# PARTICLE METHODS FOR OPTIMAL FILTER DERIVATIVE: APPLICATION TO PARAMETER ESTIMATION

George Poyiadjis - Arnaud Doucet - Sumeetpal S. Singh
Cambridge University Engineering Department, Cambridge, UK.
Email: gp243,ad2,sss40@eng.cam.ac.uk

## ABSTRACT

Particle filtering techniques are a popular set of simulation-based methods to perform optimal state estimation in non linear non Gaussian dynamic models. However, in applications related to control and identification, it is often necessary to be able to compute the derivative of the optimal filter with respect to parameters of the dynamic model. Several methods have already been proposed in the literature. In experiments, the approximation errors increase with the dataset length. We propose here original particle methods to approximate numerically the filter derivative. In simulations, these methods do not suffer from the problem mentioned. Applications to batch and recursive parameter estimation are presented.

*Keywords*: Optimal Filtering, Parameter Estimation, Sequential Monte Carlo, State-Space Models, Stochastic Approximation.

## 1 Introduction

### 1.1 State-Space Models

Let $\{X_n\}_{n\geq 0}$ and $\{Y_n\}_{n\geq 1}$ be $\mathbb{R}^p$ and $\mathbb{R}^q$-valued stochastic processes defined on a measurable space $(\Omega, \mathcal{F})$ and $\theta \in \Theta$ where $\Theta$ is an open subset of $\mathbb{R}^k$. The process $\{X_n\}_{n\geq 0}$ is an unobserved (hidden) Markov process of initial density $\mu$; i.e. $X_0 \sim \mu$, and Markov transition density $f_\theta(x'|x)$; i.e.

$$X_{n+1}|X_n = x \sim f_\theta(\cdot|x). \tag{1}$$

One observes the process $\{Y_n\}_{n\geq 1}$. It is assumed that the observations conditioned upon $\{X_n\}_{n\geq 0}$ are independent with marginal density $g_\theta(y|x)$; i.e.

$$Y_n|X_n = x \sim g_\theta(\cdot|x). \tag{2}$$

This class of models includes many nonlinear and non-Gaussian time series models such as

$$X_{n+1} = \varphi_\theta(X_n, V_{n+1}),\ Y_n = \psi_\theta(X_n, W_n)$$

where $\{V_n\}_{n\geq 1}$ and $\{W_n\}_{n\geq 1}$ are independent sequences.

Notation: for any sequence $z_k$, let $z_{i:j} = (z_i, z_{i+1}, \ldots, z_j)$.

## 2 Optimal Filter and Its Derivative

The optimal filter is the posterior density of $X_n$ given the observation $Y_{1:n}$ denoted $p_\theta(x_n|Y_{1:n})$. Introducing an arbitrary distribution $q_\theta(x_n|Y_n, x_{n-1})$ whose support includes the support of $g_\theta(Y_n|x_n) f_\theta(x_n|x_{n-1})$, one can write

$$p_\theta(x_n|Y_{1:n}) \propto \int \alpha_\theta(x_{n-1:n}, Y_n)\, q_\theta(x_n|Y_n, x_{n-1})\, p_\theta(x_{n-1}|Y_{1:n-1})\, dx_{n-1} \tag{3}$$

where

$$\alpha_\theta(x_{n-1:n}, Y_n) = \frac{g_\theta(Y_n|x_n)\, f_\theta(x_n|x_{n-1})}{q_\theta(x_n|Y_n, x_{n-1})}. \tag{4}$$

Particle methods are typically used to approximate numerically these recursions for non linear non Gaussian models, [5].

In applications such as parameter estimation and control, it is often of interest to approximate the derivative of the filter $\nabla p_\theta(x_n|Y_{1:n})$. Under regularity assumptions, we obtain the following recursion for the filter derivative by taking the derivative of (3)

$$\nabla p_\theta(x_n|Y_{1:n}) = \pi_\theta(x_n, Y_n) - p_\theta(x_n|Y_{1:n}) \int \pi_\theta(x_n, Y_n)\, dx_n$$

