Smooth Particle Filters for Likelihood Evaluation and Maximisation

Michael K Pitt

No 651

WARWICK ECONOMIC RESEARCH PAPERS

DEPARTMENT OF ECONOMICS

THE UNIVERSITY OF WARWICK

Smooth particle filters for likelihood evaluation and maximisation

Michael K Pitt
Department of Economics, University of Warwick, Coventry CV4 7AL
M.K.Pitt@warwick.ac.uk

July 16, 2002

Abstract

In this paper, a method is introduced for approximating the likelihood for the unknown parameters of a state space model. The approximation converges to the true likelihood as the simulation size goes to infinity. In addition, the approximating likelihood is continuous as a function of the unknown parameters under rather general conditions. The approach advocated is fast, robust and avoids many of the pitfalls associated with current techniques based upon importance sampling. We assess the performance of the method by considering a linear state space model, comparing the results with the Kalman filter, which delivers the true likelihood. We also apply the method to a non-Gaussian state space model, the Stochastic Volatility model, finding that the approach is efficient and effective. Applications to continuous time finance models are also considered. A result is established which allows the likelihood to be estimated quickly and efficiently using the output from the general auxilary particle filter.

Some key words: Importance Sampling, Filtering, Particle filter, Simulation, SIR, State space.[1]

[1]The author is grateful for the comments on an early draft of this paper presented at the Tinbergen Institute, Amsterdam, March 2000 and for the more recent comments recieved at the American Statistical Association, Atlanta 2001.

1 Introduction

In this paper we address the problem of likelihood evaluation for state space models via particle filters. We model a time series $\{y_t, t = 1, ..., n\}$ using a state space framework with the $\{y_t|\alpha_t\}$ being independent and with the state $\{\alpha_t\}$ assumed to be Markovian. Particle filters use simulation to estimate $f(\alpha_t|\mathcal{F}_t)$, $t = 1, ..., n$, where $\mathcal{F}_t = \{y_1, ..., y_t\}$ is contemporaneously available information. In this paper we assume a known 'measurement' density $f(y_t|\alpha_t)$ and the ability to simulate from the 'transition' density $f(\alpha_{t+1}|\alpha_t)$. We shall further assume that this state space model is indexed, possibly in both the transition and state equations, by a vector of fixed parameters, θ.

The task we are concerned is the estimation of the likelihood, its log being given by

$$\begin{aligned} \log L(\theta) &= \log f(y_1, ..., y_n|\theta) \\ &= \sum_{t=1}^{n} \log f(y_t|\theta; \mathcal{F}_{t-1}), \end{aligned}$$

via the prediction decomposition. In order to estimate the log-likelihood we exploit the relationship

$$f(y_t|\theta; \mathcal{F}_{t-1}) = \int f(y_t|\alpha_t; \theta) f(\alpha_t|\mathcal{F}_{t-1}; \theta) d\alpha_t. \tag{1·1}$$

Since we have a filtering device, the particle filter, which delivers samples from $f(\alpha_{t-1}|\mathcal{F}_{t-1}; \theta)$, and we can sample from the transition density $f(\alpha_t|\alpha_{t-1}; \theta)$ then it is clear that we can estimate (1·1).

The task in this paper is two fold. Firstly, we consider how we can estimate (1·1) efficiently, statistically and computationally, by using the output from general particle filters. Secondly, we address the problem of providing an estimator for the likelihood which is continuous as a function of the parameters, θ. The second issue is important because it means that the task of performing maximum likelihood inference is greatly facilitated.

The structure of this paper is as follows. In Section 2, we describe particle filters in general and detail the auxiliary particle filter in particular. Section 2·1 introduces a new and effective way of estimating the prediction density (1·1) by using the output which arises from the auxiliary particle filter. We go on, in Section 3, to consider resampling methods which allow likelihood estimation which is continuous as a function of the parameters θ. Section 4 describes how the basic SIR filter of Gordon et al. (1993) may be altered to allow continuous likelihood estimation. Within this section a simple Gaussian state space model is considered and the performance of the proposed simulated method for obtaining the likelihood is compared to the true likelihood, given by the Kalman filter. A number of models are considered within Section 4, including the

stochastic volatility model, Section 4·3. We also consider why the method works well for large time series models, giving an informal justification in Section 4·4.

In Section 5, we move on to consider "fully-adapted" models, which yield more efficient estimation than the standard SIR based procedure. Various models are considered in this section and the GARCH plus error model is examined in depth. Section 6 examines methods of more efficient estimation for quite general models. The details of the methodology is worked out for the stochastic volatility (SV) model.

Section 7 considers an adjustment to the smooth particle filter which allows us to consider continuous time volatility models, see Section 7·1, and locally adapted auxiliary particle filter methods, Section 7·2. Finally, in Section 8 we conclude.

2 Particle Filtering

Simulation based filters are based on the principle of recursively approximating the filtering density $f(\alpha_t|\mathcal{F}_t)$ by a large sample $\alpha_t^1, ..., \alpha_t^M$ with weights $\pi_t^1, ..., \pi_t^M$. If this approximation is regarded as perfect then this leads to the empirical filtering density at time $t+1$,

$$f(\alpha_{t+1}|\mathcal{F}_{t+1}) \propto f(y_{t+1}|\alpha_{t+1}) \sum_{k=1}^{M} \pi_t^k f(\alpha_{t+1}|\alpha_t^k). \tag{2·1}$$

Then the task is to approximate the left hand side by a sample $\alpha_{t+1}^1, ..., \alpha_{t+1}^M$ with weights $\pi_{t+1}^1, ..., \pi_{t+1}^M$. These samples and weights go through to form the empirical filtering density for the next time step and so the process continues, providing filtered samples through time. In the particle filtering literature various methods for producing samples from (2·1) are proposed. Important references include Gordon et al. (1993), Kitagawa (1996), Berzuini et al. (1997), Liu & Chen (1998), Isard & Blake (1996) and Hurzeler & Kunsch (1998). A good review is provided in Doucet et al. (2000b) and in the collection of articles in Doucet et al. (2000a). Recent important work by Andrieu & Doucet (2002) has focused on the analysis of conditionally Gaussian state space models. Filtering by numerical integration procedures have also been developed, an important reference from statistics being Kitagawa (1987) who applied the filter to many non-Gaussian models, including a stochastic volatility model for earthquake data. The particle filtering literature has centred on the on-line filtering of the states and little work has been carried out on parameter estimation via this methodology. An exception is Liu & West (2000) who consider the problem of jointly updating the posterior of the states and fixed parameters in a sequential manner. Before dealing with the issue of parameter estimation we shall focus of the details of particle filtering, regarding the parameters as fixed.

The $\left\{\pi_t^k\right\}$ and $\left\{\pi_{t+1}^k\right\}$ are typically taken as being equal. From this point on, for notational

convenience, we shall assume that all the $\pi_t^k = 1/M$. In this paper we shall follow the approach of Pitt & Shephard (1999), henceforth PS, using the rather general approach of auxiliary particle filtering.

PS showed that the sampling issues raised in particle filtering are best addressed by introducing an auxiliary variable. Such methods are generically called auxiliary particle filters. To avoid any ambiguity, let us denote the transition density as $f_2(\alpha_{t+1}|\alpha_t)$ and the measurement density as $f_1(y_{t+1}|\alpha_{t+1})$. The innovation in PS is to write down the joint density with marginal given by (2·1). Let us start by assuming we have a sample of size M from the filtering density at time t, $\alpha_t^k \sim f(\alpha_t|\mathcal{F}_t)$, $k = 1, ..., M$. We then wish to sample from the following target density

$$\begin{aligned} f(\alpha_{t+1}, k \mid \mathcal{F}_{t+1}) &\propto f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k), \quad k = 1, ..., M \\ &\simeq \overline{g}(k, \alpha_{t+1}) = \overline{g}_1(y_{t+1}|\alpha_{t+1}, k) g_2(\alpha_{t+1}|\alpha_t^k) \\ &= \overline{g}(y_{t+1}|k) g(\alpha_{t+1}|k, y_{t+1}) = C.g(k, \alpha_{t+1}). \end{aligned} \tag{2·2}$$

We have denoted unnormalised densities with a bar. So we now have a joint density $g(k, \alpha_{t+1})$, which approximates the target and that we can sample from, where

$$\overline{g}(y_{t+1}|k) = \int \overline{g}(k, \alpha_{t+1}) d\alpha_{t+1}, \quad g(\alpha_{t+1}|k, y_{t+1}) = \frac{\overline{g}(k, \alpha_{t+1})}{\overline{g}(y_{t+1}|k)},$$

and $C = \sum_{i=1}^M \overline{g}(y_{t+1}|i)$. So for our joint density $g(k, \alpha_{t+1})$, we have

$$g(k) = \frac{\overline{g}(y_{t+1}|k)}{\sum_{i=1}^M \overline{g}(y_{t+1}|i)}, \qquad g(\alpha_{t+1}|k) = g(\alpha_{t+1}|k, y_{t+1}). \tag{2·3}$$

Note that we design our approximations via $\overline{g}(k, \alpha_{t+1})$, see Table 1, so that $g(k)$ can be calculated directly and $g(\alpha_{t+1}|k, y_{t+1})$ is easy to simulate from. We refer to $\overline{g}(y_{t+1}|k)$ as the first stage weights and $g(k)$ as the first stage of probabilities from our SIR scheme. In practise for particular proposals these expressions quickly become simple, as shown in Table 1[2]. For example, for ASIR$_0$, $\overline{g}(y_{t+1}|k) = 1$ so $g(k) = 1/M$ and $g(\alpha_{t+1}|k) = f_2(\alpha_{t+1}|\alpha_t^k)$. For ASIR$_1$, $\overline{g}(y_{t+1}|k) = f_1(y_{t+1}|\widehat{\alpha}_{t+1}^k)$ and $g(\alpha_{t+1}|k) = f_2(\alpha_{t+1}|\alpha_t^k)$. The sampling from $g(k, \alpha_{t+1})$ in all cases is simple. To get a sample of size R, we sample $k^j \sim g(k)$ then $\alpha_{t+1}^j \sim g(\alpha_{t+1}|k)$, for $j = 1, ..., R$.

Having sampled from our joint proposal density (2·3) R times we then allocate weights to the resulting samples $\left(\alpha_{t+1}^j, k^j\right)$, $j = 1, ..., R$,

$$\omega_j = \omega(\alpha_{t+1}^j, k^j), \qquad \pi_j = \frac{\omega_j}{\sum_{i=1}^R \omega_i}.$$

where

$$\omega(\alpha_{t+1}, k) = \frac{f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k)}{\overline{g}_1(y_{t+1}|\alpha_{t+1}, k) g_2(\alpha_{t+1}|\alpha_t^k)}. \tag{2·4}$$

[2] $f_1^{T_1}(y_{t+1}|\alpha_{t+1})|_{\widehat{\alpha}_{t+1}^k}$ represents the exponential of a first order Taylor expansion of $f_1(y_{t+1}|\alpha_{t+1})$ around $\widehat{\alpha}_{t+1}^k$. Similarly $f_1^{T_2}(y_{t+1}|\alpha_{t+1})|_{\widehat{\alpha}_{t+1}^k}$ represents the second order Taylor expansion.

Method	Restrictions	$\overline{g}(k,\alpha_{t+1}) = \overline{g}_1(y_{t+1}\|\alpha_{t+1},k) \times g_2(\alpha_{t+1}\|\alpha_t^k)$	$g(k,\alpha_{t+1}) = g(k) \times g(\alpha_{t+1}\|k)$
ASIR_0	None	$1 \times f_2(\alpha_{t+1}\|\alpha_t^k)$	$\frac{1}{M} \times f_2(\alpha_{t+1}\|\alpha_t^k)$
ASIR_1	None	$f_1(y_{t+1}\|\widehat{\alpha}_{t+1}^k) \times f_2(\alpha_{t+1}\|\alpha_t^k)$	$\frac{f_1(y_{t+1}\|\widehat{\alpha}_{t+1}^k)}{\sum_{i=1}^M f_1(y_{t+1}\|\widehat{\alpha}_{t+1}^i)} \times f_2(\alpha_{t+1}\|\alpha_t^k)$
ASIR_2	f_2 Gaussian.	$f_1^{T_1}(y_{t+1}\|\alpha_{t+1})\|_{\widehat{\alpha}_{t+1}^k} \times f_2(\alpha_{t+1}\|\alpha_t^k)$	see PS
ASIR_3	f_2 Gaussian.	$f_1^{T_2}(y_{t+1}\|\alpha_{t+1})\|_{\widehat{\alpha}_{t+1}^k} \times f_2(\alpha_{t+1}\|\alpha_t^k)$	see PS

Table 1: Some different proposals arising from the ASIR procedure. The third column shows the unnormalised approximating form to the unnormalised target. The final column shows the normalised approximating joint density as $g(k) \times g(\alpha_{t+1}|k)$. This is a little bit more involved to write down for the last two columns, see Pitt & Shephard (1999). Typically $\widehat{\alpha}_{t+1}^k$ is the mean, the mode, a draw, or some other likely value associated with the density of $\alpha_{t+1}|\alpha_t^k$.

We then sample from this discrete distribution, using the normalised weights, π_j, yielding approximate samples from $f(\alpha_{t+1}|\mathcal{F}_{t+1})$. We refer to the ω_j as our second stage weights, and the π_j as our second stage probabilities. Again these quantities become very simple for particular proposals. For the ASIR_0 method, which reduces to the method of Gordon et al. (1993), we have simply $\omega_j = f_1(y_{t+1}|\alpha_{t+1}^j)$. The hope, and frequently the realisation (see PS), is that as we take better approximations than ASIR_0 the second stage weights become less variable leading to more efficient filtering estimation.

Before resampling from the discrete distribution on the R particles, it is more efficient, at each time step, to estimate moments under the filtering density $f(\alpha_{t+1}|\mathcal{F}_{t+1})$ by using the importance sampler approach. We estimate $\mu = E[h(\alpha_{t+1})|\mathcal{F}_{t+1}]$ using the usual importance sample estimator

$$\widehat{\mu} = \sum_{j=1}^{R} \pi_j h(\alpha_{t+1}^j).$$

So far we have advocated the sample-importance-resample (SIR) algorithm of Rubin (1988). Different algorithms, for example MCMC, based upon the auxiliary proposals may also be used, see Pitt & Shephard (2000).

2·1 *Efficient likelihood estimation*

In Section 3, we shall explore methods for estimating likelihoods which are continuous in the parameters. However, prior to this, it is necessary to explore efficient methods for calculating the likelihood. Whilst efficient estimation methods will yield gains in later Sections, the issue

is important in its own right. For instance, we may be interested in estimating the likelihood at single points in the parameter space. We may wish to compare models by evaluating their likelihoods. In addition, we may require to compute the Bayes factor associated with a model. For a time series application see, for instance, Kim et al. (1998). In this case, for model M we require,

$$f(y|M) = \frac{f(y|\theta; M)f(\theta|M)}{f(\theta|y; M)}.$$

For non-Gaussian state space models this problem is non-trivial. The estimation of the denominator at a single parameter value θ is dealt with in Kim et al. (1998) via Markov chain Monte Carlo schemes. However, it is also important that the likelihood in the numerator $f(y|\theta; M)$ be estimated well at the point θ. Efficient estimation also features in the adaptable models considered by PS and their smooth analogues considered in later sections of this paper.

