Bayesian Filtering for Jump-Diffusions with Applications to Stochastic Volatility

Andrew Golightly*
Newcastle University, NE1 7RU, U.K.

October 3, 2007

Abstract

In this paper, the problem of sequentially learning parameters governing discretely observed jump-diffusions is explored. The estimation framework involves the introduction of $m-1$ latent points between every pair of observations to allow a sufficiently accurate Euler-Maruyama approximation of the underlying (but unavailable) transition densities. Particle filtering algorithms are then implemented to sample the posterior distribution of the latent data and the model parameters online. The methodology is applied to the estimation of parameters governing a stochastic volatility (SV) model with jumps. As well as using S&P 500 index data, a simulation study is provided.

1 Introduction

Recently, much attention has been given to diffusion driven models that incorporate jumps. Indeed, the increasing popularity of such models can be attributed to their role in finance; stochastic volatility (SV) models with jumps in returns have been examined by Johannes, Polson & Stroud (2006*b*) and Liu, Longstaff & Pan (2001) among others, however, Bates (1996), Duffie, Singleton & Pan (2000) and Pan (2002) find that such models are misspecified and propose models with jumps in both returns and volatility (see also the work by Bakshi, Cao & Chen (1997), Eraker, Johannes & Polson (2003) and Eraker (2004)).

This paper considers the problem of sequential parameter estimation in diffusion driven SV models with jump components in both returns and volatility. Whilst global MCMC schemes are reasonably well developed for such models (see for example Eraker et al. (2003), Eraker (2004) and Raggi & Bordignon (2006)) relatively little work has addressed the filtering problem; Johannes et al. (2006*b*) perform sequential parameter inference for discrete time models with jumps in returns only and Johannes, Polson & Stroud (2006*a*) filter latent states in SV models with jumps whilst holding parameter values constant.

The reason for sequential learning is clear in the financial setting. Returns data arrive almost continuously and as each new data point becomes available, global MCMC schemes must be started from scratch to include the new observation. The sequential schemes developed here are computationally attractive and build on recent work in

*a.golightly@ncl.ac.uk

the area of sequential Monte Carlo. For fixed parameters particle filtering as been explored extensively in the context of discrete time state space models (Gordon, Salmond & Smith 1993, Pitt & Shephard 1999, Doucet, Godsill & Andrieu 2000) whilst sequential learning for unknown static parameters and states has been examined by Storvik (2002), Fearnhead (2002) (using sufficient statistics) and also Liu & West (2001), Gilks & Berzuini (2001) and Johannes et al. (2006*b*). For applications of particle filters to diffusion processes see Del Moral, Jacod & Protter (2002), Golightly & Wilkinson (2006*a*), Golightly & Wilkinson (2006*b*) and Johannes et al. (2006*a*).

The first particle filtering algorithm we develop extends the simulation filter of Golightly & Wilkinson (2006*a*) to diffusions with jump components. The methodology uses the technique of introducing latent data points between every pair of observations to allow the Euler-Maruyama approximation of the true transition densities (Pedersen 1995). The posterior distribution of the latent data and the model parameters is then sampled on-line via sequential MCMC. The algorithm of Storvik (2002) is also applied — it is found that both algorithms perform well on simulated data, however, when using the S&P 500 index data, the simulation filter struggles to adapt to new information arriving after the Crash of 1987. To assess the validty of each method, a simulation study is provided and the output of each filter is compared to the output of the Gibbs sampler of Eraker (2004).

The remainder of this paper is organised as follows. The model is formulated in Section 2 and the general estimation framework is described. Particle filtering algorithms are discussed in Section 3 and the methodology is illustrated in Section 5 before conclusions are drawn in Section 6.

2 Models and Estimation framework

Consider inference for a stochastic volatility model of the form

$$\begin{cases} dX_t &= \alpha\, dt + \sqrt{Z_t}\, dW_t^x + V^x\, dN_t \\ dZ_t &= (\theta + \kappa Z_t)\, dt + \sigma_z \sqrt{Z_t}\, dW_t^z + V^z\, dN_t \end{cases} \tag{1}$$

Here X_t is the logarithm of an asset's price (typically scaled by a factor of 100), Z_t is an unobserved volatility process, W_t^x and W_t^z are Brownian motions with correlation ρ, N_t is a Poisson process with constant intensity λ, and finally $V^x \sim \mathrm{N}(\mu_x, \sigma_x)$ and $V^z \sim \exp(\mu_z)$ are jump sizes in respective returns and volatilities. Note that this model belongs to the affine jump diffusion family introduced in Duffie et al. (2000) and examined by Eraker et al. (2003) and Eraker (2004). By removing the jump components ($\lambda = 0$), (1) reduces to the square-root model of Heston (1993). The SVJ model considered by Bates (1996) has Normally distributed jumps in returns only whilst Eraker et al. (2003) consider the SVIJ model with independently arriving jumps in volatility and returns. Raggi & Bordignon (2006) examine the SVCJ model with common jump component and correlated jump sizes, $V^z \sim \exp(\mu_z)$ and $V^x \sim \mathrm{N}(\mu_x + \rho_j V^z, \sigma_x^2)$. For the applications considered in Section 5, attention will be focused on the model given by (1) with $\rho = 0$ (and note that this is necessary to implement the particle filter of Storvik). Henceforth the estimation framework will be presented in this context.

Let Θ denote the vector of unknown parameters in the model. Then the goal is to infer Θ based on discrete time observations of X_t only. By adopting the Bayesian imputation approach previously persued by Pedersen (1995), it is necessary to work with the discretised version of (1), given by the Euler approximation,

$$\begin{cases} X_{t+\Delta t} &= X_t + \alpha\,\Delta t + \sqrt{Z_t}\,\Delta W_t^x + V_{t+\Delta t}^x\,J_{t+\Delta t} \\ Z_{t+\Delta t} &= Z_t + (\theta + \kappa Z_t)\,\Delta t + \sigma_z\sqrt{Z_t}\,\Delta W_t^z + V_{t+\Delta t}^z\,J_{t+\Delta t} \end{cases} \tag{2}$$

where ΔW_t^x and ΔW_t^z are independent $\mathrm{N}(0, \Delta t)$ random variables, and $J_{t+\Delta t}$ is a Bernouilli random variable with constant intensity $\lambda\Delta t$. Now suppose that measurements $x(\tau_i)$ are available at evenly spaced times τ_0, τ_1, ..., τ_T with intervals of length $\Delta^* = \tau_{i+1} - \tau_i$. For weekly or monthly asset data, Δ^* is often too large to be used as a time step in (2). We therefore put $\Delta t = \Delta^*/m$ for some positive integer $m \geq 1$. Then, choosing m to be sufficiently large ensures that the discretisation bias is arbitrarily small, but this also introduces the problem of $m - 1$ missing values in between every pair of observations.

In order to provide a framework for dealing with these missing values, the entire time interval $[\tau_0, \tau_T]$ is divided into $mT + 1$ equidistant points $\tau_0 = t_0 < t_1 < \ldots < t_n = \tau_T$ (where $n = mT$) such that X_t is observed at times $t_0, t_m, \ldots, t_n$. To fix the notation, stack all augmented values (both observed and missing) of X_t and Z_t in the matrix Y, let J contain all the jump times and V contain all the jump sizes. That is

$$Y = \begin{pmatrix} x_{t_0} & X_{t_1} & \cdots & x_{t_m} & X_{t_{m+1}} & \cdots\cdots & x_{t_n} \\ Z_{t_0} & Z_{t_1} & \cdots & Z_{t_m} & Z_{t_{m+1}} & \cdots\cdots & Z_{t_n} \end{pmatrix},$$

$$V = \begin{pmatrix} V_{t_1}^x & \cdots & V_{t_m}^x & V_{t_{m+1}}^x & \cdots\cdots & V_{t_n}^x \\ V_{t_1}^z & \cdots & V_{t_m}^z & V_{t_{m+1}}^z & \cdots\cdots & V_{t_n}^z \end{pmatrix},$$

$$J = \begin{pmatrix} J_{t_1} & \cdots & J_{t_m} & J_{t_{m+1}} & \cdots\cdots & J_{t_n} \end{pmatrix}.$$

Now let Y_i, V_i and J_i denote the ith columns of Y, V and J respectively. By adopting a fully Bayesian approach, *a priori* beliefs about Θ, Z_0, V_1 and J_1 are summarised via the prior distribution $\pi(\Theta, Z_0, V_1, J_1)$. Then the joint posterior density for parameters and augmented data is given by

$$\begin{aligned} \pi(Y, V, J, \Theta | D_n) \quad &\propto \quad \pi(\Theta, Z_0, V_1, J_1) \prod_{i=0}^{n-1} \pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}, \Theta) \\ &\qquad \times\ \pi(V_{i+1}|J_{i+1}, \Theta)\pi(J_{i+1}|\Theta) \end{aligned} \tag{3}$$

where $D_n = (x_0, x_m, \ldots x_n)$, $\pi(V_{i+1}|J_{i+1}, \Theta)$ is obtained from the distributional form of the jump sizes, $\pi(J_{i+1}|\Theta)$ is the Bernouilli p.m.f. with parameter $\lambda\Delta t$ and finally $\pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}, \Theta)$ is the Euler transition density obtained from (2) which can be written as

$$\pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}, \Theta) = \pi(X_{i+1}|Y_i, J_{i+1}, V_{i+1}^x, \Theta)\pi(Z_{i+1}|Z_i, J_{i+1}, V_{i+1}^z, \Theta) \tag{4}$$

where

$$\pi(X_{i+1}|Y_i, J_{i+1}, V_{i+1}^x, \Theta) \;=\; \phi\left(X_{i+1}; X_i + \alpha\Delta t + V_{i+1}^x J_{i+1}\,,\ Z_{i+1}\Delta t\right) \tag{5}$$

$$\pi(Z_{i+1}|Z_i, J_{i+1}, V_{i+1}^z, \Theta) \;=\; \phi\left(Z_{i+1}; Z_i + (\theta - \kappa Z_i)\Delta t + V_{i+1}^z J_{i+1}\,,\ \sigma_z^2 Z_{i+1}\Delta t\right) \tag{6}$$

and $\phi(\cdot, \psi, \Sigma)$ denotes the Gaussian pdf with mean ψ and variance matrix Σ.

