the model parameters by numerically maximizing the likelihood evaluated by the filter, observation-by-observation approximation based on the method of moments or least squares is too costly. Here, I follow Ito and Xiong (2000) and update each mixture component $(\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)})$ and $(\mu_t^{(i)}, s_t^{2(i)})$ separately by the recursions (12a)–(12b) and (13a)–(13b). To determine the recursions for the mixture weights $a_{t|t-1}^{(i)}$, $a_t^{(i)}$, note that the prediction step

(8a) becomes

$$
\begin{aligned}
p(x_t|y_{1:t-1}) &= \int p(x_t|x_{t-1}, y_{t-1}) \sum_{i=1}^{n} a_{t-1}^{(i)} p_N(x_{t-1}|\mu_{t-1}^{(i)}, s_{t-1}^{2(i)}) dx_{t-1} \\
&= \sum_{i=1}^{n} a_{t-1}^{(i)} \int p(x_t|x_{t-1}, y_{t-1}) p_N(x_{t-1}|\mu_{t-1}^{(i)}, s_{t-1}^{2(i)}) dx_{t-1} \qquad (15)
\end{aligned}
$$

By matching each component in (14a) and (15), we have

$$
a_{t|t-1}^{(i)} p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) = a_{t-1}^{(i)} \int p(x_t|x_{t-1}, y_{t-1}) p_N(x_{t-1}|\mu_{t-1}^{(i)}, s_{t-1}^{2(i)}) dx_{t-1}
$$

As this condition should hold for any x_t, we "average out" x_t by integrating it out from both sides to get

$$
\begin{aligned}
a_{t|t-1}^{(i)} \int p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) dx_t &= a_{t-1}^{(i)} \int\int p(x_t|x_{t-1}, y_{t-1}) p_N(x_{t-1}|\mu_{t-1}^{(i)}, s_{t-1}^{2(i)}) dx_{t-1} dx_t \\
a_{t|t-1}^{(i)} &= a_{t-1}^{(i)} \int p_N(x_{t-1}|\mu_{t-1}^{(i)}, s_{t-1}^{2(i)}) \left(\int p(x_t|x_{t-1}, y_{t-1}) dx_t \right) dx_{t-1} \\
&= a_{t-1}^{(i)} \qquad (16)
\end{aligned}
$$

The updating step (8b)–(8c) becomes

$$
\begin{aligned}
c_t &= \int p(y_t|x_t) \sum_{i=1}^{n} a_{t|t-1}^{(i)} p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) dx_t \\
&= \sum_{i=1}^{n} a_{t|t-1}^{(i)} \int p(y_t|x_t) p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) dx_t \\
&= \sum_{i=1}^{n} a_{t|t-1}^{(i)} c_t^{(i)} \\
p(x_t|y_{1:t}) &= \frac{1}{c_t} p(y_t|x_t) \sum_{i=1}^{n} a_{t|t-1}^{(i)} p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) \\
&= \frac{1}{c_t} \sum_{i=1}^{n} a_{t|t-1}^{(i)} p(y_t|x_t) p_N(x_t|\mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) \qquad (17)
\end{aligned}
$$

where $c_t^{(i)}$ is the contribution to the likelihood from component i in the update step (13). By matching each component in (14b) and (17) and using (16), we have

$$a_t^{(i)} p_N(x_t | \mu_t^{(i)}, s_t^{2(i)}) = \frac{1}{c_t} a_{t-1}^{(i)} p(y_t|x_t) p_N(x_t | \mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)})$$

As this condition should hold for any x_t, we "average out" x_t by integrating it out from both sides to get

$$\begin{aligned} a_t^{(i)} \int p_N(x_t | \mu_t^{(i)}, s_t^{2(i)}) dx_t &= \frac{1}{c_t} a_{t-1}^{(i)} \int p(y_t|x_t) p_N(x_t | \mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) dx_t \\ a_t^{(i)} &= \frac{a_{t-1}^{(i)} c_t^{(i)}}{c_t} = \frac{a_{t-1}^{(i)} c_t^{(i)}}{\sum_{i=1}^n a_{t-1}^{(i)} c_t^{(i)}} \end{aligned}$$

The last expression shows that the mixture weight for the i-th component is updated according to its contribution to the likelihood, a fairly intuitive result.[5]

5 Non-Gaussian stochastic volatility

A number of studies have examined stochastic volatility models with non-Gaussian innovations. This line of research is motivated by the observation that Gaussian stochastic volatility models may fail to capture fat-tails, and possibly skewness, in asset returns y_t. Therefore, most studies model the mean equation innovation u_t to be non-Gaussian. For models without leverage ($\rho = 0$), distributions such as the Student-t (Fridman and Harris 1998, Liesenfeld and Jung 2000), generalized error (GED) (Liesenfeld and Jung 2000), and skew-GED (Cappuccio et al. 2004) have been considered. For models with leverage ($\rho \neq 0$), Jacquier et al. (2004) consider Student-t and Omori et al. (2004) consider a scale mixture distribution.

One could also consider the volatility innovation v_{t+1} to be non-Gaussian. However, care must be taken to ensure that moments of y_t exist. To see this point, consider the non-Gaussian

[5] Alternatively, rather than integrating out x_t, we can "collocate" each component at a particular value of x_t, say $x_t = \mu_t^{(i)}$ (Ito and Xiong 2000). We then have

$$\begin{aligned} a_t^{(i)} p_N(x_t = \mu_t^{(i)} | \mu_t^{(i)}, s_t^{2(i)}) &= \frac{1}{c_t} a_{t-1}^{(i)} p(y_t | x_t = \mu_t^{(i)}) p_N(x_t = \mu_t^{(i)} | \mu_{t|t-1}^{(i)}, s_{t|t-1}^{2(i)}) \\ a_t^{(i)} &= \frac{a_{t-1}^{(i)}}{c_t} \frac{s_t^{(i)}}{s_{t|t-1}^{(i)}} \exp\left(-\frac{(\mu_t^{(i)} - \mu_{t|t-1}^{(i)})^2}{2 s_{t|t-1}^{2(i)}} \right) p(y_t | x_t = \mu_t^{(i)}) \end{aligned}$$

stochastic volatility model (6) where the Gaussian assumption for the innovations u_t, v_{t+1} is dropped. The r-th moment of y_t is given by

$$E[y_t^r] = E[\sigma_t^r]E[u_t^r] = E[e^{\frac{r}{2}(\alpha_0+\alpha_1 x_t)}]E[u_t^r]$$

To obtain this moment, we need to evaluate $E[e^{cx_t}]$ for some constant c. From (6) we can write

$$\begin{aligned} x_t &= \sqrt{1-\phi^2}\sum_{j=0}^{\infty}\phi^j v_{t-j} \\ E[e^{cx_t}] &= E[e^{c\sqrt{1-\phi^2}\sum_{j=0}^{\infty}\phi^j v_{t-j}}] = E[\prod_{j=0}^{\infty} e^{c\sqrt{1-\phi^2}\phi^j v_{t-j}}] = \prod_{j=0}^{\infty} E[e^{c\sqrt{1-\phi^2}\phi^j v_{t-j}}] \\ &= \prod_{j=0}^{\infty} M(c\sqrt{1-\phi^2}\phi^j) \end{aligned}$$

where $M(s) = E[e^{sv_t}]$ is the moment generating function of v_t. This expression shows that moments of y_t exist only if *all* moments of v_t exist. For example, as noted by Jacquier et al. (2004, p.190), the moments of y_t do not exist if v_t has a Student t-distribution.[6]

The filters discussed in sections 3–4 can be easily adapted for non-Gaussian models. Below I remark on the issues that might arise for each filtering method.