At each time step we wish to estimate the prediction density $f(y_{t+1}|\mathcal{F}_t)$, given by (1·1). Of course since we only have the samples from the filtering density at time t, we actually wish to estimate the empirical analogue,

$$\widehat{f}(y_{t+1}|\mathcal{F}_t) = \int f_1(y_{t+1}|\alpha_{t+1}) \left\{ \sum_{k=1}^{M} f_2(\alpha_{t+1}|\alpha_t^k) \frac{1}{M} \right\} d\alpha_{t+1}. \tag{2·5}$$

This integral cannot, in general, be evaluated directly and so needs to be estimated. There are two aspects to consider at this point. Firstly, we wish to make our method as computationally efficient as possible. Therefore, we would ideally wish to simply use the sample values given in the output from our chosen auxiliary particle filter in the estimation of (2·5). Secondly, we would like to exploit any knowledge, which enabled statistical efficiency in our particle filter, in this estimation. The following theorem addresses both of these concerns.

THEOREM 1:

$$\widehat{f}(y_{t+1}|\mathcal{F}_t, \theta) = \left[\frac{1}{M} \sum_{i=1}^{M} \overline{g}(y_{t+1}|i) \right] E\left[\omega(\alpha_{t+1}; k)\right]$$

where $\omega(\alpha_{t+1}; k)$ is given by (2·4) and the expectation is with respect to $g(k, \alpha_{t+1})$ given by (2·3).

PROOF: See Appendix, Section 10·2.

This above result is useful practically because it means we can take the sample mean of the first stage weights and the sample mean of the second stage weights. So the likelihood $\widehat{f}(y_{t+1}|\mathcal{F}_t, \theta)$ is unbiassedly estimated as

$$\left[\frac{1}{M} \sum_{i=1}^{M} \overline{g}(y_{t+1}|i) \right] \left[\frac{1}{R} \sum_{j=1}^{R} \omega_j \right]. \tag{2·6}$$

For the ASIR$_0$ method of Gordon et al. (1993) this reduces to the standard Monte Carlo estimator

$$\frac{1}{R}\sum_{j=1}^{R}\omega_j = \frac{1}{R}\sum_{j=1}^{R} f_1(y_{t+1}|\alpha_{t+1}^j).$$

When we are able to perform complete adaption, see PS and Section 5, that is we can write

$$f_1(y_{t+1}|\alpha_{t+1})f_2(\alpha_{t+1}|\alpha_t^k) = f(y_{t+1}|\alpha_t^k)f(\alpha_{t+1}|\alpha_t^k; y_{t+1}),$$

then we simply set our joint proposal as $\overline{g}(k, \alpha_{t+1}) = f(y_{t+1}|\alpha_t^k)f(\alpha_{t+1}|\alpha_t^k; y_{t+1})$. In this case (2·6) becomes $\frac{1}{M}\sum_{i=1}^{M} f(y_{t+1}|i)$ so that we are calculating (2·5) directly.

In general, when the ω_j have small variance (our criterion for an efficient particle filter) our estimator will be efficient statistically. In our sampling method we have to evaluate both of the elements of the above product to perform filtering so that the quantities in (2·6) can be regarded as a free bi-product of our auxiliary sampling scheme. The proof also extends quite straightforwardly to fixed lag filtering. Fixed lag filtering is used for robustness and is described in Pitt & Shephard (2000). The efficient estimation method above is exploited in the smooth likelihood designs used later in the paper, see Section 5. We also employ bias correction to estimate the log-likelihood, see Section 10·2.

Before outlining the smooth particle filter approach adopted presently, it is worth noting that there have been numerous alternative attempts to provide algorithms which approximate the filtering densities. Important recent work includes Kitagawa (1987), West (1992), Gerlach et al. (1999) and those papers reviewed in West & Harrison (1997, Ch. 13 and 15). Likelihood maximisation for latent variable time series has also been considered from an importance sampling perspective, see for instance Danielsson & Richard (1993), Durbin & Koopman (1997) and Shephard & Pitt (1997). The importance sampling approach can be problematic as the variance of the importance weights rises rapidly as the dimension (the length of the time series) increases.

The advantage of the approach suggested in this paper is that it provides a reasonably general method based on particle filtering. The problem of dimensionality which arises in importance sampling appears not to be a problem for this method. In the following section, a simple approach is introduced which allows continuous likelihood estimation. Section 4 describes a simple filter which produces continuous likelihood estimation. The example which is examined in Section 4.1 is the AR(1) model which is observed with Gaussian additive noise. This provides a good comparison as we know have the true filter and likelihood solution via the Kalman filter. The model also provides a fair test of the method, as no knowledge of the conjugate relationships are exploited in using the simulation filter. Therefore the performance of the method on this model is informative about the performance on models for which there is no exact likelihood solution.

The model also has a very similar structure to the stochastic volatility model, examined in Section 4·3. In Section 4·2, we consider examples of other models which may be analysed using this framework. In particular, continuous time stochastic differential equations (SDEs) which are observed with noise are considered. This type of model illustrates the advantages of the particle filter approach to likelihood estimation, as importance sampling methods are extremely difficult to apply to such models, and MCMC methods can be very inefficient. In Section 4·3 we consider the basic stochastic volatility model in detail. In Section 4·4, a heuristic justification is proposed for the good performance of the simulated filter approach for long time series.

3 Smooth Likelihood estimation

At present, despite being efficient, our estimator of the likelihood is not continuous as a function of the parameters, θ. This can be quite easily seen for the ASIR_0 method for example. Suppose the samples α_t^k, $k = 1, ..., M$, from the filtering density $f(\alpha_t|\mathcal{F}_t; \theta)$ are changed by a very small quantity. Then the proposal samples α_{t+1}^j, $j = 1, ..., R$ will also change by a very small amount. However, the discrete probabilities, proportional to $f(y_{t+1}|\alpha_{t+1}^j)$, attached to these particles will have changed as well. Since the resampling stage essentially works by generating a uniform and inverting this discrete cumulative distribution function, it is immediately clear that, even with the same uniform variables, the resampled particles will not be close. This illustrates an inherent problem with the bootstrap method for smoothness; it is not smooth. From a practical viewpoint, maximising the resulting rough surface will be extremely problematic, expensive and probably not routinely possible. In order to get around this difficulty a very simple alternative to the discrete resampling is now proposed. This, on its own, is a sufficient innovation to allow continuous likelihood estimation by using the ASIR_0 method. However, it can be combined with other adaptions in order to be used in conjunction with the ASIR_1 method and for complete adaption problems.

3·1 *Smooth bootstrapping*

We record $\widehat{l}_{t+1}$ as our estimate of $\log f(y_{t+1}|\mathcal{F}_t)$, using the method of Section 2·1 combined with our bias correction, see Section 10·2. After running through time, we calculate

$$\widehat{l}(\theta) = \sum_{t=1}^{T} \widehat{l}_t.$$

As we change the parameters θ, we must use the same random numbers and rerun the particle filter again calculating $\widehat{l}(\theta)$ at the new value. In order for the likelihood to be smooth as a function of θ, we require that the samples of α_t from the filter be smooth as a function of θ.

We must consider a smooth analogue of the usual bootstrap procedure which is computationally efficient to sample from in a smooth manner.

Let us think of the bootstrap operating on an $R \times 1$ vector x with sorted univariate elements, x_i and picture the corresponding cumulative distribution function (cdf), see Figure 1. Then instead of the discrete (steps) cdf that we usually use, it is proposed here that we replace it by a smooth cdf. The reason for this is that this will enable smooth sorted samples to be generated. Of course, we are not free to use any smooth cdf we require. A natural choice would be to use the smooth bootstrap approach adopted by, for instance, Efron & Tibshirani (1993) which uses a kernel density approach, see Silverman (1986). However, whilst appealing, this would not enable smooth sorted samples without an expensive $O(R^2)$ algorithm. This is prohibitive, so for univariate data we propose to use a piecewise linear approach. Just as the discrete cdf approaches the true cdf, so our smooth cdf will approach the true cdf as $R \to \infty$. Our construction is simple, although different approaches based upon the partitioning idea here, could be employed. Suppose for each x_i we have probability π_i. Then we define a region i, S_i as follows: $S_i = [x^i, x^{i+1}]$, $i = 1, ..., R-1$. These regions form a partition of the sample space for x. We have different densities $g(x|i)$ within each region i, S_i. We shall assign $\Pr(i) = \frac{1}{2}(\pi_i + \pi_{i+1})$, $i = 2, ..., R-2$ and $\Pr(1) = \frac{1}{2}(2\pi_1 + \pi_2)$, $\Pr(R-1) = \frac{1}{2}(\pi_{R-1} + 2\pi_R)$. Clearly, these probabilities sum to 1. Within each region we shall define the conditional densities as follows,

$$g(x|i) = \frac{1}{(x^{i+1} - x^i)}, \; x \in S_i, \; i = 2, ..., R-2,$$

and

$$g(x|1) = \begin{cases} \frac{\pi_1}{2\pi_1 + \pi_2}, & x = x^1 \\ \frac{\pi_1 + \pi_2}{2\pi_1 + \pi_2} \frac{1}{(x^2 - x^1)}, & x \in S_1 \end{cases}$$

$$g(x|R-1) = \begin{cases} \frac{\pi_R}{\pi_{R-1} + 2\pi_R}, & x = x^R \\ \frac{\pi_{R-1} + \pi_R}{\pi_{R-1} + 2\pi_R} \frac{1}{(x^R - x^{R-1})}, & x \in S_{R-1}. \end{cases}$$

Figure 1 shows a discrete cdf with the continuous interpolation for $R = 8$. Note that the continuous cdf passes through the mid-point of each step in the discrete cdf. As R becomes larger, the two cdfs become indistinguishable. The validity of the resampling method for SIR is preserved. Denoting our continuous cdf by $\widetilde{F}$ and the discrete SIR cdf by $\widehat{F}$, it can be seen that as $R \to \infty$,

$$\widetilde{F}(z) \to \widehat{F}(z) \to F(z),$$

where $F(z)$ is the true cdf. The justification for the convergence of $\widehat{F}(z)$ to $F(z)$ is given rather succinctly by Smith & Gelfand (1992).

The partitioning of the state space means that sampling from this continuous density is very efficient. We simply select the region i with $\Pr(i)$ and sample from $g(x|i)$. The form of $g(x|i)$ has been chosen to be linear. In effect we are simply inverting the smooth cdf given a uniform random variable, which remains fixed as we change the parameters. This ensures continuity and allows very fast sampling but there is no reason why a quadratic or cubic interpolation could not be used within each region, provided of course we maintain the monotonic non-decreasing shape. Indeed differentiability could be achieved by using a higher order interpolation. The algorithm for sampling from this distribution is given in the Appendix and is no more computationally demanding than standard multinomial sampling.

4 Likelihood estimation using ASIR_0

Equipped with the new method of resampling, we may now proceed to generating continuous likelihood estimates via the ASIR_0 particle filter of Gordon et al. (1993). For clarity, we shall outline the specific form of this scheme. It is, of course, simply a special case of the scheme of Section 2. However, before describing the scheme, it is necessary to spell out a couple of additional computational details which surround it. Firstly, we fix the random seeds for a particular complete run (through time) of the filter. So for different parameters θ we will be running the filter to estimate $\log f(y|\theta)$ using the same random numbers for each run. Secondly, we will be sorting the filtered samples prior to the next SIR step. Finally, when generating from both the proposals and the resampling density, we will be using stratified sampling. This stratification scheme is briefly described in Pitt & Shephard (2000) and uses the suggestion of Carpenter et al. (1999). Liu & Chen (1998) and Kitagawa (1996) also discuss methods of stratification. The stratification scheme used here works on the uniform variables which are used to invert our empirical cdf and is detailed in the Appendix. Stratification enables more efficient estimation and can reduce the problem of sample impoverishment. Pitt & Shephard (2000) demonstrate that using this approach at both the sample and resample stages produces efficient estimation for the filtered means.

The computational details out of the way, we can now outline the specific form of the ASIR_0 scheme. In the usual way, see Section 2, let us assume that at time t we have the sorted (in ascending order) samples $\alpha_t^k \sim f(\alpha_t|\mathcal{F}_t)$, $k = 1, ..., M$. Of course this filtering density is indexed by θ but for brevity we shall drop this from our notation. Using our stratified sampling method we sample R of these and pass through the transition density. We then sort in ascending order to obtain α_{t+1}^j, $j = 1, ..., R$. These will be distributed according to $f(\alpha_{t+1}|\mathcal{F}_t)$. We attach to

each of these samples the probability π_j defined through

$$\omega_j = f(y_{t+1}|\alpha_{t+1}^j), \qquad \pi_j = \frac{\omega_j}{\sum_{i=1}^R \omega_i}.$$

We now apply the smooth bootstrap approach of Section 3·1. We use a set of M sorted stratified uniforms and invert the continuous cdf corresponding to the density of Section 3·1. Hence we now have a sorted sample of size M, α_{t+1}^i, $i = 1, ..., M$. In the usual way, these arise approximately from $f(\alpha_{t+1}|\mathcal{F}_{t+1})$ and we can proceed to the next time step. We record $\widehat{l}_{t+1} = \log \widehat{f(y_{t+1}|\mathcal{F}_t, \theta)}$ as our estimate of $\log f(y_{t+1}|\mathcal{F}_t)$, using the method of Section 2·1 combined with our bias correction, see Section 10·2. After running through time, we calculate

$$\widehat{l}(\theta) = \sum_{t=1}^{T} \widehat{l}_t.$$

Provided that the transition density and the measurement density are continuous in α_{t+1} and θ, then this is sufficient to ensure that $\widehat{l}(\theta)$ is continuous in θ. It should be noted that we need to sort once for each time update. This is the only computational addition to the algorithm of Gordon et al. (1993). Indeed it is $O(R \log R)$. However, this is found to be a relatively innocuous problem as part of these algorithms, largely because the sorting algorithm exploits the fact that the variables are close to being sorted prior to being sorted. Timings indicate that with the numbers concerned the sorting takes less than 1% of the CPU time. It is only when R becomes extremely large that this consideration will play a part.

4·1 *Example 1: AR(1) + noise model*

To assess the performance of the ASIR$_0$ method we shall consider the AR(1) plus noise model. This is a linear state space form model, see Harvey (1993), the likelihood for which can be evaluated via the Kalman filter. The model is,

$$\begin{array}{rcll} y_t & = & \alpha_t + \varepsilon_t, & \varepsilon_t \sim N(0, \sigma_\varepsilon^2) \\ \alpha_{t+1} & = & \mu + \phi(\alpha_t - \mu) + \eta_t, & \eta_t \sim N(0, \sigma_\eta^2). \end{array} \tag{4·1}$$

To mimic the stochastic volatility (SV) model, see Section 6·1.1 we have $\sigma_\varepsilon^2 = 2$, $\sigma_\eta^2 = 0.02$, $\phi = 0.975$ and $\mu = 0.5$. The choice of σ_η^2, ϕ and μ are chosen as typical values for the SV model, ϕ representing the persistence in variance, whilst σ_ε^2 is chosen from the curvature for the measurement density in the SV model (the second derivative of $\log f(y_t|\alpha_t)$ with respect to α_t). Thus the AR(1) plus noise model above provides a fair test of our method. "Fair" because we are not exploiting any particular features of the conjugate Gaussian updating relations that exist. This assessment is therefore extremely informative about the performance of the smooth ASIR$_0$ procedure for non-Gaussian models. Before examining the behaviour of our estimator, we shall first take a preliminary glance at the output from the ASIR$_0$ filter compared with the

true output. We simulate one reasonably large time series, $T = 5000$, and examine the results for differing M and R. In Figure 2, we take small values of M and R as 300 and 400 respectively. We plot the true filter mean, based on the true values of the parameters, from the Kalman filter $m_t = E[\alpha_t|\mathcal{F}_t]$ together with its estimate $\widehat{m}_t$ from the smooth ASIR$_0$ procedure. Similarly we look at the log-likelihood component l_t, from the Kalman filter and the corresponding estimate $\widehat{l}_t$ from the ASIR$_0$ method, see previous section. The error $\widehat{l}_t - l_t$ is also displayed. Finally a slice through the log-likelihood for μ is taken, keeping the remaining parameters fixed at their true values. The profile log-likelihood against μ, from both the Kalman filter and the estimator of the log-likelihood are displayed. Figure 3 shows the same results as Figure 2, except that M and R are now taken as 3500 and 5000 respectively.