As discussed in Tanner & Wong (1987), inference may proceed by alternating between simulation of parameters conditional on augmented data, and simulation of the missing data given the observed data and the current state of the model parameters. As the joint posterior (3) is usually high dimensional, a Gibbs sampler (Geman &

Geman 1984) is a particularly convenient way of sampling from it. For the model given by (1) with $\rho = 0$, closed form full conditionals are available for each X_i, J_i and V_i, and for each parameter. A Metropolis-Hastings (M-H) step can be implemented to sample the full conditional of each Z_i (known as Metropolis within Gibbs). The Gibbs sampling approach has been persued in the context of SV models with jumps by Eraker et al. (2003) and Eraker (2004) whilst other global MCMC schemes have been implemented by Raggi & Bordignon (2006) among others. Such schemes require, however, that if new data become available, parameter samples must be discarded and the sampler restarted to include the new information. Furthermore, computational burden is increased with every observation and for very large datasets (common in finance) running the algorithm may not be feasible.

Attention is therefore turned to the development of two particle filtering algorithms. Firstly, the simulation filter of Golightly & Wilkinson (2006*a*) is extended to diffusions with jump components. Finally, the particle filter of Storvik (2002) is adapted. Both algorithms sample a new $(\Theta^*, Y^*, V^*, J^*)$ in three stages: first Θ^* is sampled from a suitable proposal and then J^*, V^* are sampled from $J, V|\Theta^*$. Finally Y^* is sampled from a tractable approximation to $Y|V^*, J^*, \Theta^*, D_n$. The latter step is performed using a diffusion bridge construct conditional on the proposed jumps and sizes, and the observed data. It is found that this approach is far more efficient than simply using the Euler approximation to propose the missing values.

3 Sampling Conditioned Diffusions with Jumps

In order to implement the particle filters of Section 4, it is crucial that we can sample the latent data between two observations of the diffusion. Unfortunately, sampling the missing data between two observations that are m steps apart, under the nonlinear structure of the diffusion is difficult. An MCMC step is therefore used here and the remainder of this section deals with the construction of an efficient proposal process.

Consider an arbitrary d-dimensional multivariate jump diffusion of the form

$$dY_t = \mu(Y_t)dt + \beta^{\frac{1}{2}}(Y_t)dW_t + V dN_t \tag{7}$$

for which the Euler–Maruyama approximation is

$$Y_{t+\Delta t} = Y_t + \mu(Y_t)\Delta t + \beta^{\frac{1}{2}}(Y_t)\Delta W_t + V_{t+\Delta t}J_{t+\Delta t} \tag{8}$$

after suppressing any parameter dependence to simplify the notation. Suppose that we have observations $Y_{t_j} = y_j$ and $Y_{t_M} = y_M$ (where $M = j + m$) and divide the interval $[t_j, t_M]$ into $m+1$ points $t_j < t_{j+1} < \ldots < t_{j+m} = t_M$ with each $t_{i+1} - t_i = \Delta t$. Our goal is to approximate a sample of Y_t conditional on our two observations by generating a skeleton bridge $Y_{j+1}, \ldots, Y_{M-1}$ conditioned to start at y_j and finish at y_M. We proceed by assuming that the jump times $J_{j+1}, \ldots, J_M$ and sizes $V_{j+1}, \ldots, V_M$ are known and construct a Gaussian approximation to $\pi(Y_{i+1}|Y_i, y_M, J_{j+1:M}, V_{j+1:M})$ (defined for $i = j, \ldots, M-2$) and we denote the approximate density by $\tilde{\pi}(Y_{i+1}|Y_i, y_M, J_{j+1:M}, V_{j+1:M})$. This density is derived by formulating the joint density of Y_{i+1} and Y_M (conditional on Y_i, plus the jump times and sizes) and then using MVN theory to condition on $Y_M = y_M$. We therefore start with the density of Y_M conditional on Y^{i+1} which we obtain using a very crude Euler approximation

$$\tilde{\pi}(Y_M|Y_{i+1}, J_{i+2:M}, V_{i+2:M}) = \phi\left(Y_M;\, Y_{i+1} + \mu_{i+1}\Delta^+ + \sum_{k=i+2}^{M} V_k J_k\,,\, \beta_{i+1}\Delta^+\right) \tag{9}$$

where $\Delta^+ = (M-i-1)\Delta t$, and the shorthand notation of $\mu_{i+1} = \mu(Y_{i+1})$ is adopted. To give a linear Gaussian structure, we approximate (9) further by noting that μ and β are locally constant (by assumption). Estimating μ_{i+1} and β_{i+1} by μ_i and β_i respectively, we obtain

$$\tilde{\pi}(Y_M|Y_{i+1}, J_{i+2:M}, V_{i+2:M}) = \phi\left(Y_M;\, Y_{i+1} + \mu_i\Delta^+ + \sum_{k=i+2}^{M} V_k J_k\,,\, \beta_i\Delta^+\right). \tag{10}$$

The density $\pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1})$ is the one step ahead Euler transition density given by

$$\pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}) = \phi\left(Y_{i+1};\, Y_i + \mu_i\Delta t + V_{i+1}J_{i+1}\,,\, \beta_i\Delta t\right) \tag{11}$$

and we can therefore combine (10) and (11) to construct the approximate joint density of Y_{i+1} and Y_M (conditional on Y^i, $J_{i+1:M}$ and $V_{i+1:M}$) using MVN conditioning results which yield

$$\begin{pmatrix} Y_{i+1} \\ Y_M \end{pmatrix} \sim \mathrm{N}_{2d}\left\{\begin{pmatrix} Y_i + \mu_i\Delta t + V_{i+1}J_{i+1} \\ Y_i + \mu_i\Delta^- + \sum_{k=i+1}^{M} V_k J_k \end{pmatrix}, \begin{pmatrix} \beta_i\Delta t & \beta_i\Delta t \\ \beta_i\Delta t & \beta_i\Delta^- \end{pmatrix}\right\} \tag{12}$$

where $\Delta^- = (M-i)\Delta t$. We now condition (12) on $Y^M = y_M$ to give

$$\tilde{\pi}(Y_{i+1}|Y_i, y_M, J_{j+1:M}, V_{j+1:M}) = \\ \phi\left(Y_{i+1}\,;\, Y_i + \frac{y_M - Y_i}{M-i} + V_{i+1}J_{i+1} - \frac{\sum_{k=i+1}^{M} V_k J_k}{M-i}\,,\, \frac{M-i-1}{M-i}\beta_i\Delta t\right) \tag{13}$$

This density can then be sampled for $i = j, \ldots, M-2$ to give a skeleton bridge conditioned to start at y_j and finish at y_M and we use this construct as a proposal inside an MCMC step. Note that other proposal processes are possible — the Euler-Maruyama approximation could be used, however, it does not take into account future jumps or the fixed end point of the process.