where

$$\begin{aligned}&\pi_\theta(x_n, Y_n) = \\ &\left[\int\int \alpha_\theta(x_{n-1:n}, Y_n)\, q_\theta(x_n|Y_n, x_{n-1})\, p_\theta(x_{n-1}|Y_{1:n-1})\, dx_{n-1:n}\right]^{-1} \\ &\times \Big\{\int \nabla\alpha_\theta(x_{n-1:n}, Y_n)\, q_\theta(x_n|Y_n, x_{n-1})\, p_\theta(x_{n-1}|Y_{1:n-1})\, dx_{n-1:n} \\ &+ \int \alpha_\theta(x_{n-1:n}, Y_n)\, \nabla q_\theta(x_n|Y_n, x_{n-1})\, p_\theta(x_{n-1}|Y_{1:n-1})\, dx_{n-1} \\ &+ \int \alpha_\theta(x_{n-1:n}, Y_n)\, q_\theta(x_n|Y_n, x_{n-1})\, \nabla p_\theta(x_{n-1}|Y_{1:n-1})\, dx_{n-1}\Big\}\end{aligned} \tag{5}$$

The main objective of this paper is to derive particle methods to approximate $\nabla p_\theta(x_n|Y_{1:n})$.

### 2.1 Contributions

There has been many papers devoted to particle filtering over the last few years. However, only few authors have addressed the problem of computing the filter derivative. The first algorithm was proposed in [3] for a specific continuous-time model and [7] for an extension to more general partially observed diffusions. An algorithm that uses a different resampling strategy has been proposed for general discrete-time models [4]. None of these algorithms appear satisfactory and experiments demonstrate that the approximation errors they produce tend to increase with the datalength. We propose here several original methods which appear in simulation not to suffer from these problems.

In Section 2, we detail a particle method for filter derivative which applies to general non linear non Gaussian models and discuss briefly their extensions for conditionally linear Gaussian models and partially observed Gaussian models. In Section 3, we show how it is possible to use the filter derivative so as to perform batch and recursive parameter estimation and present several applications.

## 3 Particle Methods for Filter Derivative

### 3.1 Methodology

The filter derivative $\nabla p_\theta(x_n|Y_{1:n})$ is a signed measure; i.e. it can take positive and negative values. The basic idea of the paper in [3], which is adopted in [4], is to approximate both the filter and its derivative by the same set of particles but weighted differently; i.e.

$$\widehat{p}_\theta(x_n|Y_{1:n}) = \sum_{i=1}^N \alpha_n^{(i)} \delta\left(x_n - X_n^{(i)}\right), \tag{6}$$

$$\widehat{\nabla p}_\theta(x_n|Y_{1:n}) = \sum_{i=1}^N \alpha_n^{(i)} \beta_n^{(i)} \delta\left(x_n - X_n^{(i)}\right), \tag{7}$$

so $\beta_n^{(i)}$ corresponds to an approximation of the so-called score

$$\nabla \log p_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right).$$

However, loosely speaking, the previous methods approximate the filter derivative based on the following relation

$$\begin{aligned}
&\nabla p_\theta(x_n|Y_{1:n}) \\
&= \int \nabla p_\theta(x_{0:n}|Y_{1:n})\, dx_{0:n-1} \\
&= \int p_\theta(x_{0:n}|Y_{1:n})\, \nabla \log p_\theta(x_{0:n}|Y_{1:n})\, dx_{0:n-1}.
\end{aligned}$$

When this quantity is approximated numerically, it relies on a particle approximation on the path space. Consequently the variance increases over time, hence the poor performance of the algorithm. We propose here an alternative based on a direct pointwise approximation of $\nabla p_\theta(x_n|Y_{1:n})$.

### 3.2 Algorithm

Assume that at time $n-1$, we have particle approximations of $p_\theta(x_{n-1}|Y_{1:n-1})$ and $\nabla p_\theta(x_{n-1}|Y_{1:n-1})$ of the form (6)-(7). Substituting (6) into (3), we obtain the following pointwise approximation of the filter

$$\widetilde{p}_\theta(x_n|Y_{1:n}) \propto \sum_{i=1}^N \alpha_{n-1}^{(i)} \alpha_\theta\left(X_{n-1}^{(i)}, x_n, Y_n\right) q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right). \tag{8}$$