Several things are clear from these plots. The estimated filter mean $\widehat{m}_t$ remains very close to m_t, even with the large time series. Also, the estimates of $\widehat{l}_t$ appear to be unbiassed for l_t, as indicated by the zero mean of the error $\widehat{l}_t - l_t$. The variation in this error, although apparently changing over time, does not increase but appears stable. This variation also goes down in the manner we would expect by increasing both M and R, as is apparent by comparing the error $\widehat{l}_t - l_t$ in the two figures. As far as the profile log-likelihoods are concerned, the estimator appears to do rather well, even for small M and R, being very close in both figures to the true Kalman filter log-likelihood. We also found this to be true when looking at the profiles of the other two parameters, σ_η^2 and ϕ.

We now examine an extremely long time series, of length $T = 20000$. The resulting profile likelihood and log-likelihood for the three parameters are displayed in Figure 4. Clearly, the estimation is remarkably good for each of the parameters considering the length of the time series.

The fact that the method performs well even for small sample values over such a long series is encouraging indicating the scheme is robust. The errors is the log-likelihood components l_t are stable over time and the variance, although changing, appears stationary. This contrasts with importance sampling for which the variation in the likelihood estimator can increase dramatically with the time series length. A heuristic justification for why this is the case is given in Section 4·4.

4·1.1 Estimator performance

We now examine the behaviour of our estimator by simulating two time series of length 150 and 550 from the above model. We then estimate via maximum likelihood using the Kalman filter to obtain the correct maximum likelihood estimate. The estimation is with respect to $\theta = (\sigma_\eta, \mu, \phi)$ keeping σ_ε^2 fixed at the true value of 2. We then run the smooth ASIR$_0$ filter 50 times, with

different random number seeds for each run, maximising[3] the resulting estimated log-likelihoods for each run with respect to θ. Recorded in Table 2 are the results for $T = 150$, using varying values of M and R. The average of the 50 simulated maximum likelihood estimates, the 50 variance estimates and the mean squared error are displayed for each set of M, R. The variance estimates are obtained by taking the negative of the inverse of the matrix of second derivatives for θ at the mode. It can be seen that the biases in all cases are not significantly[4] different from 0. The mean squared errors in Table 2 are small relative to the variation in the data, and become smaller as M, R increase. In addition the variance-covariance matrix is well estimated even for small M, R. These results are very encouraging.

The results for the case $T = 550$, Table 3, gives an insight into how the method might behave for the SV model for which the data is reasonably long. The results are displayed in an identical fashion to Table 2. If an importance sampler were used to estimate the likelihood, we might expect a rapid decline in the performance as T becomes larger. This is not the case here. The biases are again not significant given our simulation number of 50, suggesting the estimator is, at least approximately, unbiassed for the true ML estimator. The mean squared errors become smaller as M, R increase. Indeed the magnitude of the mean squared errors suggests that the method is workable for practical ML estimation of non-linear, non-Gaussian models in state space form.

4·2 *More general models*

There are many models which may be placed in a similar form. The exponential measurement models considered in West & Harrison (1997, Ch. 13 and 15), for instance, where we have

$$\alpha_{t+1} = \mu + \phi(\alpha_t - \mu) + \eta_t,$$

and a link function relating the evolving state to the measurement. For instance, for count data we may have

$$y_t \sim Po(\exp(\beta' x_t + \alpha_t)).$$

The smooth ASIR_0 method above may be used to estimate this model, a Poisson model with a time varying intercept, straightfowardly.

Also we may apply the smooth filter to problems involving a stochastic differential equation (SDE) observed with noise. For instance we may have the following model,

$$y(\tau_i) \quad \sim \quad f(y(\tau_i) \mid x(\tau_i))$$

[3] We use the `BFGS()` routine within `Ox`, a matrix language. See http://hicks.nuff.ox.ac.uk/Users/Doornik/.

[4] We have to test $bias \sim N\left(0, \frac{MSE}{50}\right)$.

$$dx(t) \quad = \quad \mu(x(t))dt + \sigma(x(t))dW(t),$$

where observations are made a times $\tau_1, \tau_2, ..., \tau_n$. In this case we can simulate smoothly from $f(x(\tau_{i+1}) \mid x(\tau_i))$ by exploiting the Euler approximation. Letting $\delta = (\tau_{i+1} - \tau_i)/M$, $z_o = x(\tau_i)$ we simulate through

$$z_{t+1} = z_t + \mu(z_t)\delta + \delta^{\frac{1}{2}}\sigma(z_t)u_t, \quad t = 0, ..., M-1, \tag{4·2}$$

where u_t is a standard Gaussian random variable. Then we set $x(\tau_{i+1}) = z_M$, which will be distributed according to $f(x(\tau_{i+1}) \mid x(\tau_i))$ as $M \longrightarrow \infty$, as the Euler approximation becomes exact. Now that we have a simple method for simulating from $f(x(\tau_{i+1}) \mid x(\tau_i))$ the ASIR_0 method can be applied straightforwardly.

This is an important application as analysis of such models is difficult using MCMC or importance sampling methods. Another, purely time series, application is to stochastic volatility (SV) models. We shall examine this model in greater detail below.

4·3 *Example 2: SV model*

We can examine the method by considering the stochastic volatility (SV) model,

$$y_t = \varepsilon_t \exp(\alpha_t/2), \quad \alpha_{t+1} = \mu + \phi(\alpha_t - \mu) + \eta_t, \tag{4·3}$$

where ε_t and η_t are independent Gaussian processes with variances of 1 and σ^2 respectively. This is a non-linear time series model of evolving scale. Here μ has the interpretation long run mean of the log volatility, ϕ the persistence in the volatility shocks and σ_η^2 is the volatility of the volatility. This model has attracted much recent attention in the econometrics literature as a way of generalizing the Black-Scholes option pricing formula to allow volatility clustering in asset returns; see, for instance, Hull & White (1987), Harvey et al. (1994) and Jacquier et al. (1994). MCMC methods have been used on this model by, for instance, Jacquier et al. (1994), Shephard & Pitt (1997) and Kim et al. (1998). In addition, Shephard & Pitt (1997) and Durbin & Koopman (1997) consider importance sampling to obtain the likelihood.

In this case we have,

$$\log f(y_{t+1}|\alpha_{t+1}) \equiv l(\alpha_{t+1}) = c - \frac{1}{2}y_{t+1}^2 \exp(-\alpha_{t+1}) - \frac{1}{2}\alpha_{t+1}.$$

We have apply the ASIR_0 method by first simulating a single series of length 550 with $\theta = (\sigma_\eta, \mu, \phi) = (\sqrt{0.02}, 0.5, 0.975)$, typical values for daily returns. We then carry out a Monte Carlo experiment, taking the number of Monte Carlo replications as 50, in a similar fashion

to the previous section. We use the variance of the scores, the outer-product estimator, for estimating the variance covariance matrix. The results are reported, again for varying values of M and R in Table 4. The corresponding histograms of the Monte Carlo output are given in Figures 5 to 7. It is informative to consider the ratio of the mean squared error to the variance of each parameter with respect to the data. For the case $M = 300$, $R = 400$ this is (0.035, 0.0172, 0.0307) for each of the parameters $\theta = (\sigma_\eta, \mu, \phi)$. This reduces to (0.0060, 0.0027, 0.0064) when $M = 1000$, $R = 1300$. When $M = 3000$, $R = 4000$, the ratio is just (0.00167, 0.00111, 0.00143). The histograms of Figures 5 to 7 show that there are no extreme results from the Monte Carlo experiment. Indeed the histograms do not appear to be far from normality. This is encouraging as it means that our simulated maximum likelihood procedure is robust.

The stochastic volatility model can be extended in a number of directions without any particular difficulty. A simple extension would be to include a risk premium term for stock returns so that

$$y_t \sim \mathsf{N}(\mu_y + \beta \exp(\alpha_t); \exp(\alpha_t)).$$

This may also be expressed as

$$y_t = \mu_y + \beta \exp(\alpha_t) + \exp(\alpha_t/2)\varepsilon_t.$$

A further extension is to allow shocks to returns and volatility to move together so that

$$\begin{pmatrix} \varepsilon_t \\ \eta_t \end{pmatrix} \sim \mathsf{N}\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 & \rho\sigma_\eta \\ \rho\sigma_\eta & \sigma_\eta^2 \end{pmatrix} \right),$$

where ρ is the correlation between the two shocks. Additionally, heavy tailed conditional distributions, for ε_t, can be used without posing any difficulty for the methods advocated here.

The stochastic volatility model above follows from an Euler discretisation of the Ornstein-Uhlenbeck (OU) process. Alternative models for the evolution of volatility in continuous time have been proposed and their treatment is considered in Section 7·1.

4·4 *Estimator for large time series*

We shall now give a heuristic justification for the good performance of the particle filter for large time series, see the results of Section 7. Essentially, the accuracy for large time series in the estimation of the likelihood is due to the fact that the errors in the score have a mean which is zero and arise as the addition of the score error associated with each data point. The region in which the likelihood is appreciable contracts as N, the sample size, becomes larger and this counteracts the summation of score errors which also takes place over the sample size.

Consider the true log-likelihood which is given by,

$$l_N(\theta) = \sum_{t=1}^{N} l_i(\theta),$$

where N is the time series length and where $l_i(\theta) = \log f(y_i|\theta)$ and $y_i \sim f(y_i|\overline{\theta})$, $\overline{\theta}$ being the true parameter. Now let us suppose that we obtain our estimated likelihood as,

$$\widehat{l}_N(\theta) = \sum_{t=1}^{N} \widehat{l}_i(\theta), \quad \widehat{l}_i(\theta) = l_i(\theta) + \varepsilon_i(\theta),$$

where the random error term $\varepsilon_i(\theta)$ is a smooth differentiable function of θ and $E[\varepsilon_i(\theta)] = c$, a constant. We get

$$\begin{aligned} \widehat{l}_N(\theta) &= l_N(\theta) + \sum_{t=1}^{N} \varepsilon_i(\theta) = l_N(\theta) + \varepsilon(\theta), \\ \widehat{l'}_N(\theta) &= l'_N(\theta) + s(\theta) \end{aligned}$$

The score error $s(\theta) = \partial\varepsilon/\partial\theta$, is given by

$$s(\theta) = \sum_{t=1}^{N} s_i(\theta),$$

which has mean 0 (since we may differentiate the expectation for $\varepsilon_i(\theta)$) and variance $N\sigma^2$. Note that the variance goes up only linearly with N. This is exactly what we require as we are interested in the distance,

$$d = \widehat{l}_N(\theta + \Delta) - l_N(\theta + \Delta) - \{\widehat{l}_N(\theta) - l_N(\theta)\}.$$

Intuitively this gives us the error in the shape. Since we are concerned with distances which are local for likelihood estimation, we may consider distances $\Delta = \delta/\sqrt{N}$, as on asymptotic grounds this is the scale of interest. Taking $\delta \longrightarrow 0$, we obtain,

$$d = \frac{\delta \times (\widehat{l'}_N(\theta) - l'_N(\theta))}{\sqrt{N}} = \frac{\delta \times s(\theta)}{\sqrt{N}}.$$

Hence

$$E[d] = 0, \quad Var[d] = \sigma^2.$$

This indicates that the local error, d, is independent of the sample size N. This means that the local error in our likelihood remains the same. The likelihood estimation method should therefore work as well for large samples (for which we are concerned with small distances) as it will for small samples.

We now look at a Monte Carlo simulation to see how the variance and mean of the score changes as N increases. We examine the AR(1) plus noise model of (4·1) comparing the true derivative (with respect to σ_η) with the results of 50 runs of the smooth particle filter with differing M and R. The results are given in Table 5. The variance in the derivative is small in all cases compared to the magnitude of the true derivative. We notice that as the time series varies in length the variance does appear to go up linearly with N. Also the variance goes down as M increases in the manner which we would expect.

5 Full adaption

In the case of full adaption, as described in PS, we can draw directly from $f(\alpha_{t+1}|\alpha_t, y_{t+1})$ and also evaluate $f(y_{t+1}|\alpha_t)$. This situation actually arises in a number of important special cases of the general model considered. Suppose, as before that we have sorted samples $\alpha_t^1, ..., \alpha_t^M$ from $f(\alpha_t|\mathcal{F}_t)$. We associate with each of these samples the weight $\omega_j = f(y_{t+1}|\alpha_t^i)$, $i = 1, ..., M$. We then resample M times, using the smooth bootstrap, yielding M sorted samples α_t^j, $j = 1, ..., M$. Clearly, these samples are approximately drawn from $f(\alpha_t|\mathcal{F}_{t+1})$. We now simply pass each of these samples through the density $f(\alpha_{t+1}|\alpha_t, y_{t+1})$ to produce a sample $\alpha_{t+1}^j \sim f(\alpha_{t+1}|\mathcal{F}_{t+1})$, $j = 1, ..., M$. Since we have smoothness in our samples as a function of our parameters, the estimated likelihood will be continuous in the parameters. The fully adapted algorithm is optimal, one step ahead, since we are directly sampling from (2·1).

In fact the empirical prediction density, (2·5) is exactly obtained as

$$\widehat{f}(y_{t+1}|\mathcal{F}_t, \theta) = \frac{1}{M} \sum_{k=1}^{M} f_1(y_{t+1}|\alpha_t^k),$$

where $\alpha_t^k \sim f(\alpha_t|\mathcal{F}_t)$. As in the remainder of this paper, the computational additions considered in Section 4, such as stratified sampling, are again used. This method should be extremely efficient for fully adapted models such as the ARCH plus noise model described presently.