3.1 An Empirical Example

As an illustrative example, consider the task of sampling the univariate jump diffusion

$$dY_t = (0.1 - 0.05Y_t)\,dt + 0.5\sqrt{Y_t}\,dW_t \; + V\,dN_t \tag{14}$$

conditioned on $Y_0 = 2$ and $Y_1 = 3$. Here $V \sim \exp(1.0)$ and N_t is a Poisson process with intensity 0.01. Following the notation of the preceding Section we split $[0,1]$ into $m+1$ time points $0 = t_0 < \ldots < t_m = 1$ with time step $\Delta t = 1/m$. An MCMC step is used and the jump times and sizes are then proposed by sampling $J^*_{i+1} \sim \mathrm{Bern}(0.01\Delta t)$ and $V^*_{i+1} \sim \exp(1.0)$ for $i = 0, \ldots m-1$. If the Euler scheme is used to propose $Y^*_1, \ldots, Y^*_{m-1}$ then the acceptance probability for a move to $Y^*_{1:m-1}, J^*_{1:m}, V^*_{1:m}$ reduces to

$$\min\left\{1,\, \frac{\pi(y_m|Y^*_{m-1}, J^*_m, V^*_m)}{\pi(y_m|Y_{m-1}, J_m, V_m)}\right\}.$$

If the bridging construct given by (13) is used, then the acceptance probability is

$$\min\left\{1,\, \frac{\prod_{i=0}^{m-1}\pi(Y^*_{i+1}|Y^*_i, J^*_{i+1}, V^*_{i+1})}{\prod_{i=0}^{m-1}\pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1})} \times \frac{\prod_{i=0}^{m-2}\tilde{\pi}(Y_{i+1}|Y_i, y_m, J_{j+1:m}, V_{j+1:m})}{\prod_{i=0}^{m-2}\tilde{\pi}(Y^*_{i+1}|Y^*_i, y_m, J^*_{j+1:m}, V^*_{j+1:m})}\right\}.$$

Figures 1 and 2 show 20 proposed paths generated from each scheme with $m = 5$ and $m = 50$. Empirical means and standard deviations for the acceptance probability of each scheme are tabulated for increasing m in Table 1.

[Figure 1 about here.]

[Figure 2 about here.]

[Table 1 about here.]

The Euler scheme is not conditioned on the end-point and consequently, as m is increased, we see a decrease in acceptance probability. The bridging scheme, however, is conditioned to finish at the end-point resulting in a satisfactory number of proposals being accepted.

4 Particle Filtering

4.1 The Simulation Filter

Recall the augmented data formalism of Section 2 so that data $D_j = (x_0, x_m, \dots, x_j)$, (where j is an integer multiple of m) arrive at times $t_0, t_m \dots, t_j$. Therefore, at time t_{j+m} (denoted t_M), new data x_M are accompanied by missing data $X_{j+1:M-1}$ corresponding to the observed component, $Z_{j+1:M}$ corresponding to the unobserved component, the jump times $J_{j+1:M}$ and jump sizes $V_{j+1:M}$. As each observation becomes available, interest lies in the online estimation of the unknown parameter vector Θ. When x_M is observed, assimilation of the information contained in x_M consists of generating a sample $\{(\Theta^{(s)}, Z_M^{(s)}), s = 1, \dots, S\}$ from the posterior $\pi(\Theta, Z_M|D_M)$, which is henceforth denoted by $\pi_M(\Theta, Z_M)$. This distribution can be found by formulating the posterior for parameters and all augmented data and then integrating out the latent data. Using (3),

$$\pi(Z_{j:M}, X_{j+1:M-1}, J_{j+1:M}, V_{j+1:M}, \Theta|D_M) \propto \pi_j(\Theta, Z_j) \prod_{i=j}^{M-1} \pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}, \Theta)\pi(V_{i+1}|J_{i+1}, \Theta)\pi(J_{i+1}|\Theta) \quad (15)$$

and we sample this density, for example, by MCMC and discard all components except Θ and Z_M to give a sample from the target posterior density. Note that since $\pi_j(\Theta, Z_j)$ has no analytic form, the particle filter recursively approximates $\Theta, Z_j|D_j$ by the swarm of points or particles $\{(\Theta_{(s)}, Z_j^{(s)}), s = 1, \dots, S\}$ with each $\Theta_{(s)}, Z_j^{(s)}$ having a discrete probablity mass of $w_j^{(s)} = 1/S$. It is assumed that as $S \to \infty$, the particles approximate the filtering density $\pi_j(\Theta, Z_j)$ increasingly well. As the filter treats the discrete support generated by the particles as the true (filtering) density, the simulation filter proceeds at time t_j by first selecting an integer, u, uniformly from the set $\{1, \dots, S\}$ and then drawing

$$(\Theta^*, Z_j^*)' \sim \mathrm{N}\{(\Theta^{(u)}, Z_j^{(u)})', h^2 B\} \quad (16)$$

where B is the Monte Carlo posterior variance and the overall scale of the kernel is a function of the smoothing parameter, h^2 usually dependent on the sample size, S. The effect of (16) is to replace $\pi_j(\Theta, Z_j)$ in (15) with its smooth kernel density form, and this step is included to avoid sample impoverishment (when only a few particles are propogated through each time point). Whilst this step is not entirely satisfactory (and indeed Liu & West (2001) argue that adding random noise can lead to overdispersed posteriors) it is found to work well for the examples considered here. Note that the choice of h^2 in (16) is equivalent to the choice of smoothing parameter in kernel density estimation and a trade off between under and over smoothing should therefore be made. Standard rules of thumb for calculating a suitable h^2 can be found in Silverman (1986).

Having proposed Θ^* and Z_j^*, the latent process in $(t_j, t_M]$ is updated as follows. For $i = j, \ldots, M-1$ draw the jump times $J_{i+1}^* \sim \pi(\cdot|\Theta^*)$ and the jump sizes $V_{i+1}^* \sim \pi(\cdot|J_{i+1}^*, \Theta^*)$. Draw Z_{i+1}^* recursively from the Euler transition density given by (6). That is, draw $Z_{i+1}^* \sim \pi(\cdot|Z_i^*, J_{i+1}^*, V_{i+1}^*, \Theta^*)$. Finally, draw X_{i+1}^* for $i = j, \ldots, M-2$ from $\tilde{\pi}(\cdot|Y_i^*, x_M, J_{j+1:M}^*, V_{j+1:M}^{x,*}, \Theta^*)$ given by

$$\phi\left(X_{i+1}^* \,;\, X_i^* + \frac{x_M - X_i^*}{M-i} + V_{i+1}^{x,*} J_{i+1}^* - \frac{\sum_{k=i+1}^{M} V_k^{x,*} J_k^*}{M-i} \,,\, \frac{M-i-1}{M-i} Z_i^* \Delta t\right)$$

A move to $X_{j+1:M-1}^*, Z_{j:M}^*, J_{j+1:M}^*, V_{j+1:M}^*, \Theta^*$ is accepted with probability

$$\min\left\{1\,,\, \frac{\prod_{i=j}^{M-1} \pi(X_{i+1}^*|Y_i^*, J_{i+1}^*, V_{i+1}^{x,*}, \Theta^*)}{\prod_{i=j}^{M-1} \pi(X_{i+1}|Y_i, J_{i+1}, V_{i+1}^{x}, \Theta)} \times \frac{\prod_{i=j}^{M-2} \tilde{\pi}(X_{i+1}|Y_i, x_M, J_{i+1:M}, V_{i+1:M}, \Theta)}{\prod_{i=j}^{M-2} \tilde{\pi}(X_{i+1}^*|Y_i^*, x_M, J_{i+1:M}^*, V_{i+1:M}^*, \Theta^*)}\right\} \tag{17}$$

Algorithmically, the simulation filter has the following form:

1. *Initialise* - Set $j = 0$. For $s = 1, \ldots, S$:
   - Draw $\Theta_{(s)} \sim \pi(\Theta)$ and $Z_{(s)}^0 \sim \pi(Z^0)$.
2. *MCMC* - Set $M = j + m$. For $s = 1, \ldots, S$:
   - Propose (Θ^*, Z_j^*) using (16). For $i = j, \ldots, M-1$:
     - Draw $J_{i+1}^* \sim \pi(\cdot|\Theta^*)$.
     - Draw $V_{i+1}^* \sim \pi(\cdot|J_{i+1}^*, \Theta^*)$.
     - Draw $Z_{i+1}^* \sim \pi(\cdot|Z_i^*, J_{i+1}^*, V_{i+1}^*, \Theta^*)$.
   - For $i = j, \ldots, M-2$:
     - Draw $X_{i+1}^* \sim \tilde{\pi}(\cdot|Y_i^*, x_M, J_{j+1:M}^*, V_{j+1:M}^{x,*}, \Theta^*)$.
   - Accept and store a move to $X_{j+1:M-1}^*, Z_{j:M}^*, J_{j+1:M}^*, V_{j+1:M}^*, \Theta^*$ with probability as in (17) otherwise store the current value of the chain.
3. *Pruning* - For $s = 1, \ldots, S$:
   - Discard all components except $(\Theta^{(s)}, Z_M^{(s)})$.
4. Set $j = j + m$ and return to step 2.

Note that further modifications may be made by thinning the MCMC output at the expense of running the sampler for longer; R iterations can be performed before thinning by a factor κ (such that $R = \kappa S$). This is done separately for each time point, with the final posterior sample of size S used as the prior for the next time point.

4.2 Particle Filtering with Sufficient Statistics

We now turn attention to the application of the algorithm of Storvik (2002) (see also Fearnhead (2002)) to the model given by (1). Note that the algorithm has been applied to discrete time Markov jump models, with jumps in returns only, by Johannes et al. (2006*a*). The algorithm requires that given $J_{1:j}, V_{1:j}, Y_{0:j}$ the distribution of Θ at time t_j is analytically tractable and in particular, depends on $J_{1:j}$, $V_{1:j}$ and $Y_{0:j}$ through some low dimensional sufficient statistics.