5.1 Quadrature filters for non-Gaussian stochastic volatility

The filter based on Gaussian quadratures discussed in section 3 can be adapted for non-Gaussian models as long as the conditional densities $p(x_t|x_{t-1}, y_{t-1})$ and $p(y_t|x_t)$ can be evaluated. As these densities must be evaluated at each of the m points in the state-space grid for all t, the computational cost will rise for densities that are costly to evaluate. Moreover, the Gauss-Hermite quadrature suggested in section 3 may perform poorly for non-Gaussian distributions that do not belong to the exponential family. For these cases, the Gauss-Legendre quadrature with finite bounds may be considered instead.

[6] Fridman and Harris (1998) consider a signed normal power transform $v_{t+1} = \text{sgn}(z)|z|^{\gamma}$ where $z \sim N(0,1)$. It is not clear under what conditions on γ the moments of y_t exist for this distribution.

5.2 Mixture Gaussian filters for non-Gaussian stochastic volatility

As shown in section 4, the main benefit from making the Gaussian assumption for the prediction and filtered densities is that the prediction step can be done analytically. Note that this is still the case with non-Gaussian innovations; all that is required is that the innovation is standardized so that $E[v_{t+1}] = 0$ and $E[v_{t+1}^2] = 1$. For non-Gaussian innovations, the number of mixture components n may need to increased. The computational cost for each prediction step is of order $O(mn)$ for m quadrature points and n mixture components and may still be cheaper than the $O(m^2)$ cost of the quadrature filter.

I note that an alternative approach is to approximate the non-Gaussian innovation distributions by a mixture of Gaussians (Kotecha and Djurić 2003, Omori et al. 2004). However, there is little advantage to this approach for the mixture filter as the updating step cannot be done analytically even for Gaussian innovations. The mixture approximation for the innovation distribution introduces an additional source of approximation error. Furthermore, if the non-Gaussian innovation distribution depends on an unknown parameter (such as the degrees of freedom parameter for the Student-t), then the approximating mixture for the innovation distribution may have to be updated as this parameter changes during estimation. For these reasons, I will not consider this approach in this paper.

5.3 Skew-t stochastic volatility

For a non-Gaussian stochastic volatility model, I consider the case where the mean equation innovation u_t has the skew-t distribution of Azzalini and Capitanio (2003). The skew-t distribution nests the commonly used Student-t distribution with heavy tails. The skew-t distribution is a four parameter distribution with density function[7]

$$f(u) = 2f_\nu(u)F_{\nu+1}\left(\frac{\alpha}{\sigma}(u-\xi)\sqrt{\frac{\nu+1}{\nu+(u-\xi)^2/\sigma^2}}\right)$$

[7] As noted in Azzalini and Capitanio (2003), an alternative (and simpler) density function for the skew-t is $f(u) = 2f_\nu(u)F_\nu(\alpha u)$. However, I follow the recommended formulation of Azzalini and Capitanio (2003) who argue that there is no reason to restrict the argument of $F_\nu(\cdot)$ to be linear in u.

where

$$\begin{aligned}
f_\nu(u) &= \frac{1}{\sigma\sqrt{\pi\nu}} \frac{\Gamma(\frac{\nu+1}{2})}{\Gamma(\frac{\nu}{2})} \left(1 + \frac{(u-\xi)^2}{\nu\sigma^2}\right)^{-\frac{\nu+1}{2}} \\
F_\nu(u) &= \int_{-\infty}^{u} f_\nu(s) ds
\end{aligned}$$

$f_\nu(\cdot)$ is the Student-t density function and $F_\nu(\cdot)$ its distribution function. The parameters ξ, σ^2, α, ν are the location, scale, skew, and tail-thickness parameter respectively. The Student-t distribution is obtained by setting $\alpha = 0$.

As with the Student-t distribution, the r-th moment of skew-t exists provided $\nu > r$ (Azzalini and Capitanio 2003). The first two moments are

$$\begin{aligned}
E[u] &= \xi + \sigma c & c &= \frac{\alpha}{\sqrt{1+\alpha^2}} \sqrt{\frac{\nu}{\pi}} \frac{\Gamma(\frac{\nu-1}{2})}{\Gamma(\frac{\nu}{2})} \\
E[u^2] &= \xi^2 + 2\xi\sigma c + \sigma^2 \frac{\nu}{\nu-2}
\end{aligned}$$

To standardize the innovation process so that $E[u] = 0$ and $E[u^2] = 1$, I set the location and scale parameters to

$$\sigma^2 = \frac{1}{\nu/(\nu-2) - c^2} \tag{18a}$$

$$\xi = -\sigma c \tag{18b}$$

Therefore, there are two additional parameters α (skew) and ν (tail-thickness) to estimate for this non-Gaussian stochastic volatility model.

Using (18), the third and fourth moments can be written as

$$\begin{aligned}
E[u_t^3] &= \sigma^3 \left(2c^3 + c\big(3 - \frac{\alpha^2}{1+\alpha^2}\big)\frac{\nu}{\nu-3} - 3c\frac{\nu}{\nu-2}\right) \\
E[u_t^4] &= \sigma^4 \left(\frac{3\nu^2}{(\nu-2)(\nu-4)} - 4c^2\big(3 - \frac{\alpha^2}{1+\alpha^2}\big)\frac{\nu}{\nu-3} + 6c^2\frac{\nu}{\nu-2} - 3c^4\right)
\end{aligned}$$

As the volatility innovation is assumed to be Gaussian, the r-th moment of y_t is given by

$$E[y_t^r] = \exp(\frac{ra_0}{2} + \frac{r^2 a_1^2}{8}) E[u_t^r] \tag{19}$$

6 Monte Carlo simulations

In this section, I conduct a small Monte Carlo experiment to assess the finite sample performance of the proposed nonlinear filtering methods. This section is in two parts. As numerical optimization of the likelihood is fairly expensive, I first consider the choice of tuning parameters (such as the number of quadrature nodes and mixture components) by examining how closely the filtered estimates of the state variable track the simulated state variables. Then I examine the finite sample performance of the maximum likelihood estimator.

The Monte Carlo experiments are conducted using (6) as the data generating process with sample size $T = 1000$. For the Gaussian model, the parameter values are set to

$$\alpha_0 = 0, \quad \alpha_1 = 0.45, \quad \phi = 0.975, \quad \rho = \{0, -0.3, -0.6\} \tag{20}$$

These parameter values correspond to one of those used in Harvey and Shephard (1996, Table 1).[8] For the skew-t model, the parameter values are set to

$$\alpha_0 = 0, \quad \alpha_1 = 0.50, \quad \phi = 0.980, \quad \rho = \{0, -0.3, -0.6\}, \quad \alpha = 0, \quad \nu = 11 \tag{21}$$

These parameter values roughly correspond to the estimates obtained for the S&P 500 index with Student-t distribution estimated by Fridman and Harris (1998) and Liesenfeld and Jung (2000).[9]

6.1 Filtering methods

In this subsection, I examine the choice of tuning parameters for the nonlinear filters. I assume the model parameters are known and use the root mean squared error $RMSE =$

[8] $\beta_0 = 1$, $\beta_1 = 0.975$, $\beta_2 = 0.1$ for parameterization (1).

[9] $\beta_0 = 1$, $\beta_1 = 0.980$, $\beta_2 = 0.1$ for parameterization (1).

$\sqrt{\frac{1}{MT}\sum_{i=1}^{M}\sum_{t=1}^{T}(x_t^{(i)} - \overline{x}_{t|t}^{(i)})^2}$ as the performance criterion over $M = 1000$ Monte Carlo replications of sample size $T = 1000$. $x_t^{(i)}$ is the series generated according to (6) for the i-th replication and $\overline{x}_{t|t}^{(i)}$ is the estimated mean of the filtered distribution of the state variable. For the quadrature filter, I compare filters based on the Gauss-Legendre and Gauss-Hermite quadratures for a number of quadrature nodes m. For the mixture Gaussian filter, I examine the choice of the number of mixture components n.