At time $n$, we construct the following importance distribution to impute particles

$$\begin{aligned}
q_\theta(x_{n-1}, x_n|Y_{1:n}) = &\left(\sum_{i=1}^N \eta_n^{(i)} \delta\left(x_{n-1} - X_{n-1}^{(i)}\right)\right) \\
&\times q_\theta(x_n|Y_n, x_{n-1})
\end{aligned} \tag{9}$$

whose marginal is

$$q_\theta(x_n|Y_{1:n}) = \sum_{i=1}^N \eta_n^{(i)} q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right). \tag{10}$$

In practice, one should select the parameters of the importance distribution $q_\theta(x_{n-1}, x_n|Y_{1:n})$ such that

$$q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right) \propto g_\theta(Y_n|x_n)\, f_\theta\left(x_n|X_{n-1}^{(i)}\right) \tag{11}$$

and

$$\eta_n^{(i)} \propto \alpha_{n-1}^{(i)} \int g_\theta(Y_n|x_n)\, f_\theta\left(x_n|X_{n-1}^{(i)}\right) dx_n. \tag{12}$$

In most cases, one cannot sample from (11) and/or compute (12) but one can use an approximation of these quantities.

To sample from $q_\theta(x_{n-1}, x_n|Y_{1:n})$, we first sample $\overline{X}_{n-1}^{(j)}$ where

$$\Pr\left(\overline{X}_{n-1}^{(j)} = X_{n-1}^{(i)}\right) = \eta_n^{(i)}$$

then

$$X_n^{(j)} \middle| Y_n, \overline{X}_{n-1}^{(j)} \sim q_\theta\left(\cdot \middle| Y_n, \overline{X}_{n-1}^{(j)}\right).$$

In practice, we sample the particles $\left\{\overline{X}_{n-1}^{(j)}\right\}$ using a stratified sampling strategy; any other standard resampling algorithm developed in the particle filtering framework could be used.

By plugging (6), (7) and (10) in (5) we obtain the following pointwise approximation of the filter derivative

$$\begin{aligned}
&\widetilde{\nabla p}_\theta(x_n|Y_{1:n}) \\
&= \left[N^{-1}\sum_{i=1}^N \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \alpha_\theta\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right]^{-1} \\
&\times \left[\sum_{i=1}^N \alpha_{n-1}^{(i)} \nabla\alpha_\theta\left(X_{n-1}^{(i)}, x_n, Y_n\right) q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right)\right. \\
&+ \sum_{i=1}^N \alpha_{n-1}^{(i)} \alpha_\theta\left(X_{n-1}^{(i)}, x_n, Y_n\right) \nabla q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right) \\
&\left.+ \sum_{i=1}^N \alpha_{n-1}^{(i)} \beta_{n-1}^{(i)} \alpha_\theta\left(X_{n-1}^{(i)}, x_n, Y_n\right) q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right)\right] \\
&- \left[N^{-1}\sum_{i=1}^N \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \alpha_\theta\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right]^{-2} \\
&\times N^{-1}\sum_{i=1}^N \alpha_{n-1}^{(i)} \alpha_\theta\left(X_{n-1}^{(i)}, x_n, Y_n\right) q_\theta\left(x_n|Y_n, X_{n-1}^{(i)}\right) \\
&\times \left[\sum_{i=1}^N \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \nabla\alpha_\theta\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right. \\
&+ \sum_{i=1}^N \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \alpha_\theta\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right) \nabla \log q_\theta\left(X_n^{(i)} \middle| Y_n, \overline{X}_{n-1}^{(i)}\right) \\
&\left.+ \sum_{i=1}^N \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \beta_{n-1}^{(i)} \alpha_\theta\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right]
\end{aligned} \tag{13}$$

where $\varphi(i)$ is the discrete index such that $\overline{X}_{n-1}^{(i)} = X_{n-1}^{\varphi(i)}$. Now the algorithm proceeds as follows at time $n$.

*Sampling Step*. Sample $\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}\right) \sim q_\theta(\cdot, \cdot|Y_{1:n})$ and compute using (8)-(13)-(10)

$$\alpha_n^{(i)} \propto \frac{\widetilde{p}_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)}{q_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)}, \quad \sum_{i=1}^N \alpha_n^{(i)} = 1, \tag{14}$$

$$\beta_n^{(i)} = \alpha_n^{-1(i)} \frac{\widetilde{\nabla p}_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)}{q_\theta\left(X_n^{(i)} \middle| Y_{1:n}\right)}. \tag{15}$$

*Remark*. Note that in this approach there is no explicit resampling step but it is actually performed when sampling from (9).