Denote the vector of sufficient statistics at time t_j by $T_j = T_j(J_{1:j}, V_{1:j}, Y_{0:j})$ and the distribution of Θ given T_j by $\pi(\Theta|T_j)$. For the model given by (1), the form of this distribution can be found in Appendix A. Now consider the distribution of all latent data and parameters given data D_M:

$$
\begin{aligned}
&\pi(Y_{0:M}\backslash\{D_M\}, J_{1:M}, V_{1:M}, \Theta|D_M) \\
&\quad\propto \pi(Y_{0:j}\backslash\{D_j\}, J_{1:j}, V_{1:j}|D_j)\,\pi(\Theta|Y_{0:j}, J_{1:j}, V_{1:j}) \\
&\qquad\times \prod_{i=j}^{M-1} \pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}, \Theta)\,\pi(V_{i+1}|J_{i+1}, \Theta)\pi(J_{i+1}|\Theta) \\
&\quad= \pi(Y_{0:j}\backslash\{D_j\}, J_{1:j}, V_{1:j}|D_j)\,\pi(\Theta|T_j) \\
&\qquad\times \prod_{i=j}^{M-1} \pi(Y_{i+1}|Y_i, J_{i+1}, V_{i+1}, \Theta)\,\pi(V_{i+1}|J_{i+1}, \Theta)\pi(J_{i+1}|\Theta)
\end{aligned}
\tag{18}
$$

An equally weighted particle representation of $\pi(Y_{0:j}\backslash\{D_j\}, J_{1:j}, V_{1:j}|D_j)$ is used, though only the sufficient statistics and the state of the unobserved volatility process need to be stored — that is $\{(T_j^{(s)}, Z_j^{(s)}), s = 1, \ldots, S\}$. Hence, the distribution in (18) is sampled by first selecting an integer, u, uniformly from the set $\{1, \ldots, S\}$ and then putting $T_j^* := T_j^{(u)}$ and $Z_j^* := Z_j^{(u)}$. A new Θ^* is then drawn from $\pi(\Theta|T_j^*)$ and the latent process in $(t_j, t_M]$ is proposed as in Section 4.1. A move to $X^*_{j+1:M-1}, Z^*_{j:M}, J^*_{j+1:M}, V^*_{j+1:M}, \Theta^*$ is accepted with probability given by (17) and the vector of sufficient statistics is updated accordingly. Algorithmically,

1. *Initialise* - Set $j = 0$. Initialise T_0 with the parameters indexing the prior, $\pi(\Theta)$. For $s = 1, \ldots, S$:
   - Draw $\Theta_{(s)} \sim \pi(\Theta)$ and $Z^0_{(s)} \sim \pi(Z^0)$.
2. *MCMC* - Set $M = j + m$. For $s = 1, \ldots, S$:
   - Sample an integer u from the set $\{1, \ldots, S\}$. Put $T_j^* := T_j^{(u)}$, $Z_j^* := Z_j^{(u)}$ and draw $\Theta^* \sim \pi(\cdot|T_j^*)$ using (20)–(25). For $i = j, \ldots, M-1$:
     - Draw $J^*_{i+1} \sim \pi(\cdot|\Theta^*)$.
     - Draw $V^*_{i+1} \sim \pi(\cdot|J^*_{i+1}, \Theta^*)$.
     - Draw $Z^*_{i+1} \sim \pi(\cdot|Z^*_i, J^*_{i+1}, V^*_{i+1}, \Theta^*)$.
   - For $i = j, \ldots, M-2$:
     - Draw $X^*_{i+1} \sim \tilde{\pi}(\cdot|Y^*_i, x_M, J^*_{j+1:M}, V^{x,*}_{j+1:M}, \Theta^*)$.
   - If the current state of the chain is $X_{j+1:M-1}, Z_{j:M}, J_{j+1:M}, V_{j+1:M}, \Theta$, accept and store a move to $X^*_{j+1:M-1}, Z^*_{j:M}, J^*_{j+1:M}, V^*_{j+1:M}, \Theta^*$ with probability as in (17) and put

     $$T_M^{(s)} = T(T_j^*, J^*_{j+1:M}, V^*_{j+1:M}, Y^*_{j+1:M}).$$

     Otherwise store the current value of the chain and put

     $$T_M^{(s)} = T(T_j, J_{j+1:M}, V_{j+1:M}, Y_{j+1:M}).$$
3. *Pruning* - For $s = 1, \ldots, S$:
   - Discard all components except $(\Theta^{(s)}, T_M^{(s)}, Z_M^{(s)})$.
4. Set $j = j + m$ and return to step 2.

As with the simulation filter, this algorithm can be modified by running the scheme for longer and thinning the output. Note also that although we store $\{\Theta^{(s)}, s = 1, \ldots, S\}$ at every iteration, the simulated values of Θ at time t_M do not depend on those simulated at time t_j — hence sample impoverishment can be avoided without the need to resort to ad-hoc methods such as the jittering approach discussed in Section 4.1.

5 Applications

5.1 Simulation Study

To validate the sequential estimation schemes of Section 4, evidence on the performance of the estimator of Θ in the SV model given by (1) is provided using synthetic data. Data were simulated from the model with $\mu = 0.08$, $\theta = 0.02$, $\kappa = -0.03$, $\sigma_z = 0.12$, $\lambda = 0.01$, $\mu_x = -3.1$, $\sigma_x = 2.7$ and $\mu_z = 1.7$ (calibrated to match the S&P data of Section 5.2). The Euler scheme was implemented twice with a sample interval of length 0.05— firstly every 20th point was recorded to give 1000 daily observations and secondly every 100th point was recorded to give 1000 weekly observations. Volatility paths were discarded leaving only observations on X_t.

We begin by repeating the experiments of Johannes et al. (2006*a*). The parameters are assumed to be known and the particle filter is run with $S = 5000$ particles to recover the unknown volatility path. The root-mean-squared error (RMSE) and mean-absolute error (MAE) between the filtered means and the true simulated volatilities are given in Table 2 for increasing m with daily and weekly data. Figure 3 shows filtered volatilities and true values for diferent values of m and daily data. A similar plot obtained using weekly data can be found in Figure 4.

[Table 2 about here.]

[Figure 3 about here.]

[Figure 4 about here.]

As expected, errors are larger when using the weekly data (as opposed to the daily data) as the Euler method does not approximate the underlying true transition density as well. Note that when using daily or weekly data, after an initial increase in accuracy when going from $m = 1$ (no latent points between each pair of observations) to $m = 5$, there is little to be gained from increasing m. An explanation can be found in Johannes et al. (2006*a*) — the results are sensitive to parameter values as non-normality varies with Θ and in particular with κ and σ_z. If the latter parameter values are small (as is the case here) then discretisation bias is small when using even $m = 1$.

Attention is now turned to the case of unknown parameters. The particle filter of Section 4.2 and the simulation filter of Section 4.1 are run with $S = 30,000$ particles and a thin of 150 (ie a total of 4.5×10^6 iterations) with $m = 1$, $m = 5$ and $m = 20$. Prior distributions are taken to be those given in Appendix A. Tables 3 and 4, and Figures 5- 7 summarise the posterior distribution obtained from the daily data. Similar results (not reported here) are obtained for the weekly data.

[Table 3 about here.]

[Table 4 about here.]

[Figure 5 about here.]

[Figure 6 about here.]

[Figure 7 about here.]

Tables 3 and 4 reveal substantive differences in the output of the particle filter and the simulation filter. For example, with $m = 5$, the particle filter gives respective posterior means of μ_x, σ_x and μ_z as -2.729, 2.235 and 2.094. The simulation filter, on the other hand, yields -3.851, 2.585 and 1.745. Since each estimate is consistent with the true value, we evaluate the accuracy of each sequential scheme by sampling the posterior distribution via full MCMC. That is, we run the global MCMC scheme of Eraker et al. (2003)with $m = 5$ and for 3 million iterations with a thin of 100. Smoothed densities from the output of the MCMC scheme are compared with the output of the particle filter in Figure 5 and the simulation filter in Figure 6. Whereas the output of the particle filter is consistent with that of the full MCMC scheme, there are notable inconsistencies between the output of the simulation filter and full MCMC. In particular, the simulation filter gives jump parameter posteriors that are over dispersed (compared to the 'truth') which may be a symptom of the jittering approach applied to each particle before propagation.

Figure 7 shows that both sequential algorithms recover the unknown volatility trace fairly accurately, despite having to integrate over the uncertainty associated with not knowing Θ. Filtered volatilities are provided for $m = 5$ only. Indeed, inspection of Table 3 suggests that there is little to be gained in accuracy by using a discretisation of $m > 5$ when daily data is used. This is consistent with the findings of Eraker et al. (2003), Johannes et al. (2006*a*) amongst others.