The Gaussian quadrature nodes and weights are obtained from IQPACK (Elhay and Kautsky 1987). For the quadrature filter, the presample (particle) state distribution is calibrated to the unconditional state distribution. To initialize the mixture Gaussian filter, we need to find the weights $a_0^{(i)}$ and first two moments $\mu_0^{(i)}$, $s_0^{2(i)}$ of the component Gaussian distributions that approximate the unconditional state distribution. I consider two methods to determine the initial mixture approximation. Both methods set the means $\mu_0^{(i)} = 0, \pm\delta, \ldots, \pm k\delta$, $k = (n-1)/2$ to an equi-spaced grid about zero and common variances $s_0^{2(i)} = s_0^2$.[10] The weights $a_0^{(i)}$ are then chosen to match the first two moments $E[x_0] = 0$ and $E[x_0^2] = 1$. The first method sets $\delta = 1$ and geometrically declining weights $a_0^{(i)} = a, a\lambda, \ldots, a\lambda^k$. The condition $\sum_{i=1}^{n} a_0^{(i)} = 1$ requires $a = \frac{1-\lambda}{1+\lambda-2\lambda^{n+1}}$. Then the moment matching condition givens the common variance

$$s_0^2 = 1 - \frac{2a\lambda}{(1-\lambda)^3}\big(1 + \lambda - \lambda^n(1 + \lambda + 2(1-\lambda)n + (1-\lambda)^2 n^2)\big)$$

This leaves us with a choice of $0 < \lambda < 1$ such that $s_0^2 > 0$. The second method sets equal weights $a_0^{(i)} = \frac{1}{n}$ and the moment matching condition gives the spacing of means as $\delta^2 = \frac{3(1-s_0^2)}{k(k+1)}$. For this second method, the tuning parameter is the common variance s_0^2 such that $s_0^2 \leq 1$. In the Monte Carlo simulations and empirical applications reported below, I set $\lambda = 0.2$ for the initial mixture with geometrically declining weights and $s_0^2 = 0.1$ for the initial mixture with equal weights. As reported below, the mixture filter appears not to be sensitive to the choice of the initial mixture approximation, especially for applications with a reasonably large sample size.

[10] This assumes an odd number of mixture components n.

Table 1 reports the root mean squared error of the filtered state variable for the data generating processes (20)–(21). I make the following observations. First, for all filters the filtered means become more precise as the magnitude of the leverage effect ρ increases. This is not surprising since the prediction step (8a) uses information from the observed series y_t only if $\rho \neq 0$ in (4b). This suggests the importance of not constraining $\rho = 0$. Second, for the Gauss-Hermite quadrature filter, using more than $m = 100$ nodes does not appear to result in any marked improvement of the filter precision. This is a useful result as the computational cost for using $m = 300$ is quite high. Third, the performance of the mixture Gaussian filter is surprisingly good, both in terms of precision and execution time. This is the case even for the Student-t innovation case. The performance of the mixture Gaussian filter appears not to be sensitive to the specification of the initial mixture distribution and number of mixture components, at least for these data generating processes.

6.2 Maximum likelihood estimation

The proposed maximum likelihood estimation procedure is to maximize the log likelihood as evaluated by (9). In this subsection, I conduct a small Monte Carlo experiment to assess the finite sample performance of the maximum likelihood estimators based on the quadrature and mixture filters. From the filtering results in Table 1 and because of computational costs, for the quadrature filters I restrict attention to those with $m = 100$ nodes. The log likelihood (9) was numerically maximized using the unconstrained trust region BFGS code by Gay (1983). In order to constrain the parameters, I use the one-to-one transformations

$$\phi = \frac{e^{\lambda_1} + 0.5}{e^{\lambda_1} + 1}, \qquad \rho = \frac{e^{\lambda_2} - 1}{e^{\lambda_2} + 1}, \qquad \nu = e^{\lambda_3} + 2 \tag{22a}$$

$$\lambda_1 = \log\left(\frac{\phi - 0.5}{1 - \phi}\right), \qquad \lambda_2 = \log\left(\frac{\rho + 1}{1 - \rho}\right), \qquad \lambda_3 = \log(\nu - 2) \tag{22b}$$

where λ_1, λ_2, λ_3 are the unconstrained parameters to be estimated. Note that while $-1 < \phi < 1$, for practical purposes, I follow Harvey and Shephard (1996) and restrict $0.5 < \phi < 1$.

Tables 2–3 report means and root mean squared errors of the maximum likelihood estimates based on the quadrature and mixture filters. I must note that these results are based on a rather small number of replications ($M = 100$) due to the computational cost of the quadrature filter. With this caveat in mind, I make the following observations. First, as in Table 1, the root mean squared errors decline with the magnitude of the leverage effect ρ. While this effect is not as strong as in Table 1 perhaps due to the small number of replications, it again highlights the importance of not constraining $\rho = 0$. Second, the root mean squared errors for the Gaussian case in Table 2 compares quite favorably against those from the QML estimates reported in Harvey and Shephard (1996). For example, the QML root mean squared errors reported in Harvey and Shephard (1996, Table 1) for $(\phi, \rho) = (0.975, -0.3)$ are 0.088 and 0.298, respectively. Third, as with filtering, the performance of the mixture Gaussian filter is surprisingly good even with only $n = 1$ component.

The most troubling and disappointing result from Tables 2–3 is that more than half of the cases for the Gauss-Hermite quadrature filter fail to converge. An examination of the failed cases reveals that these failures are caused by the estimate of ϕ approaching the stationarity boundary of unity during numerical optimization. I have experimented with alternative transformations to impose $-1 < \phi < 1$ such as $\phi = \frac{2}{\pi} \arctan(\lambda_1)$ and an alternative derivative-free optimizer APPSPACK (Hough, Kolda and Torczon 2001) with little effect to alleviate the problem.[11] It thus appears that while $m = 100$ might be sufficiently accurate for filtering purposes, the Gauss-Hermite quadrature filter requires a larger number of nodes to accurately evaluate the log-likelihood function for parameter estimation. As Figure 1 clearly shows, this is because for large m the Gauss-Hermite quadrature nodes have mostly zero weights compared to the Gauss-Legendre nodes.

7 Empirical application

In this section, I estimate ARSV models with leverage effects to the daily returns from the S&P 500 index for two sample periods, Jan/1980–Dec/1987 ($T = 2022$) and Jan/1990–Dec/2003 ($T = 3532$). The first sample was used by a number of authors to fit a variety of stochastic volatility models (Jacquier, Polson and Rossi 1994, Fridman and Harris 1998, Liesenfeld and

[11] APPSPACK can handle bound constraints without the use of transformations but does not alleviate the convergence failure problem. APPSPACK (Asynchronous Parallel Pattern Search) is available at http://software.sandia.gov/appspack.

Jung 2000, Jacquier et al. 2004, Yu 2004). As can be seen in Figure 2, the first sample contains the stock market crash in October 1987. To ascertain the effect of the extreme spike at the crash on the estimates, I also estimate the model for the second sample which does not contain as extreme a spike as the October 1987 crash.

Tables 4–5 report maximum likelihood estimates of the stochastic volatilty model (6) assuming Gaussian and skew-t innovations for the two sample periods. Robust standard errors are computed using the estimated QML covariance matrix

$$V(\widehat{\theta}) = \tfrac{1}{T} H^{-1} G H^{-1} \tag{23}$$

where the derivatives $H = \frac{1}{T}\sum_{t=1}^{T} \frac{\partial^2 \ell_t}{\partial\theta\partial\theta^\top}$, $G = \frac{1}{T}\sum_{t=1}^{T} \frac{\partial \ell_t}{\partial\theta}\frac{\partial \ell_t}{\partial\theta^\top}$ are evaluated numerically.