5·1 *Example 1: ARCH with error*

An example of full adaption is for the ARCH model observed with Gaussian error. The autoregressive conditional heteroskedasticity (ARCH) models, see Engle (1995), are used to model the slowly changing variance commonly observed, for example, in equity and exchange rate returns. Consider the simplest Gaussian ARCH model, the ARCH(1), observed with independent Gaussian error, see Shephard (1996). So we have

$$y_t|\alpha_t \sim N(\alpha_t, \sigma^2), \qquad \alpha_{t+1}|\alpha_t \sim N(0, \beta_0 + \beta_1\alpha_t^2).$$

It has received a great deal of attention in the econometric literature as it has some attractive multivariate generalizations: see the work by Diebold & Nerlove (1989), Harvey et al. (1992) and King et al. (1994). This model is exactly adaptable. It is clear to see that,

$$y_{t+1}|\alpha_t \sim N(0, \beta_0 + \beta_1\alpha_t^2 + \sigma^2), \qquad \alpha_{t+1}|\alpha_t, y_{t+1} \sim N(a, b^2),$$

where

$$b^2 = \frac{\sigma^2(\beta_0 + \beta_1\alpha_t^2)}{\beta_0 + \beta_1\alpha_t^2 + \sigma^2}, \qquad a = b^2\frac{y_{t+1}}{\sigma^2}.$$

As far as we know no likelihood methods currently exist in the literature for the analysis of this type of model (and its various generalizations) although a number of very good approximations have been suggested.

5·2 *Example 2: Threshold models*

Let us consider the following binary threshold model,

$$\begin{aligned} y_t &= \begin{cases} 1, & \alpha_t > 0 \\ 0, & \alpha_t < 0 \end{cases} \\ \alpha_{t+1} &= \mu + \phi(\alpha_t - \mu) + \eta_t, \quad \eta_t \sim N(0, \sigma_\eta^2). \end{aligned}$$

We have marginally,

$$\Pr(y_t = 1) = \Phi\left(\frac{\mu}{\sigma}\right)$$

where $\sigma^2 = \frac{\sigma_\eta^2}{1-\phi^2}$. This can be fully adapted. If $y_{t+1} = 1$,

$$\Pr(y_{t+1}|\alpha_t) = \Phi\left(\frac{\mu + \phi(\alpha_t - \mu)}{\sigma_\eta}\right), \quad f(\alpha_{t+1}|y_{t+1}, \alpha_t) = \mathrm{TN}_{>0}(\mu + \phi(\alpha_t - \mu); \sigma_\eta^2).$$

If $y_{t+1} = 0$,

$$\Pr(y_{t+1}|\alpha_t) = 1 - \Phi\left(\frac{\mu + \phi(\alpha_t - \mu)}{\sigma_\eta}\right), \quad f(\alpha_{t+1}|y_{t+1}, \alpha_t) = \mathrm{TN}_{<0}(\mu + \phi(\alpha_t - \mu); \sigma_\eta^2).$$

Again, this model is exactly adaptable allowing efficient estimation of the likelihood.

5·3 *Example 3: GARCH with error*

As a realistic example to consider, we examine the GARCH with error model. This is a more general model than the one previously outlined. We can write this model in the following form,

$$\begin{aligned} y_t|\alpha_t &\sim N(\alpha_t, \sigma^2), \\ \alpha_t|\sigma_t^2 &\sim N(0, \sigma_t^2) \\ \sigma_{t+1}^2 &= \beta_0 + \beta_1\alpha_t^2 + \beta_2\sigma_t^2. \end{aligned} \tag{5·1}$$

We can equivalently write the above model as

$$\begin{aligned} y_t|\sigma_t^2 &\sim N(0, \sigma^2 + \sigma_t^2), \\ \alpha_t|\sigma_t^2, y_t &\sim N\left(\frac{b^2 y_t}{\sigma^2}; b^2\right), \\ \sigma_{t+1}^2 &= \beta_0 + \beta_1\alpha_t^2 + \beta_2\sigma_t^2, \end{aligned} \tag{5·2}$$

where $b^2 = \frac{\sigma^2\sigma_t^2}{\sigma^2+\sigma_t^2}$. This is may be thought of as the "adaptable" form of the model since σ_t^2 is a deterministic function of α_{t-1}^2 and σ_{t-1}^2. Of course, in either specification it is immediate

that as $\sigma^2 \longrightarrow 0$ we obtain the GARCH(1,1) model. In fact, the second equivalent form of the model involves a univariate Markov chain in σ_t^2 which is observed with noise. It is immediately apparent from the form of (5·2) that σ^2 represents a lower bound on the overall variance of y_t. Suppose that at time t, we have M samples from $f(\sigma_t^2|Y_t)$. We sample from $f(\alpha_t|y_t, \sigma_t^2)$, given above, R times, where we choose the σ_t^2 randomly from our sample from $f(\sigma_t^2|Y_t)$. In this way since we obtain R samples $\sigma_{t+1}^{2(i)}$, $i = 1, ..., M$ from $f(\sigma_{t+1}^2|Y_t)$. Regarding these as being sorted in ascending order we now apply the smooth bootstrap method where we have weights $\omega_i = f(y_{t+1}|\sigma_{t+1}^{2(i)})$, $i = 1, ..., M$.

We illustrate this method by estimating the four parameters. We simulate a time series of length 500 and perform 100 different ML estimation procedures using the above method. The four parameters $(\beta_0, \beta_1, \beta_2, \sigma)'$ are set to $(0.01, 0.2, 0.75, 0.1)'$ in the single simulation. The procedure is then run 100 times with $M = 500$, $R = 600$. The results are shown in Table 6. Clearly, unlike the Gaussian state space form model, we cannot analytically assess the performance of our estimator by comparison with the true ML estimator. However, for all 100 runs (started with different random number seeds), we encountered no problem with convergence to the mode. The variance of the simulated maximum likelihood are many hundreds of times smaller than the variance obtained by inverting the matrix of second derivatives at the mode. In addition, the true values of the parameters lie well within their 95% confidence limits. This suggests our approach is a fast, simple and reliable procedure for a problem for which a likelihood solution is non-trivial. The smooth likelihood procedure, of course, allows testing to be carried out. For instance, likelihood ratio tests can be routinely undertaken.

We shall illustrate this by examining an actual time series of length 500, consisting of the continuously compounded daily returns on the Pound versus the US dollar from the second of January 1981 to the twelfth of December 1982. The maximum likelihood results for both the GARCH and GARCH plus error models are reported in Table 7. The GARCH model is nested within the GARCH + error model, arising from the restriction that $\sigma = 0$. Therefore using the likelihood ratio test we can reject the null hypothesis that $\sigma = 0$ at the 1% level of significance. We therefore favour our richer GARCH(1,1) plus error model for this dataset.

6 Partial adaption

The number of models for which full adaption is possible is fairly limited. Partial adaption may be carried out for more general models. In the following sections we shall consider local partial adaption methods which exploit knowledge of the mixture component α_t^k. Whilst these methods represent the natural smooth analogue of the auxiliary methods of PS, their design is a little

more involved. Global partial adaptions, which do not exploit knowledge of α_t^k, are more easily dealt with and shall be considered here. From Section 2, we have

$$\begin{aligned} f(\alpha_{t+1}, \alpha_t^k | \mathcal{F}_{t+1}) &\propto f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k), \quad k = 1, ..., M \\ &\simeq \overline{g}_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k), \end{aligned} \tag{6·1}$$

where we have constructed $\overline{g}_1(y_{t+1}|\alpha_{t+1})$, the global approximation to $f_1(y_{t+1}|\alpha_{t+1})$, to be conjugate to $f_2(\alpha_{t+1}|\alpha_t^k)$. Therefore our joint proposal becomes

$$g(\alpha_{t+1}, \alpha_t^k) \propto \overline{g}(y_{t+1}|\alpha_t^k) g(\alpha_{t+1}|\alpha_t^k, y_{t+1}).$$

We have a mass function $g(\alpha_t^k) = \overline{g}(y_{t+1}|\alpha_t^k) / \sum_{i=1}^M \overline{g}(y_{t+1}|\alpha_t^i)$. We can apply our smooth bootstrap yielding ordered samples, $\alpha_t^j,\ j = 1, ..., R$. This is very efficient as the α_t^k are sorted prior to the smooth bootstrap. We then propagate each of these through $g(\alpha_{t+1}|\alpha_t, y_{t+1})$, and sort, to give a sorted sample of $\alpha_{t+1}^j,\ j = 1, ..., R$. The smooth bootstrap is applied, with weights $\omega(\alpha_{t+1}^j)$, where

$$\omega(\alpha_{t+1}) = \frac{f_1(y_{t+1}|\alpha_{t+1})}{\overline{g}_1(y_{t+1}|\alpha_{t+1})}. \tag{6·2}$$

Note that in (6·1) we could have, if necessary, approximated $f_2(\alpha_{t+1}|\alpha_t^k)$ by $g_2(\alpha_{t+1}|\alpha_t^k)$ without changing the fundamental nature of the resulting algorithm. Following the results of Section 3, the likelihood $\widehat{f}(y_t|\mathcal{F}_{t-1}, \theta)$ is unbiassedly estimated as

$$\left[\frac{1}{M}\sum_{k=1}^{M} \overline{g}(y_t|\alpha_t^k)\right]\left[\frac{1}{R}\sum_{j=1}^{R} \omega_j\right]. \tag{6·3}$$

We shall illustrate the application of this algorithm for Gaussian transition densities in the underlying Markov chain.

6·1 *Gaussian transitions*

The reasonably general situation in which we have a Gaussian transition density shall now be considered. A comprehensive discussion of dynamic models of this kind may be found in West & Harrison (1997).

Let us suppose that $\alpha_{t+1}|\alpha_t \sim N\left\{\mu\left(\alpha_t\right), \sigma^2\left(\alpha_t\right)\right\}$ and that $f_1(y_{t+1}|\alpha_{t+1}) = \exp\{l(\alpha_{t+1}; y_{t+1})\}$. Then we form $\overline{g}_1(y_{t+1}|\alpha_{t+1})$ above as

$$\log \overline{g}_1(y_{t+1}|\alpha_{t+1}) = l'(\widehat{\alpha}_{t+1}; y_{t+1})(\alpha_{t+1} - \widehat{\alpha}_{t+1}) + \frac{1}{2} l''(\widehat{\alpha}_{t+1}; y_{t+1})(\alpha_{t+1} - \widehat{\alpha}_{t+1})^2$$

the last two terms in a second order expansion of $l(\alpha_{t+1}; y_{t+1})$ in α_{t+1} around $\widehat{\alpha}_{t+1}$[5]. We would

[5]If $l'' > 0$ we may replace this term by a negative constant or discard the second order term altogether.

generally choose $\widehat{\alpha}_{t+1}$ to be $\widehat{\alpha}_{t+1} = \sum_{k=1}^{M} \mu(\alpha_t^k)$, the forecast mean for α_{t+1}. We then obtain

$$\begin{aligned} \overline{g}(y_{t+1}|\alpha_t^k) &= \tfrac{\sigma^*}{\sigma(\alpha_t)} \exp\left(\tfrac{1}{2}\tfrac{\mu^{*2}}{\sigma^{*2}} - \tfrac{1}{2}\tfrac{\mu(\alpha_t)^2}{\sigma^2(\alpha_t)}\right), \\ g(\alpha_{t+1}|\alpha_t, y_{t+1}) &= \mathrm{N}\left(\alpha_{t+1}|\mu^*; \sigma^{*2}\right), \end{aligned}$$

where $\sigma^{*2} = \left(\frac{1}{\sigma^2(\alpha_t)} - l''\right)^{-1}$, $\mu^* = \sigma^{*2}\left(\frac{\mu(\alpha_t)}{\sigma^2(\alpha_t)} + l' - l'' \times \widehat{\alpha}_{t+1}\right)$.

This expansion technique is valuable as it leads to more efficient filtering and more efficient likelihood estimation. The efficiency of our procedure may be illustrated by examining the second stage weights $\omega(\alpha_{t+1})$. An efficient algorithm will have normalised weights which are as even as possible, see for instance Liu & Chen (1998). For the standard ASIR_0, we have that $\omega(\alpha_{t+1}) = f_1(y_{t+1}|\alpha_{t+1})$. We now have that

$$\log \omega(\alpha_{t+1}) = l(\alpha_{t+1}; y_{t+1}) - l'(\widehat{\alpha}_{t+1}; y_{t+1})(\alpha_{t+1} - \widehat{\alpha}_{t+1}) - \frac{1}{2} l''(\widehat{\alpha}_{t+1}; y_{t+1})(\alpha_{t+1} - \widehat{\alpha}_{t+1})^2.$$

As a function of α_{t+1}, this should not be very variable. The variability will be extremely small if $l(\alpha_{t+1}; y_{t+1})$ is close to a quadratic in α_{t+1}. For instance, suppose we have a conditionally Poisson model where the state follows a Gaussian autoregression and, conditionally, we have $y_t \sim \mathrm{Po}(\theta)$. As the counts, y_t, become larger then $l(\alpha_{t+1}; y_{t+1})$ approaches a quadratic function in α_{t+1}. Our weights will therefore be fairly even and this partial adaption approach will perform much more effectively than a standard SIR method. This result also holds for many conditionally exponential family observation models, such as the binomial (as $n \longrightarrow \infty$). A first order expansion may also be used leading to similar conjugate relationships.

6·1.1 Example: SV(1) model

We can examine the partial adaption method for Gaussian transitions by again considering the stochastic volatility (SV) model,

$$y_t = \varepsilon_t \exp(\alpha_t/2), \quad \alpha_{t+1} = \mu + \phi(\alpha_t - \mu) + \eta_t, \tag{6·4}$$

where ε_t and η_t are independent Gaussian processes with variances of 1 and σ^2 respectively. In this case, we have

$$l(\alpha_{t+1}) = -\frac{1}{2} y_{t+1}^2 \exp(-\alpha_{t+1}) - \frac{1}{2}\alpha_{t+1}.$$

We may apply the general Gaussian transition partial adaption approach, yielding second stage weights $\omega(\alpha_{t+1})$, where

$$\begin{aligned} \log \omega(\alpha_{t+1}) &= c - \frac{1}{2} y_{t+1}^2 \{\exp(-\alpha_{t+1}) + (\alpha_{t+1} - \widehat{\alpha}_{t+1}) \exp(-\widehat{\alpha}_{t+1}) \\ &\quad - \frac{1}{2}(\alpha_{t+1} - \widehat{\alpha}_{t+1})^2 \exp(-\widehat{\alpha}_{t+1})\}. \end{aligned}$$

These second stage weights will be much less variable than for the standard, unadapted, particle filter. The procedure is simple to apply and results in superior performance in estimation whilst still ensuring smoothness in the estimated likelihood. It is no more computationally demanding than the standard smooth SIR filter.

7 Adjustments for local adaption and integrated processes

So far we have dealt with the case where the second stage ω_{t+1} is only a function of the corresponding state so that $\omega_{t+1} = \omega_{t+1}(\alpha_{t+1})$. When ω_{t+1} is a smooth function of α_{t+1}, the methods of the previous section, which employ smooth resampling, are sufficient to ensure smoothness in the resulting likelihood. However, in a number of models, which we shall consider presently, the weights ω_{t+1} are a function not only of α_{t+1} but also of some additional information associated with the sample I_{t+1}, so that $\omega_{t+1} = \omega_{t+1}(\alpha_{t+1}, I_{t+1})$. Regarding our samples $\alpha_{t+1}^1, ..., \alpha_{t+1}^R$ as sorted in ascending order we have the associated (generally unordered) samples $I_{t+1}^1, ..., I_{t+1}^R$ and the weights $\omega_{t+1}^1, ..., \omega_{t+1}^R$. Even if two points α_{t+1}^i and α_{t+1}^{i+1} are arbitrarily close, the corresponding weights ω_{t+1}^i and ω_{t+1}^{i+1} will not be as they also depend upon I_{t+1}^i and I_{t+1}^{i+1}, which in general will be quite different. Let us assume that $\omega_{t+1}^i >> \omega_{t+1}^{i+1}$. As the parameters θ change slightly to become $\widetilde{\theta} = \theta + \varepsilon$ the order of the two points may change, in which case we have $\widetilde{\alpha}_{t+1}^i = \alpha_{t+1}^{i+1} + g_{i+1}(\varepsilon) < \widetilde{\alpha}_{t+1}^{i+1} = \alpha_{t+1}^i + g_i(\varepsilon)$. However, for the new associated weights we have $\widetilde{\omega}_{t+1}^{i+1} >> \widetilde{\omega}_{t+1}^i$. The resulting smooth cdf we constructed will have changed significantly, despite the arbitrarily small change in the parameters and the states. This results in a likelihood that is not estimated smoothly.