5.2 S&P 500 Data

In this Section, the particle filter is applied to daily observations of the S&P 500 index data, Jan. 3, 1986 - Jan. 3, 2000. Note that this corresponds to some 3539 observations and will therefore exacerbate any shortcomings of the sequential algorithm. The particle filter is run with $S = 30,000$ particles and a thin of 300 with parameter priors as in Appendix A. Note that σ_z is fixed to a value of 0.12 (as estimated using full MCMC) as it is well documented that the particle filter struggles to estimate the volatility of the volatility. Although this was not found to be the case when using simulated data, it does appear to be a problem when using the S&P 500 data. As the results of Section 5.1 suggest that there is little to be gained by using a discretisation of $m > 5$ for daily data, m is set to be 5. Posterior means and standard deviations are reported in Table 5 obtained from the output of the particle filter. Note that the output of the simulation filter is not reported — when using this method on the S&P data, propagated particles fail to adapt after the crash of 1987 and consequently, posteriors degenerate.

[Table 5 about here.]

[Figure 8 about here.]

[Figure 9 about here.]

Just as in Section 5.1, parameter posteriors obtained from the output of the particle filter are compared to those obtained by using full MCMC estimation. Inspection of Figure 8 reveals that both algorithms lead to parameter estimates that are consistent with one another although posterior samples of μ_x and σ_x from the particle filter may have degenerated slightly, suggesting that a longer run (i.e. a greater number of particles) is required to approximate the posterior sufficiently well.

Filtered volatilities are reported in Figure 9. Notice that we observe a jump in volatility on the day of the crash of 1987. Indeed, immediately prior to the crash, jump sizes were relatively small with respective posterior means for μ_x, σ_x and μ_z of -0.331, 2.054 and 1.955. Immediately after observing the crash, respective estimates become -19.547, 2.968 and 1.842.

6 Conclusions

Whereas sequential estimation for discrete time stochastic volatility models with jumps is reasonably well developed, little has been done regarding their continuous-time counterparts. In this paper, we have extended the sequential parameter estimation algorithm of Storvik (2002) to the continuous time stochastic volatility model with jumps in both returns and volatility. The simulation filter of Golightly & Wilkinson (2006*a*) was also considered, however, whilst both algorithms perform well on simulated data, when applied to the S&P 500 data, the simulation filter struggles to adapt to new information arriving after the Crash of 1987. Both algorithms rely on being able to sample the latent diffusion path between two observations conditional on the jump times and sizes. Whilst the Euler-Maruyama approximation can be used as a proposal process inside an MCMC step, it is found that the linear Gaussian construct used here is far more efficient.

It may be possible to improve the efficiency of the sequential algorithms considered here by including a 'look-ahead' step as used in the discrete time context by Johannes et al. (2006*a*) (see also Pitt & Shephard (1999)). It is less obvious how this step might be applied in the context of stochastic differential equations with jumps but remains the subject of ongoing research.

A The form of $\pi(\Theta|T_j)$

The particle filter of Section 4.2 requires that the distribution of Θ given data up to and including time t_j depends on the vector of sufficient statistics $T_j = T_j(J_{1:j}, V_{1:j}, Y_{0:j})$. Here, the form of this distribution is given for the model in (1). The following conjugate priors are adopted for each parameter:

$$\begin{aligned}
\mu &\sim \mathrm{N}(g_0, h_0^{-1}) \\
\psi = (\theta, \kappa)^T &\sim \mathrm{N}(\psi_0, \Phi_0^{-1}\sigma_z^2) \\
\sigma_z^2 &\sim \mathrm{IG}(c_0, d_0) \\
\lambda\Delta t &\sim \mathrm{Beta}(s_0, f_0) \\
\mu_x &\sim \mathrm{N}(m_0, k_0^{-1}\sigma_x) \\
\sigma_x &\sim \mathrm{IG}(a_0, b_0) \\
\mu_z &\sim \mathrm{G}(\alpha, \gamma)
\end{aligned}$$

for which the posterior conditionals are given by

$$\left(\mu\,|Y_{0:j}, J_{1:j}, V_{1:j}^x\right) \sim \mathrm{N}(g_j, h_j^{-1}) \tag{19}$$

$$\left(\psi\,|Z_{0:j}, J_{1:j}, V_{1:j}^z, \sigma_z^2\right) \sim \mathrm{N}(\psi_j, \Phi_j^{-1}\sigma_z^2) \tag{20}$$

$$\left(\sigma_z\,|Z_{0:j}, J_{1:j}, V_{1:j}^z\right) \sim \mathrm{IG}(c_j, d_j) \tag{21}$$

$$\left(\lambda\Delta t\,|J_{1:j}\right) \sim \mathrm{Beta}(s_j, f_j) \tag{22}$$

$$\left(\mu_x\,|J_{1:j}, V_{1:j}^x, \sigma_x^2\right) \sim \mathrm{N}(m_j, k_j^{-1}\sigma_x^2) \tag{23}$$

$$\left(\sigma_x^2\,|J_{1:j}, V_{1:j}^x\right) \sim \mathrm{IG}(a_j, b_j) \tag{24}$$

$$\left(\mu_z\,|J_{1:j}, V_{1:j}^z\right) \sim \mathrm{G}(\alpha_j, \gamma_j) \tag{25}$$

where

$$
\begin{aligned}
h_j &= h_0 + \Delta t \sum_{i=0}^{j-1} Z_i^{-1}, \qquad g_j = h_j^{-1}\left[g_0 h_0 + \sum_{i=0}^{j-1}\left(\frac{X_{i+1} - X_i - J_{i+1}V_{i+1}^x}{Z_i}\right)\right] \\
\psi_j &= \Phi_j^{-1}(\Phi_0\psi_0 + H^T G), \qquad H = \begin{pmatrix} \sqrt{\frac{\Delta t}{Z_0}} & \sqrt{\Delta t Z_0} \\ \vdots & \vdots \\ \sqrt{\frac{\Delta t}{Z_{j-1}}} & \sqrt{\Delta t Z_{j-1}} \end{pmatrix}, G = \begin{pmatrix} \frac{Z_1 - Z_0 - J_1 V_1^z}{\sqrt{\Delta t Z_0}} \\ \vdots \\ \frac{Z_j - Z_{j-1} - J_j V_j^z}{\sqrt{\Delta t Z_j}} \end{pmatrix} \\
\Phi_j &= \Phi_0 + H^T H \\
c_j &= c_0 + \frac{j}{2}, \qquad d_j = d_0 + \frac{1}{2}(\psi_0^T \Phi_0 \psi_0 + G^T G - \psi_j^T \Phi_j \psi_j) \\
s_j &= s_0 + \sum_{i=0}^{j-1} J_{i+1}, \qquad f_j = f_0 + j - \sum_{i=0}^{j-1} J_{i+1} \\
k_j &= k_0 + \sum_{i=0}^{j-1} J_{i+1}, \qquad m_j = k_j^{-1}\left(k_0 m_0 + \sum_{i=0}^{j-1} J_{i+1} V_{i+1}^x\right) \\
a_j &= a_0 + \frac{1}{2}\sum_{i=0}^{j-1} J_{i+1}, \qquad b_j = b_0 + \frac{1}{2}\left(k_0 m_0^2 + \sum_{i=0}^{j-1} J_{i+1}(V_{i+1}^x)^2 - k_j m_j^2\right) \\
\alpha_j &= \alpha_0 + \sum_{i=0}^{j-1} J_{i+1}, \qquad \gamma_j = \gamma_0 + \sum_{i=0}^{j-1} J_{i+1} V_{i+1}^z
\end{aligned}
$$

Note that for the applications in Section 5, the prior parameters used by Eraker et al. (2003) are adopted. That is, $g_0 = 1, h_0 = 0.04, \psi_0 = (0,0)', \Phi = \text{diag}\{1\}, c_0 = 2.5, d_0 = 0.1, s_0 = 2, f_0 = 40, m_0 = 0, k_0 = 0.1, a_0 = 5, b_0 = 20$.

References

Bakshi, G., Cao, C. & Chen, Z. (1997), 'Empirical performance of alternative option pricing models', *Journal of Finance* **52**(5), 2003–2049.

Bates, D. (1996), 'Jumps and stochastic volatility: Exchange rate processes implicit in deutsche mark options', *Review of Financial Studies* **9**(1), 69–107.

Del Moral, P., Jacod, J. & Protter, P. (2002), 'The Monte Carlo method for filtering with discrete-time observations', *Probability Theory and Related Fields* **120**, 346–368.

Doucet, A., Godsill, S. & Andrieu, C. (2000), 'On sequential Monte Carlo sampling methods for Bayesian filtering', *Statistics and Computing* **10**, 197–208.

Duffie, D., Singleton, K. & Pan, J. (2000), 'Transform analysis and asset pricing for affine jump-diffusions', *Econometrica* **68**, 1343–1376.

Eraker, B. (2004), 'Do stock prices and volatility jump?', *Journal of Finance* **59**(3), 1367–1403.

Eraker, B., Johannes, M. S. & Polson, N. G. (2003), 'The impact of jumps in volatility and returns', *Journal of Finance* **59**, 227–260.

Fearnhead, P. (2002), 'MCMC, sufficient statistics and particle filters', *Journal of Graphical Statistics* **11**, 848–862.

Geman, S. & Geman, D. (1984), 'Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images', *IEEE Transactions on Pattern Analysis and Machine Intelligence* **6**, 721–741.