Table 4 reports estimates for the sample Jan/80–Dec/87. The maximized log-likelihood values suggest that for the Gauss-Legendre quadrature filter we need to cover at least seven standard deviations of the unconditional state distribution.[12] For the mixture Gaussian filter, the log-likelihood values from estimates based on $s_0^2 = 0.1$ with equal weights were always slightly lower than those based on $\lambda = 0.2$ with geometrically declining weights and are not reported. For the Jan/80–Dec/87 sample, Yu (2004) reports MCMC posterior means of $\phi = 0.972$ and $\rho = -0.3179$ for the Gaussian model. For a Student-t model restricting $\rho = 0$ and $\alpha = 0$, Liesenfeld and Jung (2000) report point estimates of $\phi = 0.986$ and $\nu = 10.718$ which are very close to those reported in Table 4 with $\rho \neq 0$ and $\alpha \neq 0$.

For the sample period Jan/90–Dec/03 reported in Table 5, estimates based on the Gauss-Legendre quadrature do not improve once we cover five standard deviations of the unconditional state distribution. This is expected as there is no obvious extreme spike as the October 1987 crash in the sample. Compared to the Jan/80–Dec/87 sample, the size of the leverage effect ρ and the skew parameter α is much larger in the Jan/90–Dec/03 sample. The estimates obtained from the mixture Gaussian filters are remarkably similar to those obtained from the quadrature filters, especially in the Jan/90–Dec/03 sample. This is an important finding as the computational cost of the mixture Gaussian filter is a fraction of that from the

[12] For the skew-t model, both the Gauss-Legendre and Gauss-Hermite quadrature filters with $m = 100$ failed to converge as $\phi \to 1$.

quadrature filter.[13]

As diagnostic check of the distributional assumption, Figure 3 displays the qq-plots (quantile-quantile plots) for the Gaussian and skew-t models for the two sample periods.[14] As the quantiles of the skew-t distribution are not readily computable, I use the result in Azzalini and Capitanio (2003) who show that if $u_t \sim SK_t(\xi, \sigma^2, \alpha, \nu)$ then $(u_t - \xi)^2/\sigma^2 \sim F(1, \nu)$. Thus for the Gaussian model, we compare the sample quantiles of $\widehat{u}_t^2$ against the $\chi^2(1)$ quantiles where $\widehat{u}_t = y_t/\sigma_{t|t}$ and $\sigma_{t|t}$ is the square root of the filtered volatility. For the skew-t model, we compare the sample quantiles of $(\widehat{u}_t - \widehat{\xi})^2/\widehat{\sigma}^2$ against the $F(1, \widehat{\nu})$ quantiles. If the distributional assumption is correct, the qq-plot should lie on the 45 degree line. Figure 3 shows that for the Jan/80–Dec/87 sample, neither model captures the extreme spike at the October 1987 crash. The fit is much better for the Jan/90–Dec/03 sample but the skew-t model does not appear to provide a much better fit than the Gaussian model, at least according to the qq-plots.

Table 6 compares the sample moments of the daily returns with the implied moments from the estimated model for the two sample periods. The skewness and kurtosis are computed as $E[y_t^r]/(E[y_t^2])^{r/2}$ for $r = 3, 4$, respectively. The moments are given by $E[y_t^3] = 0$ and $E[y_t^4] = 3e^{a_1^2}$ for the Gaussian model and by (19) for the skew-t model. Consistent with the qq-plots in Figure 3, Table 6 indicates that allowing for fat-tails in the mean innovation does not improve much upon the Gaussian model in terms of replicating the observed moments of the return series, especially in the sample containing the October 1987 market crash.

8 Concluding remarks

I have considered two filtering algorithms based on numerical integration for maximum likelihood estimation of stochastic volatility models with leverage effects. In particular, the mixture Gaussian filter was shown to perform remarkably well both in terms of accuracy and

[13] To obtain the maximum likelihood estimates and QML standard errors for the Jan/90–Dec/03 sample (3532 observations), it takes about 7.3 hours using the GL-300-7 filter for the Gaussian model on a P4 2GHz machine. For the same model, the MFG-13 filter takes about 46 seconds on the same machine.

[14] Sample quantiles based on the Gauss-Legendre and Gauss-Hermite quadrature are visually identical as can be expected from the estimates reported in Tables 4–5. To avoid clutter, Figure 3 only plots quantiles based on the Gauss-Legendre filter with $m = 300$ nodes over $[-7, 7]$.

computational cost/robustness. Given its remarkable performance, it appears worth pursuing other applications of the mixture Gaussian filter for estimation of non-linear non-Gaussian state-space models. As the algorithms proposed in the paper are quite simple to implement, I hope they see widespread use among practitioners in the finance field.

In concluding the paper, I suggest a few areas for further research. While the Gauss-Hermite quadrature removes the need to specify the integration bounds, the Monte Carlo results in section 6 indicate that its use in maximum likelihood estimation often fails to converge. As it is desirable to remove the need to specify the integration bounds, one needs to further investigate the instability of the Gauss-Hermite quadrature for estimation purposes. Some preliminary experimentation suggests that using parameterization (1) rather than (6) might improve the performance of the Gauss-Hermite quadrature (albeit with a large number of nodes m). The main difference between the two parameterization is the presence of the Jacobian term $1/\sqrt{1-\phi^2}$ in the conditional densities for parameterization (6). This might be the reason for the optimizer converging towards the boundary $\phi \rightarrow 1$. In this paper, I have preferred parameterization (6) to (1) as the unconditional state density does not depend on the parameters. Moreover, parameterization (6) extends naturally to higher order models as in Meddahi (2001) and Kawakatsu (2004).

While the mixture Gaussian filter was shown to perform remarkably well, a number of issues remain to be investigated. First, the choice of the initial mixture approximation may have an important effect on the performance, especially for short sample sizes. I have only discussed two ad hoc approximations with geometric declining and equal mixture weights. Second, the choice of the number of mixture components needs to be investigated. While I have used a predetermined number of components, one might consider data based methods to “optimally” choose the number of mixture components.

$u_t \sim N(0,1)$

	m	n	$[a,b]$	$\rho = 0$	$\rho = -0.3$	$\rho = -0.6$	secs
QF-GL	300		$[-5,5]$	0.7087	0.6873	0.6114	38.20
QF-GH	300			0.7087	0.6873	0.6114	58.93
	100			0.7087	0.6873	0.6114	5.62
	50			0.7091	0.6881	0.6184	1.27
MF-GW	10	5		0.7087	0.6873	0.6114	0.03
MF-EW	10	5		0.7087	0.6873	0.6114	0.03
MF	10	1		0.7087	0.6873	0.6114	0.01

$u_t \sim t(0,1)$, $\alpha = 0$, $\nu = 11$

	m	n	$[a,b]$	$\rho = 0$	$\rho = -0.3$	$\rho = -0.6$	secs
QF-GL	300		$[-5,5]$	0.6851	0.6650	0.5940	38.26
QF-GH	300			0.6851	0.6650	0.5940	59.98
	100			0.6851	0.6650	0.5942	5.82
	50			0.6875	0.6691	0.6251	1.37
MF-GW	10	5		0.6851	0.6650	0.5940	0.08
MF-EW	10	5		0.6852	0.6652	0.5942	0.08
MF	10	1		0.6851	0.6650	0.5941	0.02

Table 1: Root mean squared errors of the filtered state variable. QF-GL and QF-GH are the quadrature filters based on m nodes of Gauss-Legendre over the interval $[a,b]$ and Gauss-Hermite quadrature, respectively. MF-GW and MF-EW are the mixture Gaussian filters based on n components and m nodes of Gauss-Hermite quadrature with initial weights geometrically declining and equally weighted, respectively. Root mean squared errors based on $M = 1000$ replications of sample size $T = 1000$. The last column indicates the execution time in seconds per replication of the corresponding filter.