*Remark.* As a byproduct of the algorithm for filter derivative, we obtain an estimate of the filter which is different from standard methods. Compared to the Auxiliary Particle Filter (APF) [11], computing the importance weights (14) require $O\left(N^2\right)$ operations instead of $O(N)$. However, for a fixed number of particles, this algorithm will outperform the APF because we integrate analytically a discrete latent variable.

*Remark.* It is possible to modify the algorithm to ensure that $\sum_{i=1}^N \beta_n^{(i)} = 0$. Because of a lack of space, this is not presented here.

### 3.3 Extensions

For an important class of dynamic models, it is possible to integrate analytically a subset of the state variables; hence reducing the variance of the Monte Carlo estimates. This has been extensively used to develop efficient particle filters; e.g. [1], [5].

Consider the following conditionally linear Gaussian state-space model

$$\begin{array}{l} X_0 \sim \mu,\ X_n|\, X_{n-1} = x \sim f_\theta\left(\cdot|\, x\right) \\ Z_0 \sim \mathcal{N}\left(m, \Sigma\right),\, Z_n = A_\theta\left(X_n\right) Z_{n-1} + B_\theta\left(X_n\right) V_n, \\ Y_n = C_\theta\left(X_n\right) Z_n + D_\theta\left(X_n\right) W_n, \end{array}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, I\right)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, I\right)$. Conditional upon $\{X_n\}_{n\geq 0}$, the model $\{Y_n, Z_n\}$ is linear Gaussian. It follows that $p\left(x_{0:n}|\, Y_{1:n}\right)$ can be computed up to a normalizing constant as this distribution satisfies

$$p\left(x_{0:n}|\, Y_{1:n}\right) \propto \mu\left(x_0\right) \prod_{k=1}^{n} f\left(x_k|\, x_{k-1}\right) p\left(Y_k|\, Y_{1:k-1}, x_{0:n}\right)$$

where $p\left(Y_k|\, Y_{1:k-1}, x_{0:n}\right)$ can be computed pointwise using a Kalman filter. In this case, it is possible to develop an efficient particle filter which corresponds to a mixture of Kalman filters. The derivation of the algorithm for filter derivative is too long to be presented here but consists of a mixture of Kalman filters and its derivatives. Similar developments can be performed for partially observed Gaussian state-space models [1].

## 4 Application to Parameter Estimation

### 4.1 Batch ML

Assuming the true parameter $\theta^*$ generating the data $Y_{1:n}$ is unknown, we can estimate it by maximizing the following likelihood function

$$\begin{aligned} L_\theta\left(Y_{1:n}\right) = \sum_{k=1}^{n} \log\Bigg(\int \alpha_\theta\left(x_{k-1:k}, Y_k\right) q_\theta\left(x_k|\, Y_k, x_{k-1}\right) \\ \times\ p_\theta\left(x_{k-1}|Y_{1:k-1}\right) dx_{k-1:k}\Bigg) \end{aligned} \tag{16}$$

with the convention $Y_{1:0} = \varnothing$. To obtain the maximum likelihood estimate, we propose to use a simple gradient ascent algorithm. The gradient of the likelihood can directly be estimated numerically by