The problem is not necessarily particularly dramatic in applications as the change to our smooth cdf is only a very local one, particularly when R is large. Also the variability of the weights ω_{t+1} may be small which again minimises this difficulty. However, for numerical maximisation schemes this may cause difficulties and it is necessary to ensure smoothness. We only have to ensure that very small changes in the order of the states leads to small changes in the weights ω_{t+1}. To do this we use a kernel approach to smooth our weights, and therefore the probabilities π^*, prior to sampling via

$$\omega_j^* = \frac{\sum_{i=1}^R \omega_i \phi^*\left((x_j - x_i)/h\right)}{\sum_{i=1}^R \phi^*\left((x_j - x_i)/h\right)}, \qquad \pi_j^* = \frac{\omega_j^*}{\sum_{i=1}^R \omega_i^*}, \quad j = 1, ..., R,$$

where ϕ^* is the density for a standard Gaussian random variable, and $x_i = \alpha_{t+1}^i$ for notational convenience. In practise, for computational efficiency, we truncate this at values where $|x_j - x_i| > 3h$ allowing very fast computation. We make h very small as this is sufficient to ensure continuity

and makes the computations very close to linear in R. We make h a function of R[6], so that $h = cR^{-\frac{1}{2}}$, where c is a very small number. When R is very large, h will be very small but the smoothing will still take place over approximately the same number of points. The efficient algorithm is given in the Appendix. In the following applications, we found a negligible overhead from using this presmoothing technique on the weights.

7·1 *Continuous time models*

We shall now consider estimation of partially observed stochastic differential equations (SDEs). We focus on the specific example of a stochastic volatility model. Let us allow the log of a price, $y(t)$, to evolve as

$$\begin{aligned} \mathrm{d}y(t) &= \{\mu + \beta\sigma^2(t)\}\mathrm{d}t + \sigma(t)\mathrm{d}W_1(t) \\ \mathrm{d}\sigma^2(t) &= a(\sigma^2(t))\mathrm{d}t + b(\sigma^2(t))\mathrm{d}W_2(t). \end{aligned} \tag{7·1}$$

A discussion of these models is provided in Campbell et al. (1997, Chapter 9). The models may be used to proce options in the presence of stochastic volatility, see for instance Heston (1993). The β parameter represents a risk premium term. We would expect this to be positive as higher volatility should lead to higher expected returns. Reviews of special cases of this model are given in Taylor (1994), Ghysels et al. (1996) and Shephard (1996). Note that we may also have a linear term in $y(t)$ in the drift and that we can have non-linear functions of $\sigma^2(t)$ in both the drift and volatility terms without affecting our analysis. Let us suppose that we observe the log price at times $\tau_1 < \tau_2 < \tau_2 < ... < \tau_n < \tau_{n+1}$. We therefore have n returns $r_s = y(\tau_{s+1}) - y(\tau_s)$, for $s = 1, ..., n$. Note that we now have,

$$r_s \sim \mathrm{N}(\mu\Delta_s + \beta\sigma_s^{2*}; \sigma_s^{2*}),$$

regardless of the process for $\sigma^2(t)$, where

$$\Delta_s = \tau_{s+1} - \tau_s, \quad \sigma_s^{2*} = \int_{\tau_s}^{\tau_{s+1}} \sigma^2(v)\ \mathrm{d}v.$$

We refer to $\sigma^2(t)$ as the instantaneous volatility and σ_s^{2*} as the integrated volatility, for reasons which are clear. The important aspect to notice from the point of view of our particle filter is that we no longer have simple dependence on the corresponding state. Rather it is the integral of the path of the state between two points which is crucial.

We shall now use a very fine Euler approximation to the process. We shall place $M_s - 1$ latent points between our instantaneous volatilities $\sigma^2(\tau_s)$ and $\sigma^2(\tau_{s+1})$. We shall define our points as $\sigma_{s,1}^2, ..., \sigma_{s,M_s-1}^2$ where for notational convenience we will set $\sigma_{s,0}^2 = \sigma^2(\tau_s)$ and $\sigma_{s,M_s}^2 = \sigma^2(\tau_{s+1})$.

[6] As R becomes larger, the original smooth cdf is less affected by changes in the parameters.

These latent points are evenly spaced in time by $\delta = \Delta_s/M_s$. We now have the Euler evolution, $\sigma^2_{s,0} = \sigma^2_{s-1,M_s}$,

$$\sigma^2_{s,m+1} = \sigma^2_{s,m} + a(\sigma^2_{s,m})\delta + b(\sigma^2_{s,m})\sqrt{\delta}u_m, \quad m = 0, ..., M_s - 1. \tag{7·2}$$

where $u_m \sim \text{NID}(0, 1)$. We have a Markov chain in $\sigma^2_{s,m}$, and having approximated our volatility evolution, we may write

$$r_s \sim \text{N}(\mu\Delta_s + \beta\widehat{\sigma}^{2*}_s; \widehat{\sigma}^{2*}_s), \tag{7·3}$$

where we now have,

$$\widehat{\sigma}^{2*}_s = \delta \sum_{m=0}^{M_s-1} \sigma^2_{s,m}. \tag{7·4}$$

Note that as $\delta \to 0$, we have that $\widehat{\sigma}^{2*}_s \to \sigma^{2*}_s$. Note that (7·3) can also be derived from the aggregation of the Euler discretisation of the measurement equation of (7·1),

$$y_{s,m+1} \quad = \quad y_{s,m} + \{\mu + \beta\sigma^2_{s,m}\}\delta + \sigma_{s,m}\sqrt{\delta}\eta_m, \quad m = 0, ..., M_s - 1,$$

where $y_{s,0} = y(\tau_s)$ and $y_{s,M_s} = y(\tau_{s+1})$ and $\eta_m \sim \text{NID}(0, 1)$.

We note that the particle filter approach is straightforward. Our state at time τ_s is $\sigma^2(\tau_s)$ and we can simulate from the transition $f(\sigma^2(\tau_{s+1}) \mid \sigma^2(\tau_s))$ by using (7·2). Suppose at time τ_s we have our sample $\sigma^{2,1}(\tau_s), ..., \sigma^{2,M}(\tau_s)$ from $f(\sigma^2(\tau_s) \mid r_1, ..., r_{s-1})$. Then we propagate from our Euler transition density R times, as usual in a stratified manner, to get our new states $\sigma^{2,1}(\tau_{s+1}), ..., \sigma^{2,R}(\tau_{s+1})$ from $f(\sigma^2(\tau_{s+1}) \mid r_1, ..., r_{s-1})$ and record the associated estimated integrated volatilities $\widehat{\sigma}^{2*,1}_s, ..., \widehat{\sigma}^{2*,R}_s$. We then attach to the sorted sample the weights and probabilities,

$$\omega_j = f(r_s \mid \widehat{\sigma}^{2*,j}_s), \qquad \pi_j = \frac{\omega_j}{\sum_{i=1}^R \omega_i}, \quad j = 1, ..., R,$$

where $f(r_s \mid \widehat{\sigma}^{2*}_s)$ is given by (7·3). We use the smoothing device on these weights and then sample using our smooth bootstrap method. It is worth noting that the aggregation result allowing us to directly calculate $f(r_s \mid \widehat{\sigma}^{2*}_s)$ is particularly helpful in this application, making the particle filter relatively simple to implement.

The above analysis assumed that the terms $\text{d}W_1(t)$ and $\text{d}W_2(t)$ were independent. We note that it is straightforward to consider correlated disturbances. If we have

$$\begin{pmatrix} \text{d}W_1(t) \\ \text{d}W_2(t) \end{pmatrix} \sim \text{N}\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} \text{d}t & \rho\text{d}t \\ \rho\text{d}t & \text{d}t \end{pmatrix} \right\},$$

then we obtain,

$$\begin{pmatrix} \eta_m \\ u_m \end{pmatrix} \sim \text{N}\left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix}; \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix} \right\}.$$

so that $\eta_m|u_m \sim \mathrm{N}(\rho u_m; 1-\rho^2)$. Our Euler measurement equation again aggregates to become

$$\begin{aligned} y_{s,M} &= y_{s,0} + \mu\Delta_s + \beta\widehat{\sigma}_s^{2*} + \sqrt{\delta}\sum_{m=0}^{M_s-1}\sigma_{s,m}\eta_m \\ &= y_{s,0} + \mu\Delta_s + \beta\widehat{\sigma}_s^{2*} + \sqrt{\delta}\sum_{m=0}^{M_s-1}\sigma_{s,m}(\rho u_m + \sqrt{1-\rho^2}\xi_m). \end{aligned}$$

So we obtain,

$$\begin{aligned} r_s &= y_{s,M} - y_{s,0} \\ &\sim \mathrm{N}\left\{ y_{s,0} + \mu\Delta_s + \beta\widehat{\sigma}_s^{2*} + \rho\sqrt{\delta}\sum_{m=0}^{M_s-1}\sigma_{s,m}u_m; \delta(1-\rho^2)\sum_{m=0}^{M_s-1}\sigma_{s,m}^2 \right\}, \end{aligned}$$

noting of course that the shock to the volatility term,

$$u_m = \frac{(\sigma_{s,m+1}^2 - \sigma_{s,m}^2 - a(\sigma_{s,m}^2)\delta)}{b(\sigma_{s,m}^2)\sqrt{\delta}}.$$

So given our volatility path $\sigma_{s,0}^2, \sigma_{s,1}^2, ..., \sigma_{s,M_s}^2$, the measurement density is easily calculated. The correlation is important as it captures the leverage hypothesis which suggests that innovations to volatility are negatively correlated to shocks in returns.

The reweighting only has to be carried out when actual returns are observed, not at the frequency of our discretisation. Simulating forward from an Euler approximation is clearly fast and straightforward. It is also easy to keep the same random number stream as we are simply using Gaussian variates.

7·1.1 Example 2: Nelson process

As an illustration, we shall consider the following model,

$$\begin{aligned} \mathrm{d}y(t) &= \sigma(t)\mathrm{d}W_1(t) \\ \mathrm{d}\sigma^2(t) &= k(\theta - \sigma^2)\mathrm{d}t + \sqrt{\xi}\sigma^2\mathrm{d}W_2(t), \end{aligned} \tag{7·5}$$

where $\mathrm{d}W_1(t)$, $\mathrm{d}W_2(t)$ are considered independent.

This is essentially the model of Hull & White (1987), which they use to price options by assuming volatility is uncorrelated with aggregate consumption. It is briefly discussed in Campbell et al. (1997, Chapter 9). The volatility equation is the limit of a standard GARCH(1,1) as the frequency of observation tends to infinity, see Nelson (1990). The stationary distribution for the volatility is inverse gamma, $\mathrm{IGa}(\upsilon, \beta)$ where $\upsilon = 1 + 2k/\xi$ and $\beta = 2k\theta/\xi$, as shown by Nelson (1990).

Regarding the returns as being observed at unit frequency, we carry out the simulation of the time series with $\delta = 0.01$. We fix the parameters for the simulation, of length $n = 4000$,

at $\Psi = (k, \theta, \xi)' = (0.02, 0.5, 0.0178)'$. These values reflect the marginal distribution of the volatility (mean is 0.5, variance 0.2) and the persistence for a unit of one day in the returns. In our likelihood analysis we fix $\delta = 0.05$ and take $M = 1000$, $R = 1300$. In order to examine the performance of our sampler we examine the profile of the estimated likelihood on a grid of points in the parameter space. We carry out three different runs (using different random number seeds for each) of the smooth particle filter on each parameter in Ψ and plot a profile of the estimated likelihood function, keeping the remaining two parameters fixed at their true values. Due to the fairly large sample, we would hope to be in the vicinity of the true values. The results are displayed in Figure 8 which plots the likelihood function and its log. There is very little variability in the three estimates of the likelihood which are entirely smooth, even when viewed more locally. In addition, the estimated likelihoods are centred around the true values quite tightly indicating that the method is working well.

The advantage of this type of approach is that any SDE in σ^2 can be considered. Heston (1993) considers a Feller process in σ^2, in which the volatility term in σ^2 is replaced by $b(\sigma^2) = \sqrt{\xi}\sigma$. Other authors assume that $\log(\sigma^2)$ follows an Ornstein-Uhlenbeck process so that $b(\sigma^2) = \sqrt{\xi}$. Jumps, see for instance Eraker et al. (2003), may also be included in the return and volatility processes without many additional complications.

7·2 *The ASIR method*

At the beginning of this paper, Section 2, general auxiliary particle filter methods of PS were outlined which exploit the local adaption. That is, information about the mixture component, α_t^k, is used to form the proposal within the sampling importance resampling procedure. We have,

$$\begin{aligned} f(\alpha_{t+1}, k \mid \mathcal{F}_{t+1}) &\propto f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k), \quad k = 1, ..., M \\ &\simeq \overline{g}(k, \alpha_{t+1}) = \overline{g}_1(y_{t+1}|\alpha_{t+1}, k) g_2(\alpha_{t+1}|\alpha_t^k) \\ &= \overline{g}(y_{t+1}|k) g(\alpha_{t+1}|k, y_{t+1}) = C.g(k, \alpha_{t+1}). \end{aligned} \tag{7·6}$$

We can proceed in the usual way by sampling from the joint proposal density below

$$g(k) = \frac{\overline{g}(y_{t+1}|k)}{\sum_{i=1}^{M} \overline{g}(y_{t+1}|i)}, \qquad g(\alpha_{t+1}|k) = g(\alpha_{t+1}|k, y_{t+1}),$$

ensuring that the α_t^k which are sampled are chosen in ascending order. This is a linear operation as the α_t^k are sorted prior to sampling. We sort the samples α_{t+1}^j, k^j by α_{t+1}^j where $j = 1, .., R$,and associate with each of the samples the weights

$$\omega_j = \omega(\alpha_{t+1}^j, k^j), \qquad \pi_j = \frac{\omega_j}{\sum_{i=1}^{R} \omega_i}.$$

where

$$\omega(\alpha_{t+1}, k) = \frac{f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k)}{\overline{g}_1(y_{t+1}|\alpha_{t+1}, k) g_2(\alpha_{t+1}|\alpha_t^k)}.$$

These weights will not necessarily be smooth as a function of α_{t+1} so we presmooth using the kernel smoothing method described previously associated smoothed weights ω_j^*, and the corresponding normalised probabilities π_j^*, with each sample α_{t+1}^j. We now apply the smooth bootstrap procedure which yields the sorted sample $\alpha_{t+1}^i$where $j = 1, .., M$. In most applications the weights ω_j, by construction, will be close to constant across the samples j.