$u_t \sim N(0,1)$

	m	n	α_0	α_1	ϕ	ρ	fail
DGP			0	0.45	0.975	0.00	
QF-GL	100		-0.0088	0.4019	0.9651	0.0078	4
			(0.1341)	(0.1782)	(0.0228)	(0.2065)	
QF-GH	100		-0.0100	0.3952	0.9497	-0.0269	53
			(0.1142)	(0.1345)	(0.0316)	(0.2161)	
MF-GW	10	5	-0.0092	0.4064	0.9663	0.0177	0
			(0.1359)	(0.1755)	(0.0226)	(0.2179)	
MF-EW	10	5	-0.0084	0.4074	0.9663	0.0177	0
			(0.1378)	(0.1774)	(0.0226)	(0.2182)	
MF	10	1	-0.0088	0.4068	0.9663	0.0178	0
			(0.1363)	(0.1761)	(0.0226)	(0.2177)	
DGP			0	0.45	0.975	-0.30	
QF-GL	100		-0.0031	0.4118	0.9669	-0.3056	3
			(0.1257)	(0.1563)	(0.0207)	(0.2059)	
QF-GH	100		-0.0002	0.4050	0.9495	-0.3020	59
			(0.1222)	(0.0847)	(0.0306)	(0.2010)	
MF-GW	10	5	-0.0055	0.4129	0.9677	-0.3011	0
			(0.1268)	(0.1560)	(0.0206)	(0.2074)	
MF-EW	10	5	-0.0043	0.4145	0.9678	-0.3032	0
			(0.1292)	(0.1584)	(0.0206)	(0.2069)	
MF	10	1	-0.0048	0.4135	0.9677	-0.3016	0
			(0.1273)	(0.1569)	(0.0206)	(0.2069)	
DGP			0	0.45	0.975	-0.60	
QF-GL	100		-0.0032	0.4275	0.9710	-0.6286	0
			(0.1097)	(0.1111)	(0.0148)	(0.1601)	
QF-GH	100		0.0009	0.4148	0.9593	-0.6013	53
			(0.1047)	(0.1304)	(0.0199)	(0.1808)	
MF-GW	10	5	-0.0053	0.4205	0.9711	-0.6180	0
			(0.1130)	(0.1373)	(0.0147)	(0.2040)	
MF-EW	10	5	-0.0048	0.4220	0.9711	-0.6178	0
			(0.1154)	(0.1394)	(0.0147)	(0.2034)	
MF	10	1	-0.0048	0.4210	0.9711	-0.6177	0
			(0.1133)	(0.1380)	(0.0147)	(0.2038)	

Table 2: Finite sample performance of MLE with Gaussian innovations. QF-GL and QF-GH are the quadrature filters based on m nodes of Gauss-Legendre over the interval $[-5, 5]$ and Gauss-Hermite quadrature, respectively. MF-GW and MF-EW are the mixture Gaussian filters based on n components and m nodes of Gauss-Hermite quadrature with initial weights geometrically declining and equally weighted, respectively. Reported are means over $M = 100$ replications of sample size $T = 1000$. Numbers in parentheses are the root mean squared errors. The last column is the number of cases out of $M = 100$ in which the numerical optimizer failed to converge.

$u_t \sim t(0,1)$, $\alpha = 0$, $\nu = 11$

	m	n	α_0	α_1	ϕ	ρ	α	ν	fail
DGP			0	0.50	0.98	0.00	0	11	
QF-GL	100		−0.0022	0.4481	0.9672	−0.0016	0.0062	15.8675	6
			(0.1746)	(0.2022)	(0.0222)	(0.2043)	(0.4224)	(16.6669)	
QF-GH	100		0.0203	0.4443	0.9518	−0.0116	0.0157	14.9081	57
			(0.1682)	(0.1614)	(0.0321)	(0.1993)	(0.4360)	(10.4903)	
MF-GW	10	5	−0.0012	0.4473	0.9685	−0.0108	0.0055	15.6090	1
			(0.1777)	(0.1976)	(0.0218)	(0.2068)	(0.4112)	(16.9559)	
MF-EW	10	5	−0.0050	0.4489	0.9686	−0.0109	0.0051	15.4918	1
			(0.1764)	(0.1978)	(0.0219)	(0.2069)	(0.4113)	(15.9110)	
MF	10	1	−0.0020	0.4474	0.9685	−0.0109	0.0053	15.5913	1
			(0.1768)	(0.1976)	(0.0218)	(0.2067)	(0.4112)	(16.8141)	
DGP			0	0.50	0.98	−0.30	0	11	
QF-GL	100		−0.0108	0.4441	0.9658	−0.2816	0.0020	14.8346	5
			(0.1658)	(0.1958)	(0.0250)	(0.2334)	(0.4150)	(10.5379)	
QF-GH	100		0.0150	0.4256	0.9516	−0.2710	0.0420	14.5191	53
			(0.1663)	(0.1953)	(0.0350)	(0.2175)	(0.4288)	(7.6994)	
MF-GW	10	5	−0.0058	0.4527	0.9672	−0.3019	0.0055	14.4423	0
			(0.1647)	(0.1712)	(0.0245)	(0.2354)	(0.4042)	(9.9018)	
MF-EW	10	5	−0.0102	0.4541	0.9673	−0.3027	0.0044	14.4008	0
			(0.1645)	(0.1711)	(0.0245)	(0.2362)	(0.4043)	(9.6974)	
MF	10	1	−0.0069	0.4529	0.9672	−0.3021	0.0052	14.4309	0
			(0.1642)	(0.1712)	(0.0245)	(0.2357)	(0.4042)	(9.8307)	
DGP			0	0.50	0.98	−0.60	0	11	
QF-GL	100		−0.0032	0.4501	0.9710	−0.5846	0.0106	14.0938	3
			(0.1423)	(0.1687)	(0.0148)	(0.2545)	(0.3947)	(10.1199)	
QF-GH	100		−0.0217	0.4468	0.9598	−0.5812	0.0096	13.6599	61
			(0.1445)	(0.1582)	(0.0220)	(0.2146)	(0.3808)	(6.1794)	
MF-GW	10	5	−0.0037	0.4656	0.9718	−0.6217	0.0079	13.8454	0
			(0.1426)	(0.1236)	(0.0148)	(0.1902)	(0.3881)	(9.3075)	
MF-EW	10	5	−0.0035	0.4587	0.9719	−0.6042	0.0063	13.8246	0
			(0.1497)	(0.1532)	(0.0148)	(0.2400)	(0.3888)	(9.3052)	
MF	10	1	−0.0039	0.4658	0.9719	−0.6218	0.0075	13.8340	0
			(0.1431)	(0.1234)	(0.0148)	(0.1902)	(0.3883)	(9.2268)	

Table 3: Finite sample performance of MLE with Student-t innovations. QF-GL and QF-GH are the quadrature filters based on m nodes of Gauss-Legendre over the interval $[-5, 5]$ and Gauss-Hermite quadrature, respectively. MF-GW and MF-EW are the mixture Gaussian filters based on n components and m nodes of Gauss-Hermite quadrature with initial weights geometrically declining and equally weighted, respectively. Reported are means over $M = 100$ replications of sample size $T = 1000$. Numbers in parentheses are the root mean squared errors. The last column is the number of cases out of $M = 100$ in which the numerical optimizer failed to converge.