$$\begin{array}{l} \nabla L_\theta\left(Y_{1:n}\right) = \sum_{k=1}^{n} \left(\sum_{i=1}^{N} \eta_k^{-1\varphi(i)} \alpha_{k-1}^{\varphi(i)} \alpha_\theta\left(\overline{X}_{k-1}^{(i)}, X_k^{(i)}, Y_k\right)\right)^{-1} \\ \times \left[\sum_{i=1}^{N} \eta_k^{-1\varphi(i)} \alpha_{k-1}^{\varphi(i)} \nabla\alpha_\theta\left(\overline{X}_{k-1}^{(i)}, X_k^{(i)}, Y_k\right)\right. \\ + \sum_{i=1}^{N} \eta_k^{-1\varphi(i)} \alpha_{k-1}^{\varphi(i)} \alpha_\theta\left(\overline{X}_{k-1}^{(i)}, X_k^{(i)}, Y_k\right) \nabla \log q_\theta\left(X_k^{(i)}\middle|\, Y_k, \overline{X}_{k-1}^{(i)}\right) \\ \left. + \sum_{i=1}^{N} \eta_k^{-1\varphi(i)} \alpha_{k-1}^{\varphi(i)} \beta_{k-1}^{\varphi(i)} \alpha_\theta\left(\overline{X}_{k-1}^{(i)}, X_k^{(i)}, Y_k\right)\right]. \end{array}$$

### 4.2 Recursive ML

Under regularity assumptions including the stationarity of the state-space model, one has

$$\frac{1}{n} L_\theta\left(Y_{1:n}\right) \to L\left(\theta\right)$$

where $L\left(\theta\right)$ is equal to

$$\int\int_{\mathbb{R}^q \times \mathcal{P}(\mathbb{R}^p)} \log\left(\int g_\theta\left(y|\, x\right) \mu\left(x\right) dx\right) \lambda_{\theta,\theta^*}\left(dy, d\mu\right),$$

where $\mathcal{P}\left(\mathbb{R}^p\right)$ is the space of probability distributions on $\mathbb{R}^p$ and $\lambda_{\theta,\theta^*}\left(dy, d\mu\right)$ is the joint invariant distribution of the couple $\left(Y_k, p_\theta\left(x_k|Y_{1:k-1}\right)\right)$. It is dependent on both $\theta$ and the true parameter $\theta^*$. Maximizing $L\left(\theta\right)$ corresponds to minimizing the following Kullback–Leibler information measure given by

$$K\left(\theta, \theta^*\right) \triangleq L\left(\theta^*\right) - L\left(\theta\right) \geq 0.$$

To optimize this cost function, Recursive Maximum Likelihood (RML) is based on a stochastic gradient algorithm

$$\begin{aligned} \theta_n = \theta_{n-1} + \gamma_n \nabla \log\Bigg(\int \alpha_{\theta_{n-1}}\left(x_{n-1:n}, Y_n\right) q_{\theta_{n-1}}\left(x_n|\, Y_n, x_{n-1}\right) \\ \times\ p_{\theta_{1:n-1}}\left(x_{n-1}|Y_{1:n-1}\right) dx_{n-1:n}\Bigg). \end{aligned} \tag{17}$$

This requires the computation of $p_{\theta_{1:n}}\left(x_n|Y_{1:n}\right)$ and its derivatives with respect to $\theta$ using the parameter $\theta_k$ at time $k$. The step-size sequence $\{\gamma_n\}_{n\geq 1}$ is a positive non-increasing sequence such that $\sum \gamma_n = \infty$ and $\sum \gamma_n^2 < \infty$; typically one selects $\gamma_n = \gamma_0 . n^{-\alpha}$ where $\gamma_0 > 0$ and $0.5 < \alpha \leq 1$ [2].

Numerically, we approximate (17) as

$$\begin{array}{l} \theta_{n+1} = \theta_n + \gamma_n \left(\sum_{i=1}^{N} \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \alpha_{\theta_n}\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right)^{-1} \\ \times \left[\sum_{i=1}^{N} \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \nabla\alpha_{\theta_n}\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right. \\ + \sum_{i=1}^{N} \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \alpha_{\theta_n}\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right) \nabla \log q_{\theta_n}\left(X_n^{(i)}\middle|\, Y_n, \overline{X}_{n-1}^{(i)}\right) \\ \left. + \sum_{i=1}^{N} \eta_n^{-1\varphi(i)} \alpha_{n-1}^{\varphi(i)} \beta_{n-1}^{\varphi(i)} \alpha_{\theta_n}\left(\overline{X}_{n-1}^{(i)}, X_n^{(i)}, Y_n\right)\right] \end{array}$$

### 4.3 Linear Gaussian State Space Model

Let us consider the following scalar linear Gaussian state space model

$$\begin{array}{l} X_{n+1} = \phi X_n + \sigma_V V_{n+1},\ X_0 \sim \mathcal{N}\left(0, 1\right) \\ Y_n = X_n + \sigma_W W_n \end{array}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 1\right)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 1\right)$. We are interested in estimating the parameter $\theta \triangleq \left(\phi, \sigma_V, \sigma_W\right)$. In this case the optimal filter is given by the Kalman filter and it is possible to get exact expressions for its derivative. This allows to compare our numerical methods with the ground truth. Figure 1 compares the particle approximations with the exact values of the derivative of the log-likelihood with respect to the three parameters. The results are almost indistinguishable.