8 Conclusions

In this paper we have attempted to tackle two related issues. Firstly, we have provided an effective and simple method for efficiently estimating the prediction density $f(y_{t+1}|Y_t)$ from the standard output of the auxiliary particle filter. Secondly, we have shown how the prediction density and therefore the likelihood can be estimated so that the estimator is continuous as a function of the parameters. The ASIR_0 technique is fast and allows any models considered in this paper to be estimated. The computational implementation is reasonably simple. In particular, we only need to be able to simulate from the transition density, a considerable advantage when focusing on, for instance, models expressed in continuous time. Applications to models in continuous time are highlighted in Section 4·2, in which an stochastic differential equation is observed with error, and in Section 7·1 where we observe an integrated process with error. The methods are shown to be robust and accurate even for large time series.

The advantages over importance sampling are evident. Firstly, the consideration of a proposal density over the whole state space is not necessary. This is fortunate as many of the importance sampling schemes, see for instance Durbin & Koopman (1997), rely on a Gaussian proposal. Whilst Gaussian proposals may be quite effective when applied to cases in which $f(\alpha|Y)$ is close to Gaussian, they are going to be ineffective when applied to models which depart dramatically from being linear Gaussian. Secondly, importance samplers lead to much more variable estimation as more data are included. This can create serious problems in time series of a reasonable length. Particle filters to some extent circumvent this problem with the variance of estimators being linear in the observation size.

An advantage of using particle filters to estimate non-Gaussian state space models is that, as a by-product of the estimation scheme we also obtain the filtered path of the states and can perform diagnostics straightforwardly, see Pitt & Shephard (1999).

9 Acknowledgements

I am grateful to Robert Kohn, Neil Shephard and Hans Kunsch for comments and suggestions regarding earlier versions of this paper.

10 Appendix

10·1 *Multinomial sampling*

The following algorithm is discussed in Carpenter et al. (1997). Suppose we have $x_1, ..., x_R$ with probability of $\pi_1, ..., \pi_R$. Then the task will be to draw a sample of size M from this discrete distribution in $O(M)$ computations. We carry this out by sampling an ordered version of these variables, so that we obtain $y_1 \leq y_2 \leq ... \leq y_{M-1} \leq y_M$ as our sample, assuming that our x values are sorted by ascending order[7]. Drawing order variables will be carried out by first drawing order uniforms (see, for example, Ripley (1987, p. 96)). Let $u_1, ..., u_M \sim$UID$(0, 1)$, then

$$u_{(M)} = u_M^{1/M}, \quad u_{(k)} = u_{(k+1)} u_k^{1/k}, \quad k = M-1, M-3, ..., 2, 1,$$

where $u_{(1)} < u_{(2)} < ... < u_{(M-1)} < u_{(M)}$. This is most easily carried out in logarithms. Then we obtain the ordered y using the following trivial algorithm,

Algorithm 1:

```
set s=0,j=1;
for (i=1 to R)
{
    s=s+π_i;
    while (u_(j) ≤ s   AND   j ≤ M)
    {
        y_j = x_i;
          j=j+1;
    }
}
```

We generally use stratification, as discussed in Pitt & Shephard (2000). For the application in this paper we generate the M sorted uniforms as $u_{(j)} = (j-1)/M + u/M$, where the single random variate is $u \sim$UID$(0, 1)$. Algorithm 1 is then implemented.

[7]When we have sorted univariate x values, we may interpret this algorithm as simply inverting our cumulative distribution function. The algorithm is still valid if the x values are not sorted.

10·1.1 Smooth resampling

Suppose we have our x in ascending order, $x_1 \leq x_2 \leq ... \leq x_{R-1} \leq x_R$, again with associated probabilities $\pi_1, ..., \pi_R$. As seen in Section 3·1 we define a region i, S_i as follows: $S_i = [x^i, x^{i+1}]$, $i = 1, ..., R-1$. These regions form a partition of the sample space for x. We have different conditional densities $g(x|i)$, given in Section 3·1 conditional upon each region i, S_i. We shall assign $\Pr(i) = \widetilde{\pi}_i = \frac{1}{2}(\pi_i + \pi_{i+1})$, $i = 2, ..., R-2$ and $\Pr(1) = \widetilde{\pi}_1 = \frac{1}{2}(2\pi_1 + \pi_2)$, $\Pr(R-1) = \widetilde{\pi}_{R-1} = \frac{1}{2}(\pi_{R-1} + 2\pi_R)$. Here we have two algorithms which together efficiently implement the smooth sampling, see Section 3·1 and Figure 1. This samples, smoothly, M times from the empirical cdf. This produces an ordered sample $y_1 \leq y_2 \leq ... \leq y_{M-1} \leq y_M$. Again, as above, we generate the M sorted uniforms as $u_{(j)} = (j-1)/M + u/M$ where $j = 1, ..., M$. The first algorithm, algorithm 2, samples the index (corresponding to the region) storing these as $r_1, r_2, ..., r_M$ and storing $u_1^*, u_2^*, ..., u_M^*$.

Algorithm 2:

```
set s=0,j=1;
for (i=1 to R-1)
{
   s=s+π̃_i;
   while (u_(j) ≤ s   AND   j ≤ M)
  {
       r_j = i;
      u*_j=(u_(j)  - (s - π̃_i))/π̃_i;
       j=j+1;
   }
}
```

Having obtained the region we are in r_j, $j = 1, ..., M$, we then sample conditional upon that region from $g(x|r_j)$, given in Section 3·1, using the corresponding uniform u_j^*. We have region r_j, $S_{r_j} = [x^{r_j}, x^{r_j+1}]$. If $r_j = 2, ..., R-2$ then we set $y_j = (x^{r_j+1} - x^{r_j}) \times u_j^* + x^{r_j}$ since $g(x|r_j)$ is simply a uniform distribution. Similarly we can directly simply invert the cdf, given in Section 3·1, for $g(x|r_j = 1)$ or for $g(x|r_j = R-1)$.

10·1.2 Presmoothing algorithm

Suppose we have our ordered sample $(x_1, ..., x_R)'$ with associated weights $(\omega_1, ..., \omega_R)'$. Then we wish to calculate the smooth weights,

$$\omega_j^* = \frac{\sum_{i=1}^R \omega_i \phi^*\left((x_j - x_i)/h\right)}{\sum_{i=1}^R \phi^*\left((x_j - x_i)/h\right)}, \qquad \pi_j^* = \frac{\omega_j^*}{\sum_{i=1}^R \omega_i^*},$$

for $j = 1, ..., R$ where π_j^* represent the corresponding probabilities. We have initially set $h = cR^{-\frac{1}{2}}$, this ensures the efficiency of our method together with the trucation choice below:

set min = 0;

for $j = 1$ to R

{

set sum1 = 0, sum2 = 0;

set max = R;

for $i =$`min` to `max`

{

`sum1 = sum1 +` $\omega_i \phi((x_j - x_i)/h)$;

`sum2 = sum2 +` $\phi((x_j - x_i)/h)$;

`if (`$|x_j - x_i| < -3h$`) set min = j;`

`if (`$|x_j - x_i| > 3h$`) set max = j;`

}

set $\omega_j^* =$`sum1/sum2;`

}

Typically we choose $c = 0.6\sqrt{var(x)}$, so that for example when $var(x) = 1$, $R = 100$, we have $h = 0.06$.

10·2 **Proof of Theorem 1**

Let $I = \widehat{f}(y_{t+1}|\mathcal{F}_t)$, then from (2·5) we have

$$\begin{aligned} I &= \int f_1(y_{t+1}|\alpha_{t+1}) \left\{ \sum_{k=1}^M f_2(\alpha_{t+1}|\alpha_t^k) \frac{1}{M} \right\} d\alpha_{t+1} \\ &= \sum_{k=1}^M \int f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k) \frac{1}{M} d\alpha_{t+1} \\ &= \sum_{k=1}^M \int \frac{f_1(y_{t+1}|\alpha_{t+1}) f_2(\alpha_{t+1}|\alpha_t^k)}{\overline{g}(k, \alpha_{t+1})} \frac{1}{M} \overline{g}(k, \alpha_{t+1}) d\alpha_t. \end{aligned}$$

Using the expressions for C and for $\omega(\alpha_{t+1}, k)$ from Section 2, we obtain

$$\begin{aligned} I &= \sum_{k=1}^M \int \omega(\alpha_{t+1}, k) \frac{1}{M} C.g(k, \alpha_{t+1}) d\alpha_{t+1} \\ &= \left\{ \frac{1}{M} \sum_{i=1}^M \overline{g}(y_{t+1}|i) \right\} \left\{ \sum_{k=1}^M \int \omega(\alpha_{t+1}, k) g(k, \alpha_{t+1}) d\alpha_{t+1} \right\} \end{aligned}$$

$$= \left\{ \frac{1}{M} \sum_{i=1}^{M} \overline{g}(y_{t+1}|i) \right\} E[\omega(\alpha_{t+1}, k)],$$

where the expectation is with respect to $g(k, \alpha_{t+1})$ as required.

Bias Correction Note that at present the log-likelihood will not be unbiassed. To correct this to first order we use the usual Taylor expansion method. Abstracting from likelihoods we have the large sample result that our estimated likelihood, $\overline{X}$ is unbiassed for the true likelihood, μ with we obtain $E[\overline{X}] = \mu$, $Var[\overline{X}] = \frac{\sigma^2}{R}$ We therefore have

$$E[\log \overline{X}] \simeq \log \mu - \frac{1}{2} \frac{\sigma^2}{R\mu^2},$$

an approximation which is very good for large R. Therefore we may bias correct by substituting μ as $\overline{X}$, setting

$$\widehat{\log \mu} = \log \overline{X} + \frac{1}{2} \frac{\widehat{\sigma}^2}{R\overline{X}^2}.$$

References

Andrieu, C. & Doucet, A. (2002). Particle filtering for partially observed gaussian state space models. *JRSSB* forthcoming.

Berzuini, C., Best, N. G., Gilks, W. R., & Larizza, C. (1997). Dynamic conditional independence models and Markov chain Monte Carlo methods. *J. Am. Statist. Assoc.* **92**, 140312.

Campbell, J. Y., Lo, A. W., & MacKinlay, A. C. (1997). *The Econometrics of Financial Markets*. Princeton University Press, Princeton, New Jersey.

Carpenter, J. R., Clifford, P., & Fearnhead, P. (1997). Efficient implementation of particle filters for non-linear systems. 4th Interim Report, DRA contract WSS/U1172, Department of Statistics, Oxford University.

Carpenter, J. R., Clifford, P., & Fearnhead, P. (1999). An improved particle filter for non-linear problems. *IEE Proceedings on Radar, Sonar and Navigation* **146**, 2–7.

Danielsson, J. & Richard, J. F. (1993). Accelerated Gaussian importance sampler with application to dynamic latent variable models. *J. Appl. Econometrics* **8**, S153–S174. Reprinted as pp. 169–190 in van Dijk, H.K., A. Monfort and B.W. Brown(1995), *Econometric Inference Using Simulation Techniques*, Chichester: Wiley.

Diebold, F. X. & Nerlove, M. (1989). The dynamics of exchange rate volatility: a multivariate latent factor ARCH model. *J. Appl. Econometrics* **4**, 1–21.

Doucet, A., de Freitas, J. F. G., & Gordon, N. (2000a). *Sequential Monte Carlo Methods in Practice*. Cambridge University Press, Cambridge.

Doucet, A., Godsill, S., & Andrieu, C. (2000b). On sequential simulation-based methods for Bayesian filtering. *Statistics and Computing* **10**, 197–208.

Durbin, J. & Koopman, S. J. (1997). Monte Carlo maximum likelihood estimation of non-Gaussian state space model. *Biometrika* **84**, 669–84.

Efron, B. & Tibshirani, R. J. (1993). *An Introduction to the Bootstrap*. Chapman & Hall, New York.

Engle, R. F. (1995). *ARCH: Selected Readings*. Oxford University Press, Oxford.

Eraker, B., Johannes, M., & Polson, N. (2003). The impact of jumps in volatility and returns. *J. Finance* Forthcoming.

Gerlach, R., Carter, C., & Kohn, R. (1999). Diagnostics for time series analysis. *J. Time Series Analysis* **21**. Forthcoming.

Ghysels, E., Harvey, A. C., & Renault, E. (1996). Stochastic volatility. In Rao, C. R. & Maddala, G. S., editors, *Statistical Methods in Finance*, pages 119–191. North-Holland,

Amsterdam.

Gordon, N. J., Salmond, D. J., & Smith, A. F. M. (1993). A novel approach to non-linear and non-Gaussian Bayesian state estimation. *IEE-Proceedings F* **140**, 107–33.

Harvey, A. C. (1993). *Time Series Models*. Harvester Wheatsheaf, Hemel Hempstead, 2nd edition.

Harvey, A. C., Ruiz, E., & Sentana, E. (1992). Unobserved component time series models with ARCH disturbances. *J. Econometrics* **52**, 129–158.

Harvey, A. C., Ruiz, E., & Shephard, N. (1994). Multivariate stochastic variance models. *Rev. Economic Studies* **61**, 247–64.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility, with applications to bond and currency options. *Rev. Financial Studies* **6**, 327–343.

Hull, J. & White, A. (1987). The pricing of options on assets with stochastic volatilities. *J. Finance* **42**, 281–300.

Hurzeler, M. & Kunsch, H. (1998). Monte Carlo approximations for general state space models. *Journal of Computational and Graphical Statistics* **7**, 175–193.

Isard, M. & Blake, A. (1996). Contour tracking by stochastic propagation of conditional density. *Proceedings of the European Conference on Computer Vision, Cambridge* **1**, 343–356.

Jacquier, E., Polson, N. G., & Rossi, P. E. (1994). Bayesian analysis of stochastic volatility models (with discussion). *J. Business and Economic Statist.* **12**, 371–417.

Kim, S., Shephard, N., & Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. *Rev. Economic Studies* **65**, 361–93.

King, M., Sentana, E., & Wadhwani, S. (1994). Volatility and links between national stock markets. *Econometrica* **62**, 901–933.

Kitagawa, G. (1987). Non-Gaussian state space modelling of non-stationary time series. *J. Am. Statist. Assoc.* **82**, 503–514.

Kitagawa, G. (1996). Monte Carlo filter and smoother for non-Gaussian nonlinear state space models. *J. Computational and Graphical Statistics* **5**, 1–25.

Künsch, H. R. (2000). State space and hidden markov models. In Klüppelberg, C., editor, *Complex Stochastic Systems*. Chapman and Hall. Forthcoming.

Liu, J. & Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems. *J. American Statistical Association* **93**, 1032–1044.

Liu, J. & West, M. (2000). Combined parameter and state estimation in simulation-based filtering. In Doucet, A., de Freitas, J. F. G., & Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*. Cambridge University Press.

Nelson, D. B. (1990). ARCH models as diffusion approximations. *J. Econometrics* **45**, 7–38. Reprinted as pp. 176–192 in Engle, R.F.(1995), *ARCH: Selected Readings*, Oxford: Oxford University Press.

Pitt, M. & Shephard, N. (2000). Auxiliary variable based particle filters. In Doucet, A., de Freitas, J. F. G., & Gordon, N., editors, *Sequential Monte Carlo Methods in Practice*. Cambridge University Press.

Pitt, M. K. & Shephard, N. (1999). Filtering via simulation based on auxiliary particle filters. *J. American Statistical Association* **94**, 590–599.