Gilks, W. & Berzuini, C. (2001), 'Following a moving target - Monte Carlo inference for dynamic Bayesian models', *Journal of the Royal Statistical Society, Series B: Statistical Methodology* **63**, 127–146.

Golightly, A. & Wilkinson, D. J. (2006*a*), 'Bayesian sequential inference for nonlinear multivariate diffusions', *Statistics and Computing* **16**(4), 323–338.

Golightly, A. & Wilkinson, D. J. (2006*b*), 'Bayesian sequential inference for stochastic kinetic biochemical network models', *Journal of Computational Biology* **13**(3), 838–851.

Gordon, N. J., Salmond, D. J. & Smith, A. F. M. (1993), 'Novel approach to nonlinear/non-Gaussian Bayesian state estimation', *IEE Proceedings-F* **140**, 107–113.

Heston, S. (1993), 'A closed-form solution of options with Stochastic Volatility with applications to Bond and Currency options', *Review of Financial Studies* **6**, 327–343.

Johannes, M. S., Polson, N. G. & Stroud, J. R. (2006*a*), Optimal filtering of jump diffusions: Extracting latent states from asset prices.

Johannes, M. S., Polson, N. G. & Stroud, J. R. (2006*b*), 'Sequential parameter estimation in stochastic volatility models with jumps', *Working Paper* .

Liu, J., Longstaff, F. & Pan, J. (2001), 'Dynamic asset allocation with event risk', *Finance. Paper 2-01. http://repositories.cdlib.org/anderson/fin/2-01* .

Liu, J. & West, M. (2001), Combined parameter and state estimation in simulation-based filtering, *in* A. Doucet, N. de Freitas & N. Gordon, eds, 'Sequential Monte Carlo Methods in Practice', Springer-Verlag, New York.

Pan, J. (2002), 'The jump-risk premia implicit in options: Evidence from an integrated time-series study', *Journal of Financial Economics* **63**(1), 3–50.

Pedersen, A. (1995), 'A new approach to maximum likelihood estimation for stochastic differential equations based on discrete observations', *Scandinavian Journal of Statistics* **1995**(22), 55–71.

Pitt, M. K. & Shephard, N. (1999), 'Filtering via simulation: Auxiliary particle filters', *Journal of the American Statistical Association* **446**(94), 590–599.

Raggi, D. & Bordignon, S. (2006), 'Comparing stochastic volatility models through monte carlo simulations', *Computational Statistics and Data Analysis* **50**(7), 1678–1699.

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

Storvik, G. (2002), 'Particle filters for state-space models with the presence of unknown static parameters', *IEEE Transactions on Signal Processing* **50**(2), 281–289.

Tanner, M. A. & Wong, W. H. (1987), 'The calculation of posterior distributions by data augmentation', *Journal of the American Statistical Association* **82**(398), 528–540.

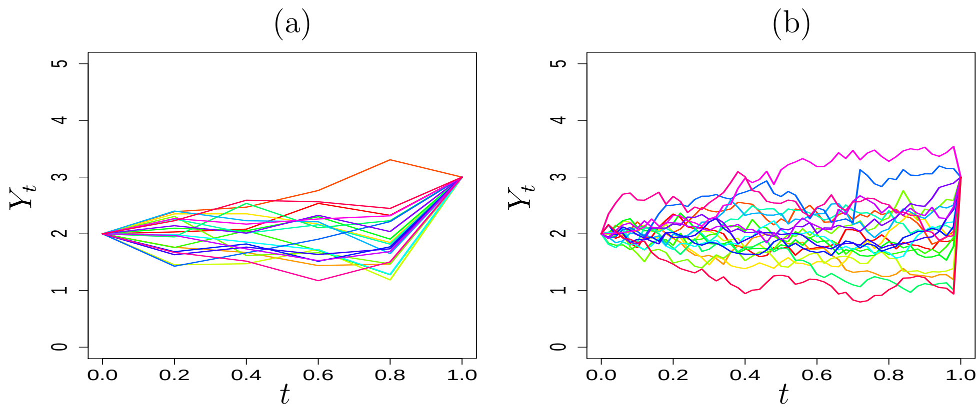

Figure 1: 20 *proposed* paths of the jump diffusion defined by (14) on $[0, 1]$, conditioned on $Y(0) = 2$ and $Y(1) = 3$, generated using the Euler scheme with (a) $m = 5$ and (b) $m = 50$.

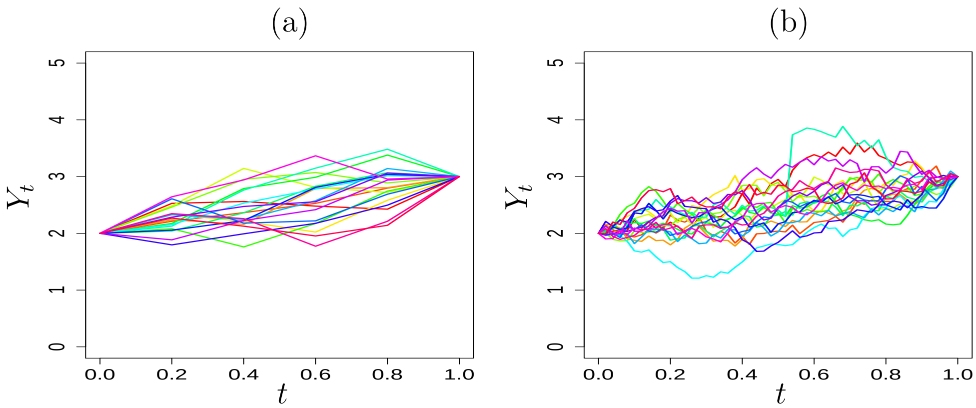

Figure 2: 20 *proposed* paths of the jump diffusion defined by (14) on $[0,1]$, conditioned on $Y(0) = 2$ and $Y(1) = 3$, generated using the diffusion bridge scheme with (a) $m = 5$ and (b) $m = 50$.

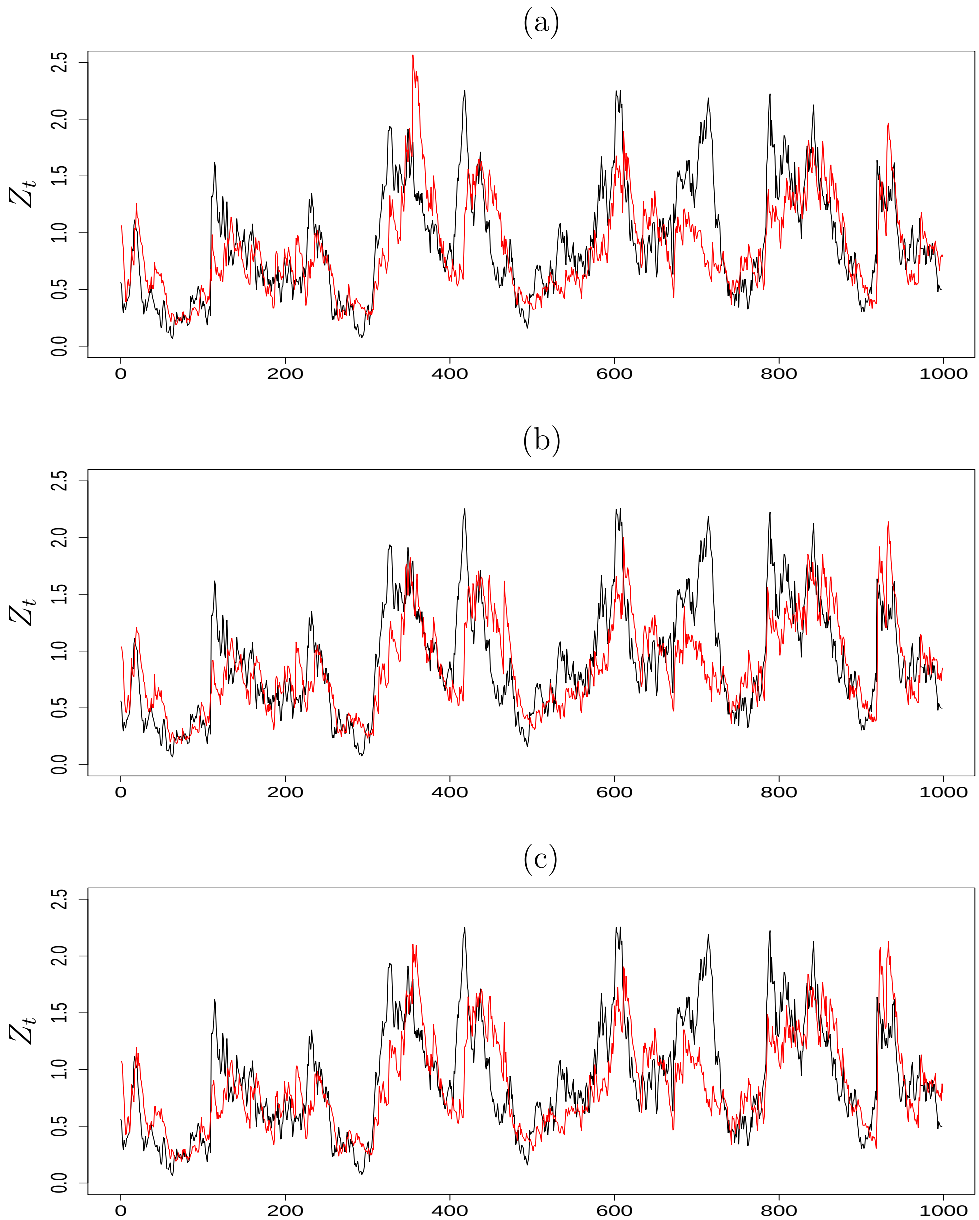

Figure 3: Filtered volatilities (red line) and true simulated volatilities (black line) from the output of the particle filter (with fixed parameters) using daily data and (a) $m = 5$, (b) $m = 5$ and (c) $m = 20$.