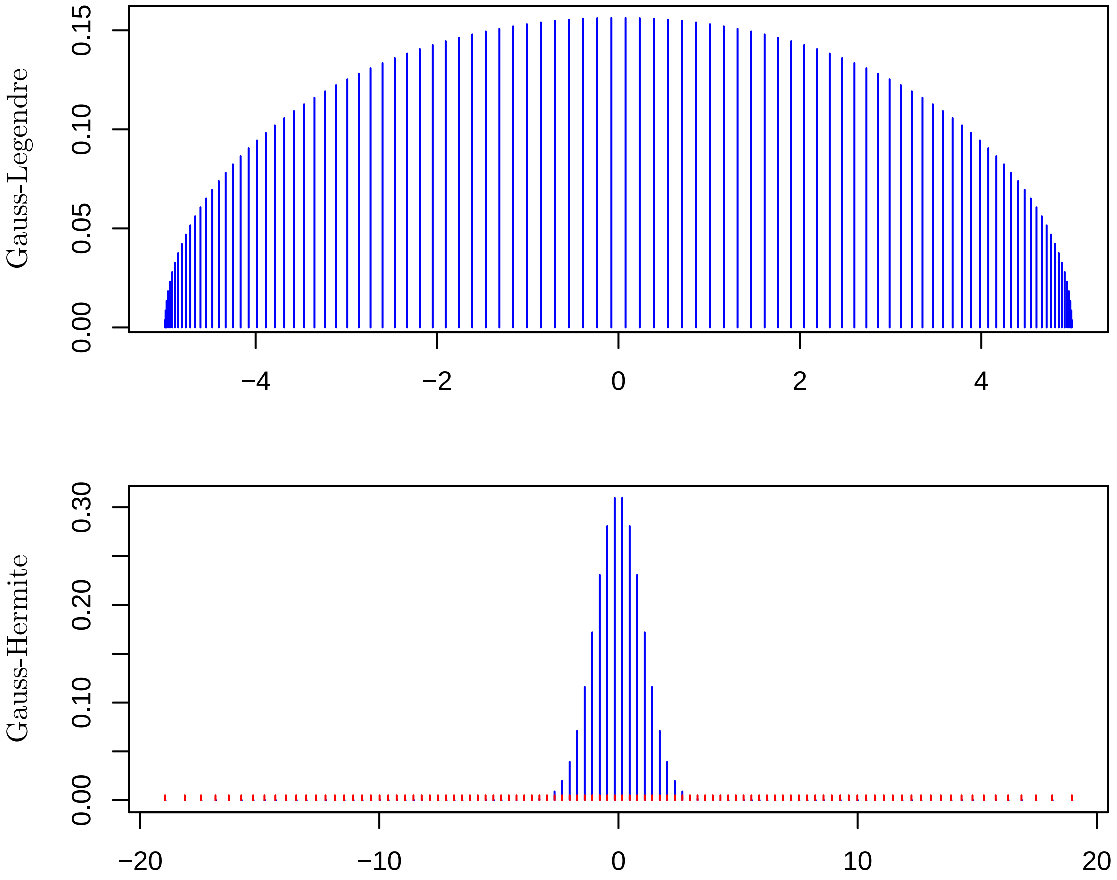

Figure 1: Gaussian quadrature nodes x_i and weights w_i for $m = 100$. The Gauss-Legendre nodes are for the interval $[-5, 5]$. The tick marks for the Gauss-Hermite nodes indicate the positions of the nodes x_i which have mostly zero weight outside the $[-5, 5]$ interval.

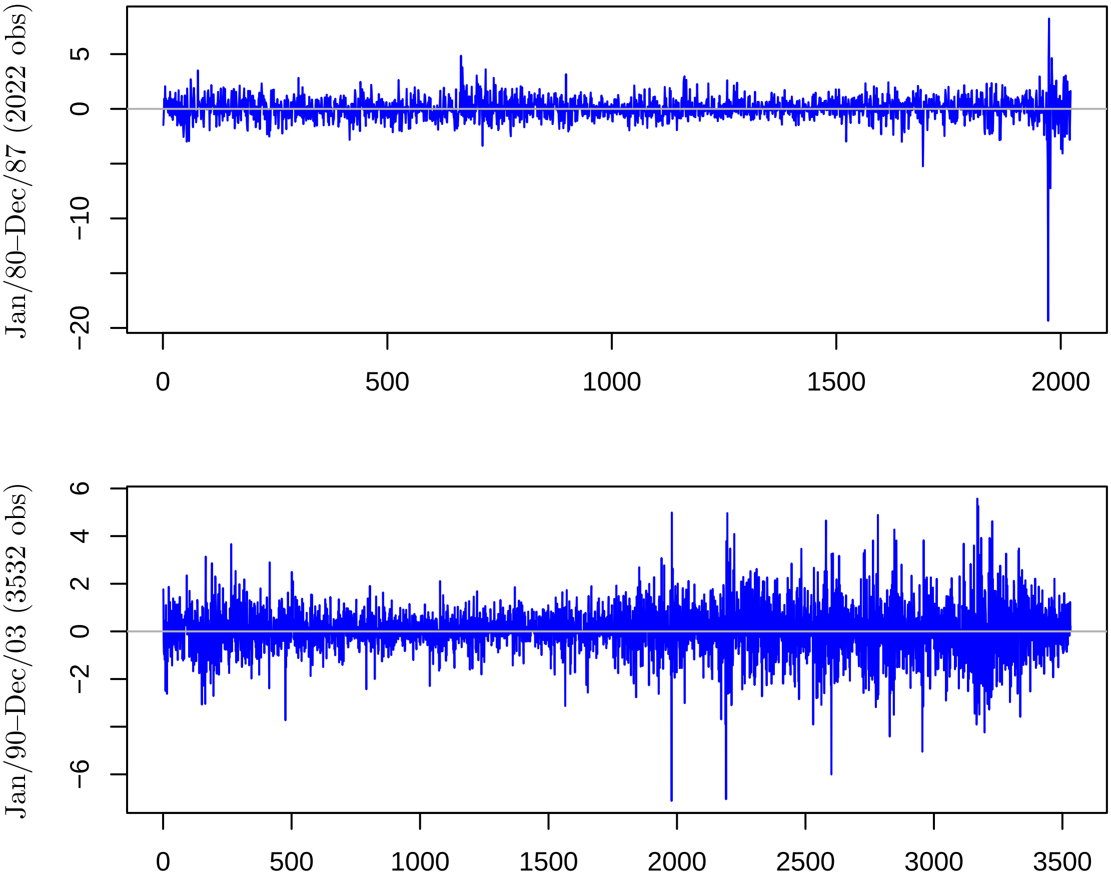

Figure 2: Daily returns from the S&P 500 index. The daily returns are computed as the log difference of the index, dropping any missing observations and treating the resulting sample as equi-spaced.

Jan/80–Dec/87: $u_t \sim N(0,1)$

	α_0	α_1	ϕ	ρ	ℓ
GL-300-5	-0.0535	0.6616*	0.9716*	-0.3383*	-2775.9637
	(0.1143)	(0.0901)	(0.0128)	(0.0872)	
GL-300-7	-0.0817	0.6353*	0.9680*	-0.3346*	-2775.6913
	(0.1175)	(0.1099)	(0.0142)	(0.0857)	
GL-300-9	-0.0817	0.6353*	0.9680*	-0.3346*	-2775.6913
	(0.1175)	(0.1099)	(0.0142)	(0.0857)	
GH-300	-0.0817	0.6353*	0.9680*	-0.3346*	-2775.6913
	(0.1175)	(0.1099)	(0.0142)	(0.0857)	
MFG-13	-0.0818	0.6449*	0.9641*	-0.3542*	-2778.9196
	(0.1115)	(0.1061)	(0.0187)	(0.0832)	
MFG-5	-0.0822	0.6445*	0.9640*	-0.3541*	-2778.9538
	(0.1115)	(0.1060)	(0.0187)	(0.0832)	
MFG-1	-0.0827	0.6440*	0.9639*	-0.3541*	-2778.9878
	(0.1114)	(0.1059)	(0.0188)	(0.0831)	