### 4.4 Stochastic Volatility Model

We consider the following model [11]

$$\begin{array}{l} X_{n+1} = \phi X_n + \sigma V_{n+1},\ X_0 \sim \mathcal{N}\left(0, \dfrac{\sigma^2}{1-\phi^2}\right) \\ Y_n = \beta \exp\left(X_n/2\right) W_n \end{array}$$

where $V_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 1\right)$ and $W_n \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(0, 1\right)$ are two mutually independent sequences, independent of the initial state $X_0$. We are interested in estimating the parameter $\theta \triangleq \left(\sigma, \phi, \beta\right)$ where $\Theta = \left(0, \Xi\right) \times \left(-1, 1\right) \times \left(0, \Xi\right)$ with $\Xi = 100$. We apply our batch ML method to the pound/dollar daily exchange rates; see [6]. This time series consists of $945$ data points. The parameter estimates for $M = 1000$ iterations using $N = 1000$ particles are shown in Figure 2.

Our results are consistent with results obtained in [6]. We obtain $\widehat{\theta}_{ML} = \left(0.161, 0.976, 0.628\right)$ whereas the estimate in [6] is $\widehat{\theta}_{ML} = \left(0.173, 0.973, 0.634\right)$.

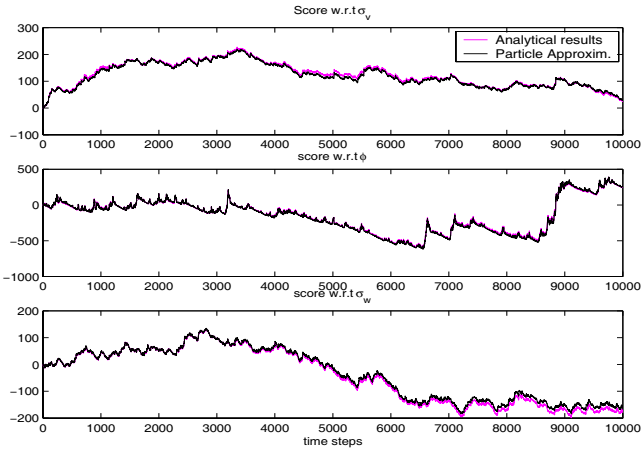


Figure 1: Analytical and numerical results for the scores using $N = 1000$.

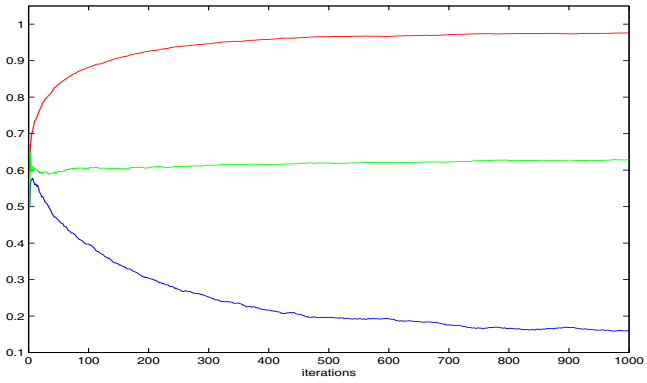


Figure 2: Sequence of parameter estimates for RML $\theta_n = (\sigma_n, \phi_n, \beta_n)$ for $N = 1000$. From top to bottom: $\phi_n$, $\beta_n$ and $\sigma_n$.