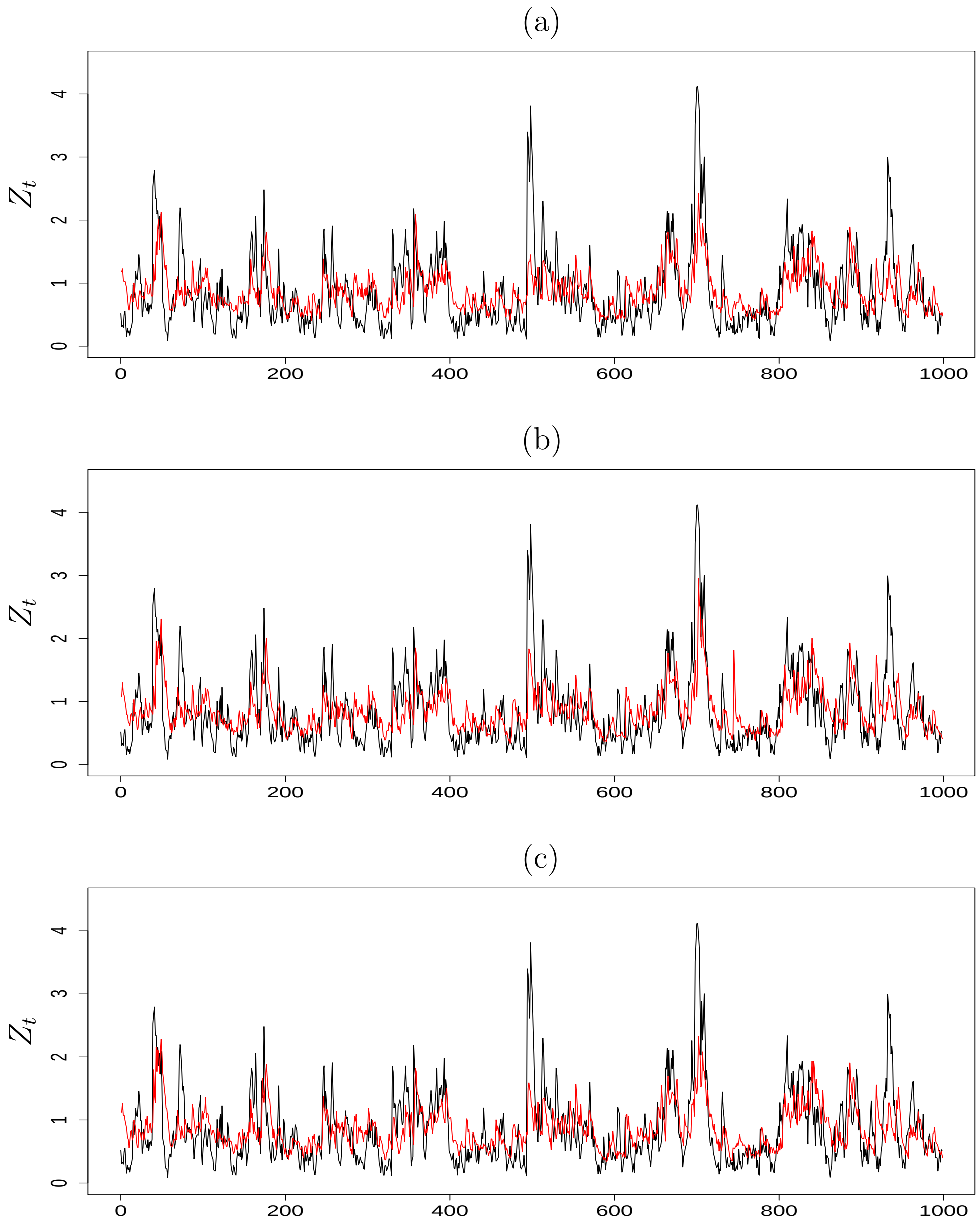

Figure 4: Filtered volatilities (red line) and true simulated volatilities (black line) from the output of the particle filter (with fixed parameters) using weekly data and (a) $m = 5$, (b) $m = 5$ and (c) $m = 20$.

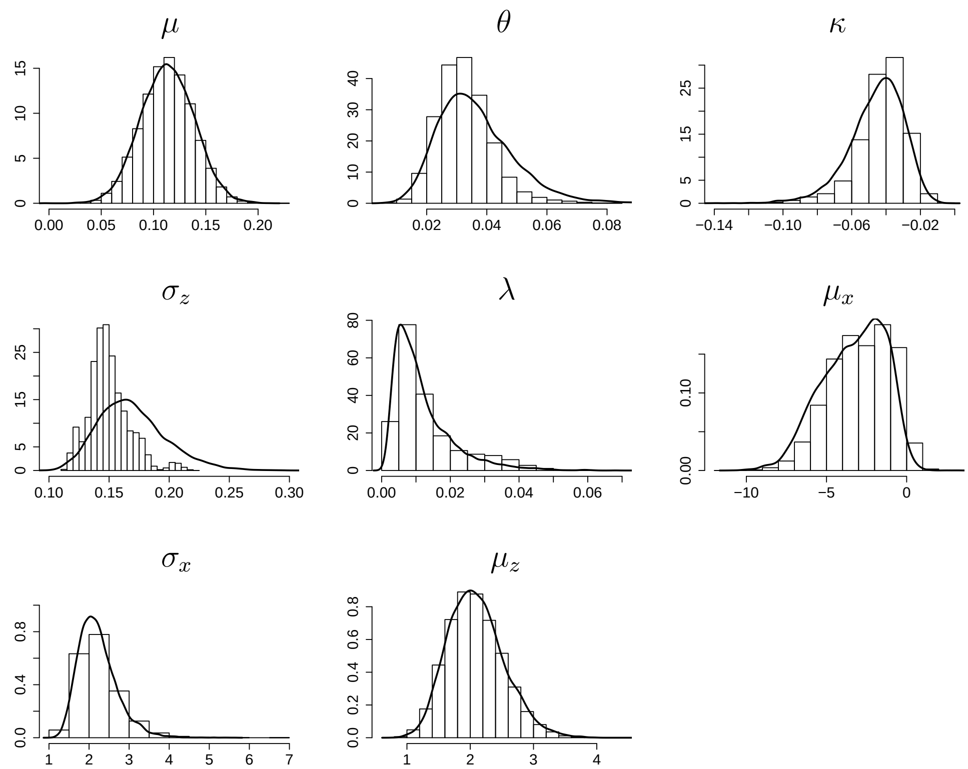

Figure 5: Parameter posteriors for the particle filter (histogram) and full MCMC (smoothed density) using 100 simulated daily observations and $m = 5$.

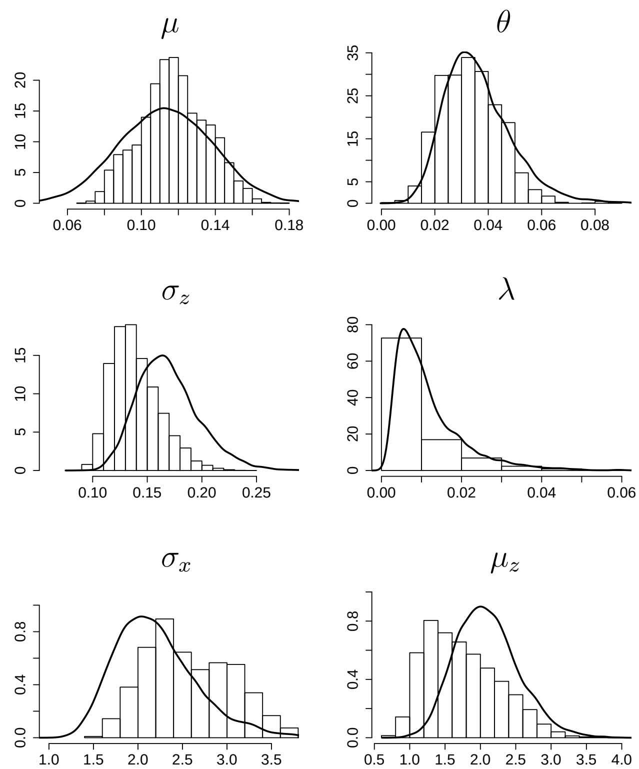

Figure 6: Parameter posteriors for the simulation filter (histogram) and full MCMC (smoothed density) using 100 simulated daily observations and $m = 5$.