Jan/80–Dec/87: $u_t \sim SK_t(0,1)$

	α_0	α_1	ϕ	ρ	α	ν	ℓ
GL-300-5	0.0290	0.5785*	0.9859*	-0.4158*	-0.1827*	10.0714*	-2764.3638
	(0.1501)	(0.0881)	(0.0070)	(0.1010)	(0.0305)	(3.2083)	
GL-300-7	0.0289	0.5784*	0.9859*	-0.4158*	-0.1827*	10.0719*	-2764.3637
	(0.1503)	(0.0883)	(0.0070)	(0.1010)	(0.0305)	(3.2088)	
GL-300-9	0.0289	0.5784*	0.9859*	-0.4158*	-0.1827*	10.0718*	-2764.3637
	(0.1503)	(0.0883)	(0.0070)	(0.1010)	(0.0305)	(3.2134)	
GH-300	0.0290	0.5785*	0.9859*	-0.4158*	-0.1827*	10.0703*	-2764.3634
	(0.1507)	(0.0887)	(0.0071)	(0.1010)	(0.0305)	(3.2095)	
MFG-13	0.0354	0.5790*	0.9860*	-0.4274*	-0.1848*	9.7879*	-2764.9785
	(0.1499)	(0.0878)	(0.0071)	(0.1014)	(0.0305)	(2.9578)	
MFG-5	0.0343	0.5781*	0.9859*	-0.4272*	-0.1847*	9.7890*	-2765.0179
	(0.1500)	(0.0873)	(0.0071)	(0.1014)	(0.0305)	(2.9543)	
MFG-1	0.0327	0.5771*	0.9859*	-0.4270*	-0.1846*	9.7866*	-2765.0484
	(0.1498)	(0.0866)	(0.0071)	(0.1014)	(0.0305)	(2.9539)	

Table 4: Maximum likelihood estimates for the Jan/80–Dec/87 sample (2022 observations). Numbers in parentheses are robust standard errors based on (23) and ℓ is the log-likelihood value. For the skew-t distribution (SK_t), α is the skew parameter and ν is the degrees of freedom. * indicates (two-sided) significance at size 0.05. GL-m-b is the quadrature filter based on m Gauss-Legendre nodes over $[-b, b]$, GH-m is the quadrature filter based on m Gauss-Hermite nodes, and MFG-n is the mixture filter with n components.

Jan/90–Dec/03: $u_t \sim N(0,1)$

	α_0	α_1	ϕ	ρ	ℓ
GL-300-5	−0.0916	0.8385*	0.9806*	−0.6747*	−4635.1650
	(0.1162)	(0.0685)	(0.0050)	(0.0457)	
GL-300-7	−0.0916	0.8385*	0.9806*	−0.6747*	−4635.1650
	(0.1162)	(0.0685)	(0.0050)	(0.0457)	
GL-300-9	−0.0916	0.8385*	0.9806*	−0.6747*	−4635.1650
	(0.1162)	(0.0685)	(0.0050)	(0.0457)	
GH-300	−0.0916	0.8385*	0.9806*	−0.6747*	−4635.1650
	(0.1162)	(0.0685)	(0.0050)	(0.0457)	
MFG-13	−0.0913	0.8379*	0.9805*	−0.6768*	−4635.4832
	(0.1148)	(0.0679)	(0.0051)	(0.0449)	
MFG-5	−0.0920	0.8376*	0.9805*	−0.6767*	−4635.5427
	(0.1150)	(0.0678)	(0.0051)	(0.0449)	
MFG-1	−0.0931	0.8370*	0.9804*	−0.6764*	−4635.6145
	(0.1152)	(0.0677)	(0.0051)	(0.0449)	

Jan/90–Dec/03: $u_t \sim SK_t(0,1)$

	α_0	α_1	ϕ	ρ	α	ν	ℓ
GL-300-5	−0.0352	0.8457*	0.9859*	−0.7425*	−0.6029*	12.6937*	−4618.1876
	(0.1219)	(0.0739)	(0.0036)	(0.0356)	(0.0346)	(2.3904)	
GL-300-7	−0.0352	0.8457*	0.9859*	−0.7425*	−0.6029*	12.6937*	−4618.1876
	(0.1219)	(0.0739)	(0.0036)	(0.0356)	(0.0346)	(2.3967)	
GL-300-9	−0.0352	0.8457*	0.9859*	−0.7425*	−0.6029*	12.6937*	−4618.1876
	(0.1219)	(0.0739)	(0.0036)	(0.0356)	(0.0346)	(2.3941)	
GH-300	−0.0311	0.8485*	0.9862*	−0.7437*	−0.6012*	12.6276*	−4618.1566
	(0.1270)	(0.0790)	(0.0046)	(0.0367)	(0.0346)	(2.3834)	
MFG-13	−0.0375	0.8441*	0.9857*	−0.7423*	−0.6068*	12.7539*	−4618.0284
	(0.1204)	(0.0726)	(0.0035)	(0.0355)	(0.0348)	(2.4206)	
MFG-5	−0.0374	0.8437*	0.9857*	−0.7423*	−0.6067*	12.7513*	−4618.0924
	(0.1208)	(0.0726)	(0.0035)	(0.0355)	(0.0348)	(2.4165)	
MFG-1	−0.0374	0.8432*	0.9857*	−0.7423*	−0.6065*	12.7449*	−4618.1693
	(0.1213)	(0.0724)	(0.0036)	(0.0356)	(0.0348)	(2.4230)	

Table 5: Maximum likelihood estimates for the Jan/90–Dec/03 sample (3532 observations). Numbers in parentheses are robust standard errors based on (23) and ℓ is the log-likelihood value. For the skew-t distribution (SK_t), α is the skew parameter and ν is the degrees of freedom. * indicates (two-sided) significance at size 0.05. GL-m-b is the quadrature filter based on m Gauss-Legendre nodes over $[-b, b]$, GH-m is the quadrature filter based on m Gauss-Hermite nodes, and MFG-n is the mixture filter with n components.