### 4.5 Conditionally Linear Gaussian model

The stochastic volatility model of the previous section can be linearized under the transformation $\Psi_n = \log Y_n^2 = X_n + \log \beta^2 + \varepsilon_n$, where $\varepsilon_n = \log W_n^2$. The non-Gaussian distribution of $\{\varepsilon_n\}$ can be well approximated by a mixture of seven Gaussians [8]. We implemented the RML algorithm on the resultant conditionally linear Gaussian model using $N = 1000$ particles using the Rao-Blackwellised algorithm. It converged to a value $\widehat{\theta}$ in the neighborhood of the true parameter $\theta^* = (0.2, 0.98, 0.7)$. Figure 3 shows an example of the estimates obtained.

## 5 Discussion

In this paper, we have proposed original particle methods to estimate the derivative of the optimal filter in general state-space models. These methods can be used to perform batch/on-line parameter estimation but also for control applications. Their computational cost is quadratic in the number of particles. However by taking advantage of fast computation methods [9], we expect that this problem could be significantly mitigated. Finally, from a theoretical viewpoint we are currently studying the stability of these particle approximations.

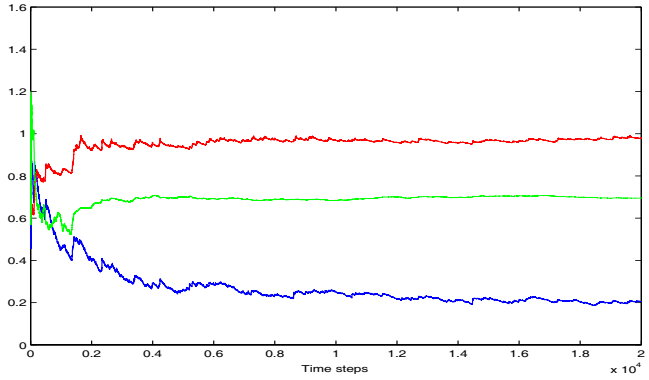


Figure 3: Sequence of parameter estimates for RML $\theta_n = (\sigma_n, \phi_n, \beta_n)$ for $N = 1000$. True parameters were $\theta^* = (0.2, 0.98, 0.7)$. From top to bottom: $\phi_n$, $\beta_n$ and $\sigma_n$.

## 6 REFERENCES


[1] Andrieu, C. and Doucet, A. (2002). Particle filtering for partially observed gaussian state space models. *J. Royal Statist. Soc.* B, **64**, 827-836.

[2] Benveniste, A., Métivier, M. and Priouret, P. (1990). *Adaptive Algorithms and Stochastic Approximation*. New York: Springer-Verlag.

[3] Cérou F., LeGland F. and Newton N.J. (2001) Stochastic particle methods for linear tangent equations. in *Optimal Control and PDE's - Innovations and Applications* (eds. J. Menaldi, E. Rofman & A. Sulem), pp. 231-240, IOS Press, Amsterdam.

[4] Doucet, A. and Tadić, V.B. (2003). Parameter estimation in general state-space models using particle methods. *Ann. Inst. Stat. Math.*, **55**, 409-422.

[5] Doucet, A., de Freitas, J.F.G. and Gordon N.J. (eds.) (2001). *Sequential Monte Carlo Methods in Practice*. New York: Springer-Verlag.

[6] Durbin, J. and Koopman, S. J. (2000). Time series analysis of non-Gaussian observations based on state space models from both classical and Bayesian perspectives (with discussion). *J. R. Statist. Soc.* B, **62**, 3-56.

[7] Guyader, A., LeGland, F. and Oudjane, N. (2003) A particle implementation of the recursive MLE for partially observed diffusions. *Proceedings of the 13th IFAC Symposium on System Identification*, 1305-1310.

[8] Kim, S., Shephard, N. and Chib, S. (1998).Stochastic volatility: likelihood inference and comparison with ARCH models. *Review of Economic Studies*, **65**, 361-393.

[9] Klaas, M., Lang, D., Hamze, F. and de Freitas, N. (2004) Fast probability propagation: Beyond belief(s). Technical report, CS Department, University of British Columbia.

[10] LeGland F. and Mevel L. (1997) Recursive identification in hidden Markov models, *Proc. 36th IEEE Conf. Decision and Control*, pp. 3468-3473.

[11] Pitt, M.K. and Shephard, N. (1999). Filtering via simulation: auxiliary particle filter. *J. Am. Statist. Ass.*, **94**, 590-599.