Ripley, B. D. (1987). *Stochastic Simulation*. Wiley, New York.

Rubin, D. B. (1988). Using the SIR algorithm to simulate posterior distributions. In Bernardo, J. M., DeGroot, M. H., Lindley, D. V., & Smith, A. F. M., editors, *Bayesian Statistics 3*, pages 395–402. Oxford University Press Press, Oxford.

Shephard, N. (1996). Statistical aspects of ARCH and stochastic volatility. In Cox, D. R., Hinkley, D. V., & Barndorff-Nielson, O. E., editors, *Time Series Models in Econometrics, Finance and Other Fields*, pages 1–67. Chapman & Hall, London.

Shephard, N. & Pitt, M. K. (1997). Likelihood analysis of non-Gaussian measurement time series. *Biometrika* **84**, 653–67.

Silverman, B. W. (1986). *Density Estimation for Statistical and Data Analysis*. Chapman & Hall, London.

Smith, A. F. M. & Gelfand, A. E. (1992). Bayesian statistics without tears: a sampling-resampling perspective. *American Statistican* **46**, 84–88.

Taylor, S. J. (1994). Modelling stochastic volatility. *Mathematical Finance* **4**, 183–204.

West, M. (1992). Mixture models, Monte Carlo, Bayesian updating and dynamic models. *Computer Science and Statistics* **24**, 325–333.

West, M. & Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. Springer-Verlag, New York, 2 edition.

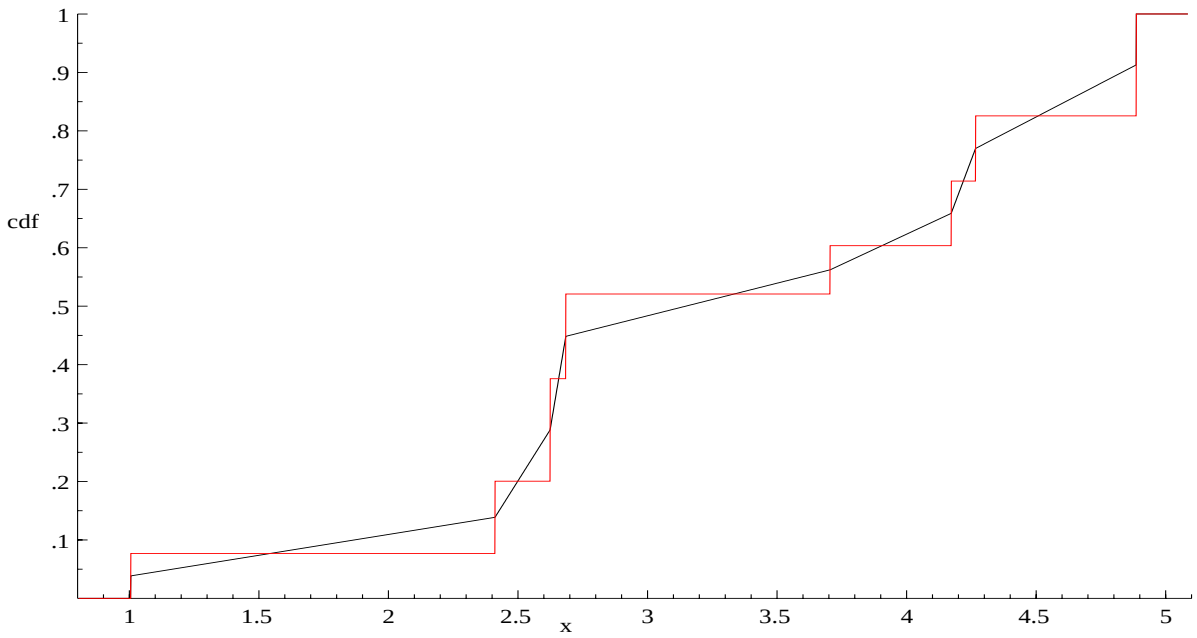

Figure 1: *Plot of the distribution function associated with the bootstrap (step function solid line) and the linear interpolation approach (dotted line). Here the number of points to be resampled is 8.*

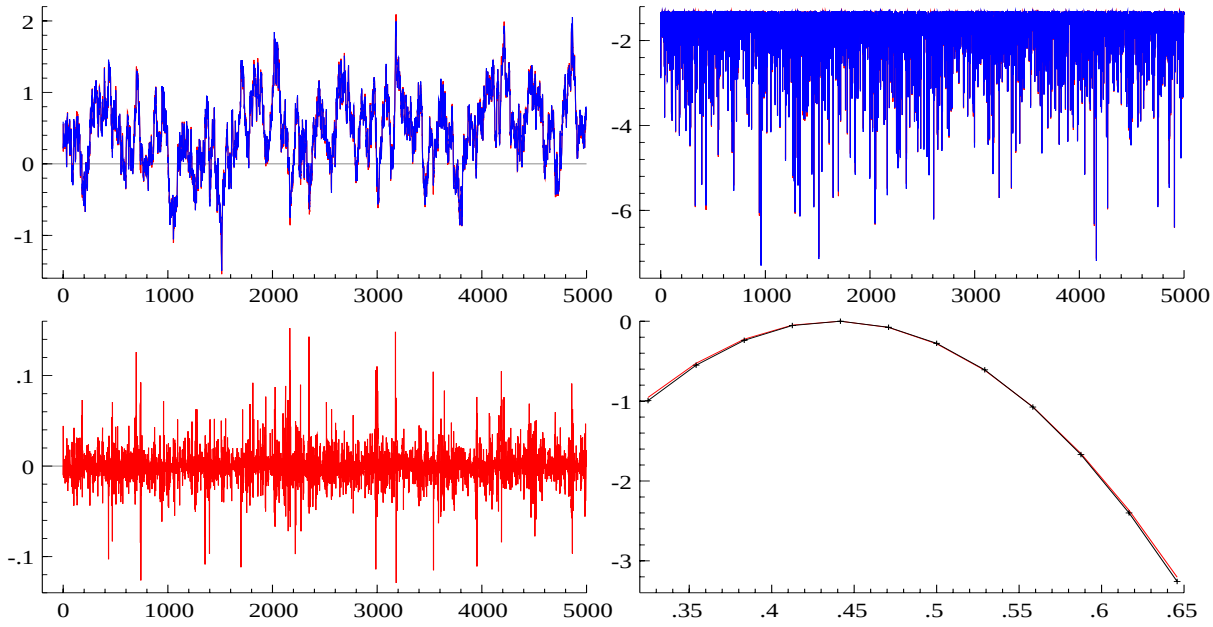

Figure 2: *TOP: left is sample filter mean and true KF mean for AR(1) plus noise example, right is the true log-likelihood component together with estimated component. BOTTOM: left is the error in the estimate of the log-likelihood component, right is the Kalman filter log-likelihood slice for* μ (solid line), together with its estimate both plotted against μ (dashed line). $T = 5000$, $M = 300$, $R = 400$. *True parameters in simulation* $\phi = 0.975$, $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$, $\sigma_\varepsilon = \sqrt{2}$. Smooth *particle filter with stratification used.*

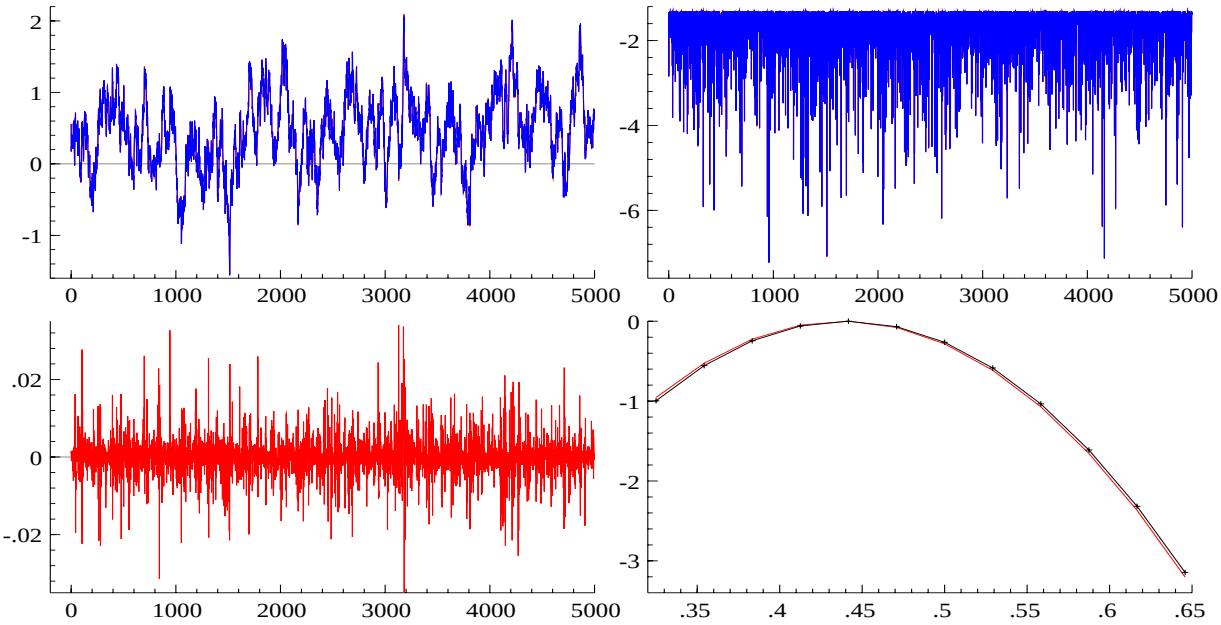

Figure 3: *TOP: left is sample filter mean and true KF mean for AR(1) plus noise example, right is the true log-likelihood component together with estimated component. BOTTOM: left is the error in the estimate of the log-likelihood component, right is the Kalman filter log-likelihood slice for* μ (solid line), together with its estimate both plotted against μ (dashed line). $T = 5000$, $M = 3500$, $R = 5000$. *True parameters in simulation* $\phi = 0.975$, $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$, $\sigma_\varepsilon = \sqrt{2}$. Smooth *particle filter with stratification used.*

Kalman Filter			
$ML\,(\sigma_\eta,\,\mu,\,\phi)'$	$Var\,(\sigma_\eta,\,\mu,\,\phi)' \times 10^3$		
0.07543	3.814	*−0.089*	*−0.722*
0.58276	.	40.31	*0.140*
0.96626	.	.	2.050

SIR Particle Filter; $M = 300$, $R = 400$, $SIM = 50$.

$\overline{ML_S}$	$\overline{Var} \times 10^3$		$MSE(ML_S) \times 10^4$
0.07574	3.155	*−0.067*	*−0.688*	0.182
0.58170	.	35.02	*0.112*	2.217
0.96610	.	.	1.846	0.103

SIR Particle Filter; $M = 1000$, $R = 1300$, $SIM = 50$.

$\overline{ML_S}$	$\overline{Var} \times 10^3$		$MSE(ML_S) \times 10^4$
0.07502	3.524	*−0.083*	*−0.708*	0.0628
0.58360	.	35.75	*0.135*	0.5295
0.96623	.	.	1.950	0.0450

SIR Particle Filter; $M = 3000$, $R = 4000$, $SIM = 50$.

$\overline{ML_S}$	$\overline{Var} \times 10^3$		$MSE(ML_S) \times 10^4$
0.07520	3.693	*−0.076*	*−0.715*	0.0190
0.58177	.	37.00	*0.124*	0.1495
0.96629	.	.	1.988	0.0123

Table 2: *Performance of standard smooth SIR particle filter for* $T = 150$. The model is AR(1) + noise with true parameters $\phi = 0.975$, $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$. *Additionally the measurement noise is fixed at* $\sigma_\varepsilon = \sqrt{2}$. *The off diagonal elements of the variance matrix are are reported as correlations not covariances covariances.*

Kalman Filter

$ML\,(\sigma_\eta,\,\mu,\,\phi)'$	$Var\,(\sigma_\eta,\,\mu,\,\phi)' \times 10^4$		
0.08080	7.670	*0.019*	*−0.698*
0.45900	.	429.1	*−0.036*
0.98398	.	.	1.223

SIR Particle Filter; $M = 300$, $R = 400$, $SIM = 50$.

$\overline{ML_S}$	$\overline{Var} \times 10^4$		$MSE(ML_S) \times 10^4$
0.08077	7.702	*−0.003*	*−0.697*	0.1347
0.45380	.	450.8	*−0.035*	4.666
0.98395	.	.	1.241	0.02287

SIR Particle Filter; $M = 1000$, $R = 1300$, $SIM = 50$.

$\overline{ML}_S$	$\overline{Var} \times 10^4$		$MSE(ML_S) \times 10^4$
0.08075	7.754	*0.006*	*−0.699*	0.0307
0.45965	.	437.3	*−0.020*	1.02
0.98398	.	.	1.224	0.00439

SIR Particle Filter; $M = 3000$, $R = 4000$, $SIM = 50$.

$\overline{ML}_S$	$\overline{Var} \times 10^4$		$MSE(ML_S) \times 10^4$
0.08061	7.459	*0.014*	*−0.689*	0.0101
0.45812	.	437.0	*−0.036*	0.3253
0.98405	.	.	1.1882	0.00150

Table 3: *Performance of standard smooth SIR particle filter for* $T = 550$. The model is AR(1) + noise with true parameters $\phi = 0.975$, $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$. *Additionally the measurement noise is fixed at* $\sigma_\varepsilon = \sqrt{2}$. *The off diagonal elements of the variance matrix are are reported as correlations not covariances covariances.*

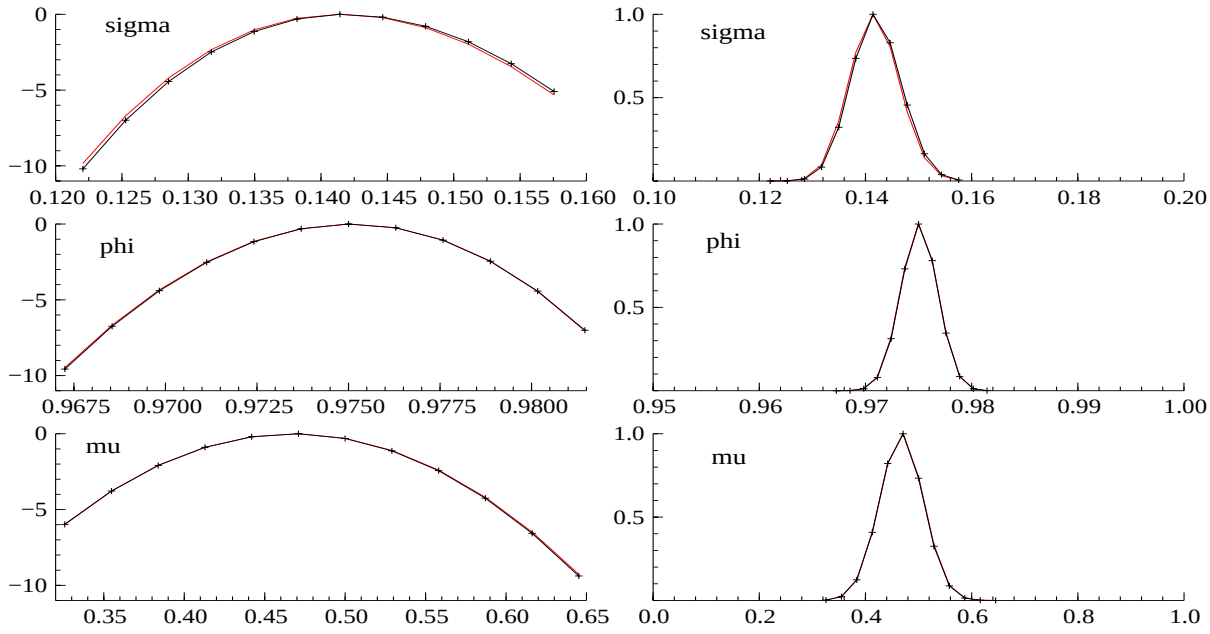

Figure 4: *The true and estimated log-likeliood (left) and likelihood (right) profiles for the AR(1) model plus noise for (from top to bottom)* σ_η, ϕ, μ. *Length of series T=20000.* True parameters $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$, $\phi = 0.975$. M=1000, R=1300.