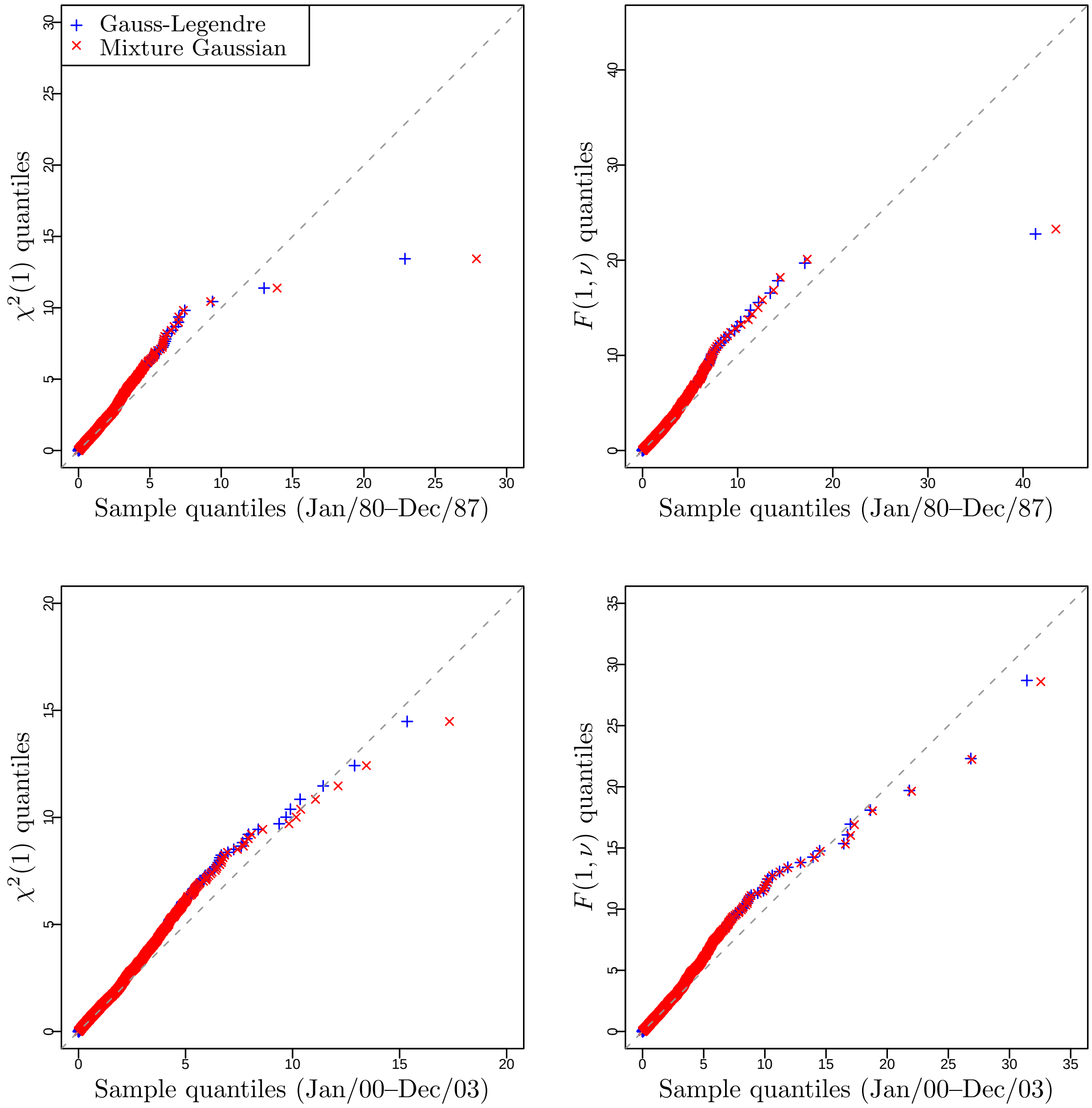

Figure 3: QQ-plots for standardized mean innovation in quadratic forms. The Gauss-Legendre quadrature filter is based on $m = 300$ nodes over $[-7, 7]$ and the mixture Gaussian filter is based on $n = 13$ components. The $\chi^2(1)$ quantile plots are for the Gaussian model, while the $F(1, \nu)$ quantile plots are for the skew-t model.

	Sample	$u_t \sim N(0,1)$		$u_t \sim SK_t(0,1)$	
	Jan/80–Dec/87	GL-300-7	MFG-13	GL-300-7	MFG-13
Skewness	−2.56	0.00	0.00	−0.07	−0.07
Kurtosis	51.68	4.49	4.55	5.58	5.65

	Sample	$u_t \sim N(0,1)$		$u_t \sim SK_t(0,1)$	
	Jan/90–Dec/03	GL-300-7	MFG-13	GL-300-7	MFG-13
Skewness	−0.10	0.00	0.00	−0.23	−0.23
Kurtosis	6.57	6.06	6.05	7.66	7.63

Table 6: Sample and implied moments of S&P 500 index daily returns.

References

Alspach, Daniel L. and Harold W. Sorenson, "Nonlinear Bayesian Estimation Using Gaussian Sum Approximations," *IEEE Transactions on Automatic Control*, 1972, *17*, 439–448.

Azzalini, Adelchi and Antonella Capitanio, "Distributions generated by perturbation of symmetry with emphasis on a multivariate skew-t distribution," *Journal of Royal Statistical Society: Series B*, 2003, *65*, 367–389.

Bølviken, Erik and Geir Storvik, "Deterministic and Stochastic Particle Filters in State-Space Models," in Arnaud Doucet, Nando de Freitas, and Neil Gordon, eds., *Sequential Monte Carlo Methods in Practice*, Springer, 2001, chapter 5, pp. 97–116.

Broto, Carmen and Esther Ruiz, "Estimation Methods for Stochastic Volatility Models: A Survey," *Journal of Economic Surveys*, 2004, *18*, 613–649.

Cappuccio, Nunzio, Diego Lubian, and Davide Raggi, "MCMC Bayesian Estimation of a Skew-GED Stochastic Volatility Model," *Studies in Nonlinear Dynamics & Econometrics*, 2004, *8.*

Elhay, Sylvan and Jaroslav Kautsky, "Algorithm 655 IQPACK: FORTRAN Subroutines for the Weights of Interpolatory Quadratures," *ACM Transactions on Mathematical Software*, 1987, *13*, 399–415.

Fridman, Moshe and Lawrence Harris, "A Maximum Likelihood Approach for Non-Gaussian Stochastic Volatility Models," *Journal of Business & Economic Statistics*, 1998, *16*, 284–291.

Gallant, A. Ronald and George Tauchen, "Reprojecting Partially Observed Systems with Application to Interest Rate Diffusions," *Journal of American Statistical Association*, 1998, *93*, 10–24.

Gay, David M., “Algorithm 611: Subroutines for unconstrained minimization using a model/trust-region approach,” *ACM Transactions on Mathematical Software*, 1983, *9*, 503–524.

Ghysels, Eric, Andrew Harvey, and Eric Renault, “Stochastic Volatility,” in G.S. Maddala and C.R. Rao, eds., *Handbook of Statistics*, Vol. 14, North-Holland, 1996, pp. 119–191.

Harvey, Andrew C. and Neil Shephard, “Estmation of an Asymmetric Stochastic Volatility Model for Asset Returns,” *Journal of Business & Economic Statistics*, 1996, *14*, 429–434.

Hough, Patricia D., Tamara G. Kolda, and Virginia J. Torczon, “Asynchronous Parallel Pattern Search for Nonlinear Optimization,” *SIAM Journal on Scientific Computing*, 2001, *23*, 134–156.

Ito, Kazufumi and Kaiqi Xiong, “Gaussian Filters for Nonlinear Filtering Problems,” *IEEE Transactions on Automatic Control*, 2000, *45*, 910–927.

Jacquier, Eric, Nicholas G. Polson, and Peter E. Rossi, “Bayesian Analysis of Stochastic Volatility Models,” *Journal of Business & Economic Statistics*, 1994, *12*, 371–417.

___ , ___ , and ___ , “Bayesian Analysis of Stochastic Volatility Models with Fat-tails and Correlated Errors,” *Journal of Econometrics*, 2004, *122*, 185–212.

Judd, Kenneth L., *Numerical Methods in Economics*, MIT Press, 1998.

Kawakatsu, Hiroyuki, “Quadratic Stochastic Volatility Models,” 2004. unpublished.

Kitagawa, Genshiro, “Non-Gaussian State-Space Modeling of Nonstationary Time Series,” *Journal of American Statistical Association*, 1987, *82*, 1032–1041.

___ and Will Gersch, *Smoothness Priors Analysis of Time Series*, Springer, 1996.

Kotecha, Jayesh H. and Petar M. Djurić, “Gaussian Sum Particle Filtering,” *IEEE Transactions on Signal Processing*, 2003, *51*, 2602–2612.

Liesenfeld, Roman and Robert C. Jung, "Stochastic Volatility Models: Conditional Normality versus Heavy-Tailed Distributions," *Journal of Applied Econometrics*, 2000, *15*, 137–160.

Meddahi, Nour, "An Eigenfunction Approach for Volatility Modeling," 2001. unpublished.

Omori, Yasuhiro, Siddhartha Chib, Neil Shephard, and Jouchi Nakajima, "Stochastic volatility with leverage: fast likelihood inference," 2004. unpublished.

Yu, Jun, "On Leverage in a Stochastic Volatility Model," 2004. unpublished.