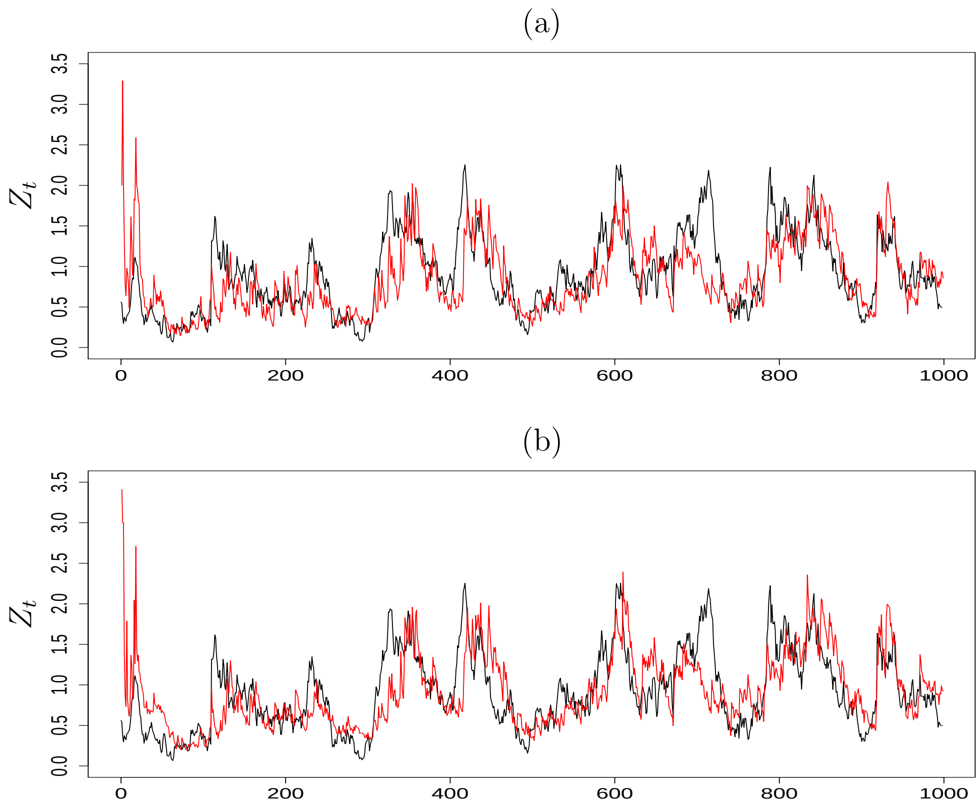

Figure 7: Filtered volatilities (red line) and true simulated volatilities (black line) from the output of (a) the particle filter and (b) the simulation filter. Both algorithms used simulated daily data with $m = 5$ and unknown parameters.

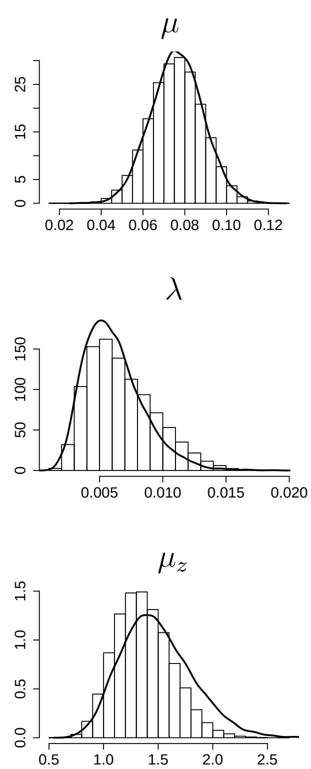

Figure 8: Parameter posteriors for the particle filter (histogram) and full MCMC (smoothed density) using daily observations on the S&P 500, Jan. 3, 1986 - Jan. 3, 2000.

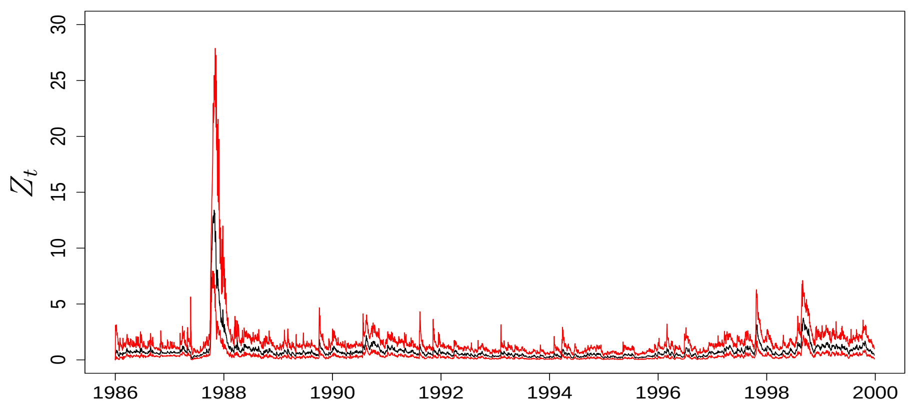

Figure 9: Filtered volatilities — 2.5 and 97.5 percentiles (red lines) and 50 percentiles (black line) from the output of the particle filter (using daily observations on the S&P 500, Jan. 3, 1986 - Jan. 3, 2000) with $m = 5$.

Scheme	Mean (standard deviation)				
	$m = 2$	$m = 5$	$m = 10$	$m = 50$	$m = 100$
Euler	0.421	0.259	0.164	0.074	0.055
	(0.058)	(0.049)	(0.040)	(0.030)	(0.024)
Bridging Construct	0.870	0.868	0.877	0.871	0.874
	(0.042)	(0.051)	(0.044)	(0.051)	(0.051)

Table 1: Empirical means and standard deviations for the acceptance probability of each scheme, based on 100 runs of 1000 MCMC iterations.

	$m = 1$	$m = 5$	$m = 20$
		Daily	
MAE	2.67	2.62	2.60
RMSE	3.62	3.54	3.53
		Weekly	
MSE	3.55	3.51	3.48
RMSE	4.82	4.77	4.77

Table 2: Mean-absolute and root-mean-squared errors between the filtering density (fixed parameters) and the true simulated volatilities (multiplied by 10) for increasing m.

	μ	θ	κ	σ_z	λ	μ_x	σ_x	μ_z
				True Values				
	0.08	0.02	-0.03	0.12	0.01	-3.1	2.7	1.7
				$m = 1$				
Mean	0.108	0.032	-0.042	0.159	0.011	-3.432	2.248	2.072
S.D.	(0.025)	(0.009)	(0.013)	(0.017)	(0.007)	(2.141)	(0.459)	(0.440)
				$m = 5$				
Mean	0.114	0.036	-0.048	0.169	0.016	-2.729	2.235	2.094
S.D.	(0.026)	(0.013)	(0.016)	(0.028)	(0.008)	(1.961)	(0.485)	(0.445)
				$m = 20$				
Mean	0.113	0.035	-0.046	0.166	0.012	-3.306	2.213	2.053
S.D.	(0.025)	(0.012)	(0.016)	(0.029)	(0.010)	(2.011)	(0.480)	(0.430)

Table 3: Posterior means and standard deviations for Θ (estimated on 1000 simulated daily observations), obtained from the output of the particle filter. Results are based on a single run of 4.5×10^6 iterations with a thin of 150.

	μ	θ	κ	σ_z	λ	μ_x	σ_x	μ_z
				True Values				
	0.08	0.02	-0.03	0.12	0.01	-3.1	2.7	1.7
				$m = 1$				
Mean	0.116	0.048	-0.051	0.170	0.013	-3.455	2.053	1.863
S.D.	(0.024)	(0.029)	(0.030)	0.039)	(0.007)	(2.160)	(0.506)	(0.554)
				$m = 5$				
Mean	0.121	0.033	-0.038	0.139	0.009	-3.851	2.585	1.745
S.D.	(0.017)	(0.011)	(0.011)	(0.022)	(0.009)	(2.571)	(0.488)	(0.517)
				$m = 20$				
Mean	0.082	0.053	-0.059	0.160	0.010	-3.651	2.541	2.136
S.D.	(0.014)	(0.016)	(0.017)	(0.055)	(0.004)	(2.295)	(0.508)	(0.200)

Table 4: Posterior means and standard deviations for Θ (estimated on 1000 simulated daily observations), obtained from the output of the simulation filter. Results are based on a single run of 4.5×10^6 iterations with a thin of 150.

	μ	θ	κ	λ	μ_x	σ_x	μ_z
Mean	0.076	0.018	-0.030	0.007	-3.175	2.595	1.489
S.D.	(0.013)	(0.002)	(0.004)	(0.003)	(0.812)	(0.321)	(0.260)

Table 5: Posterior means and standard deviations for Θ (estimated using daily observations on the S&P 500, Jan. 3, 1986 - Jan. 3, 2000), obtained from the output of the particle filter. Results are based on a single run of 9×10^6 iterations with a thin of 300.