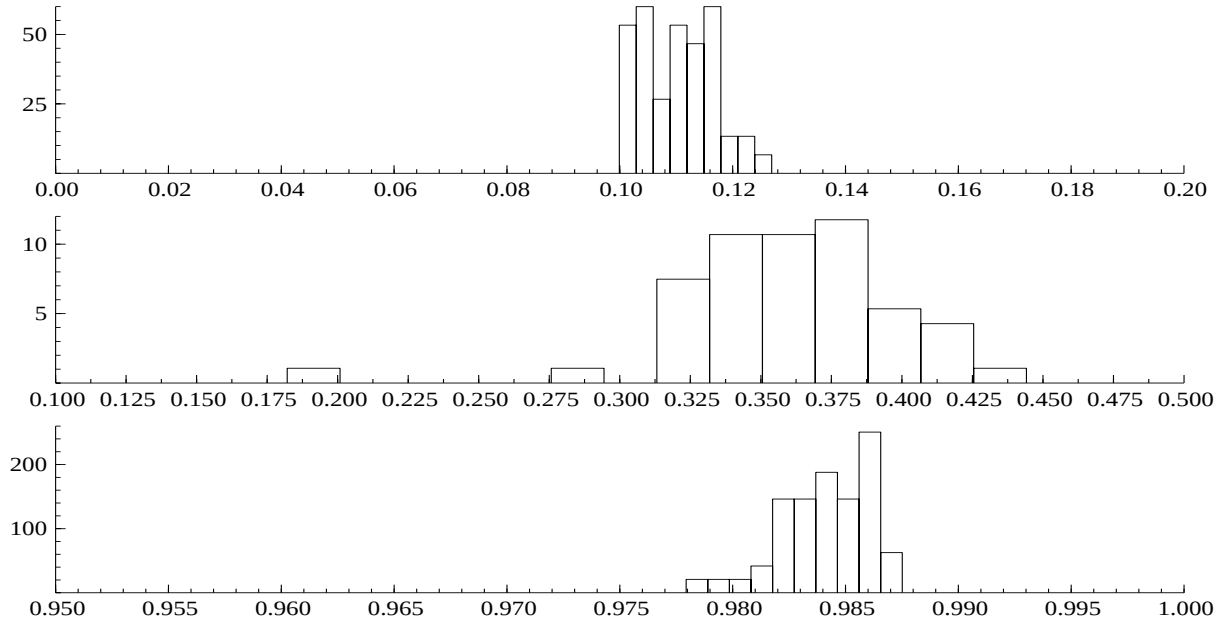

Figure 5: *Histogram of the Monte Carlo samples of the ML estimates for* $(\sigma, \mu, \phi)'$, *from top to bottom, for the SV model for* $T = 550$. True parameters $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$, $\phi = 0.975$. *M=300, R=400.*

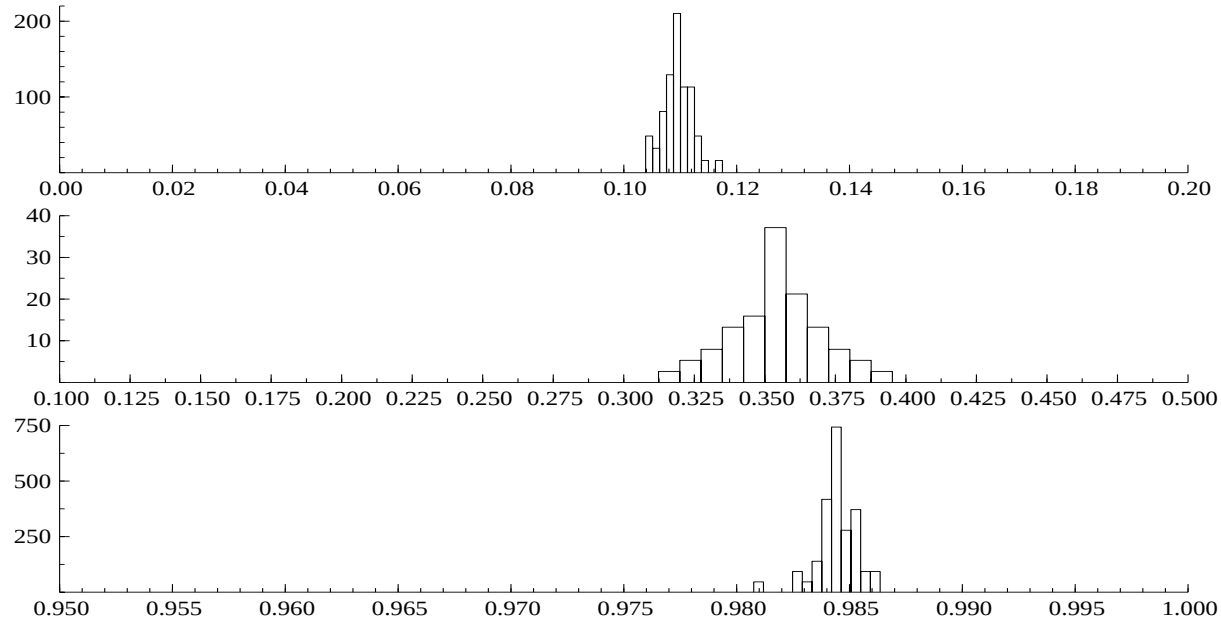

Figure 6: *Histogram of the Monte Carlo samples of the ML estimates for* $(\sigma, \mu, \phi)'$*, from top to bottom, for the SV model for* $T = 550$. True parameters $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$, $\phi = 0.975$. *M=1000, R=1300.*

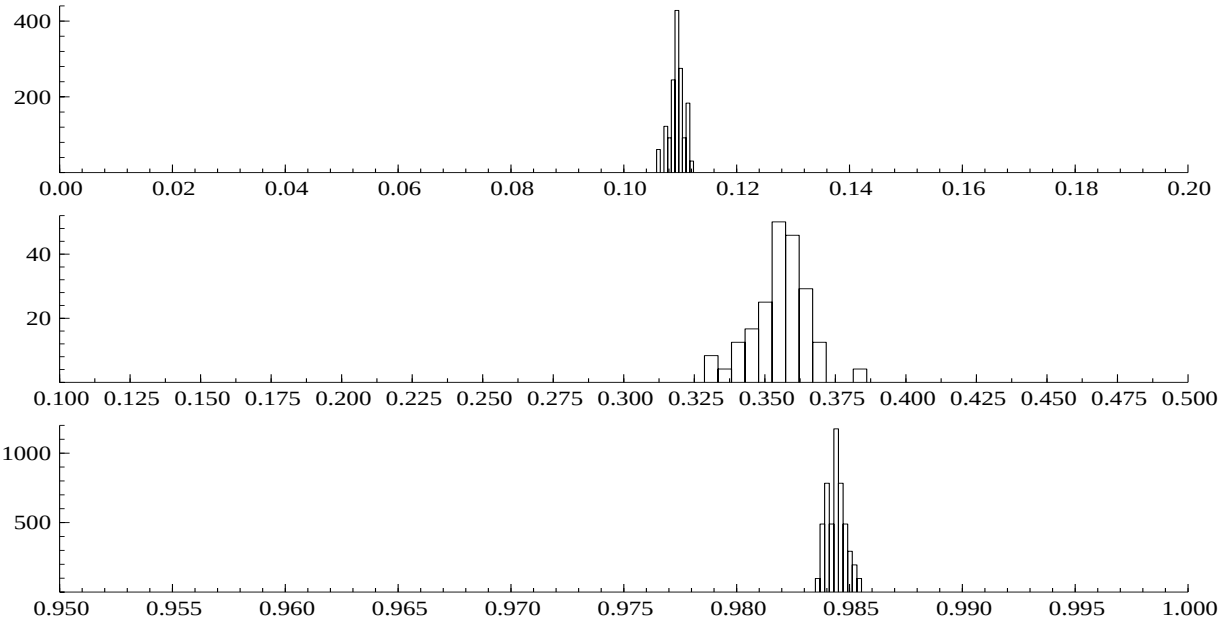

Figure 7: *Histogram of the Monte Carlo samples of the ML estimates for* $(\sigma, \mu, \phi)'$*, from top to bottom, for the SV model for* $T = 550$. True parameters $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$, $\phi = 0.975$. *M=3000, R=4000.*

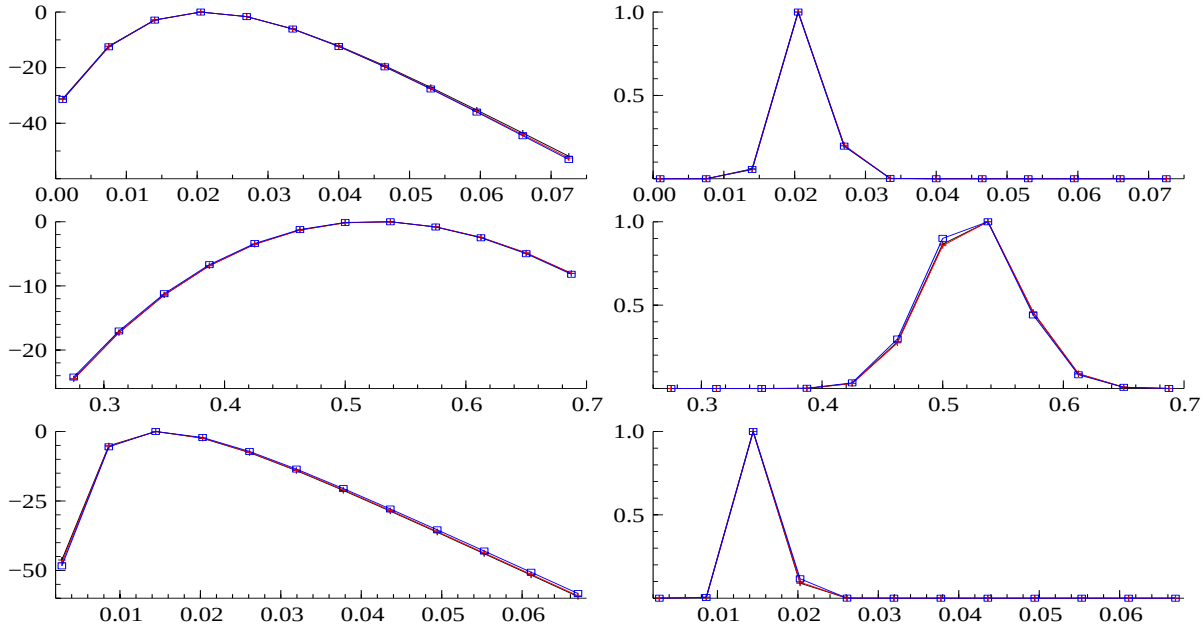

Figure 8: *The log-likeliood (left) and likelihood (right) profiles for the Nelson volatility model. The parameters are* (k, θ, ξ) *from top to bottom.* $n = 4000$, *true parameters* $\Psi = (0.02, 0.5, 0.0178)$. *M=1000, R=1300.*

SIR Particle Filter; $M = 300$, $R = 400$, $SIM = 50$.				
$\overline{ML_S}$	$\overline{Var} \times 10^4$			$Var(ML_S) \times 10^4$
0.11042	10.945	*0.004*	*−0.678*	0.3831
0.35993	.	865.20	*−0.135*	14.910
0.98417	.	.	1.2707	0.0390

SIR Particle Filter; $M = 1000$, $R = 1300$, $SIM = 50$.				
$\overline{ML_S}$	$\overline{Var} \times 10^4$			$Var(ML_S) \times 10^4$
0.10944	10.816	*0.0066*	*−0.673*	0.06516
0.35377	.	882.06	*−0.148*	2.4167
0.98442	.	.	1.2410	0.007897

SIR Particle Filter; $M = 3000$, $R = 4000$, $SIM = 50$.				
$\overline{ML_S}$	$\overline{Var} \times 10^4$			$Var(ML_S) \times 10^4$
0.10941	10.843	*0.0084*	*−0.674*	0.0182
0.35497	.	879.65	*−0.147*	0.978
0.98441	.	.	1.239	0.0018

Table 4: *Performance of standard smooth SIR particle filter for* $T = 550$. The model is SV with true parameters $\phi = 0.975$, $\sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$. *The off diagonal elements of the variance matrix are are reported as correlations not covariances.*

T	True deriv	$M = 300$, R=400 (mean, var)	$M = 1000$, R=1300 (mean, var)	$M = 3000$, R=4000 (mean, var)
250	-24.419	(−0.4210, 5.0619)	(−0.0276, 0.8721)	(0.02201, 0.1766)
500	-46.428	(−0.8541, 8.9987)	(0.0200, 1.5193)	(0.011169, 0.34916)
1000	-90.289	(−1.1570, 21.478)	(−0.00945, 2.4145)	(0.017049, 0.71764)
10,000	-285.93	(5.4056, 201.51)	(0.085527, 27.430)	(−0.10459, 8.799)

Table 5: *Monte Carlo assesement of the error in score estimation for the AR(1) + noise model with* $\sigma_\varepsilon = \sqrt{2}$, $\phi = 0.975, \sigma_\eta = \sqrt{0.02}$, $\mu = 0.5$. The score is with respect to σ_η. Number of Monte Carlo replications = 50.

GARCH plus noise model: $M = 500$, $R = 600$, $SIM = 100$, $n = 500$.					
$\overline{ML_S}(\beta_0, \beta_1, \beta_2, \sigma)'$	$\overline{Var} \times 10^4$				$Var(ML_S) \times 10^4$
0.01080	0.23094	*−0.3984*	*0.0365*	*−0.6078*	0.000718
0.22563	.	49.540	*−0.8860*	*0.7468*	0.158110
0.71723	.	.	39.018	*−0.6539*	0.082196
0.14436	.	.	.	70.991	0.36961

Table 6: Monte Carlo results for GARCH + error model.

GARCH model. Log-lik = −548.60.				
Par	ML	Var × 10^{-4}		
β_0	0.00602	0.25517	0.16912	−0.65797
β_1	0.02703	.	1.1312	−1.3720
β_2	0.96220	.	.	2.6044

GARCH+error model. $M = 3000, 4000$. Log-lik = −545.26.					
β_0	0.000644	0.00437	−0.0149	0.00096	−0.0765
β_1	0.12874	.	7.725	−7.672	2.398
β_2	0.86911	.	.	7.692	−2.183
σ	0.55315	.	.	.	7.225

Table 7: Estimation of both GARCH and GARCH + error models from 1/2/1981 to 12/29/1982. T=500.