JEL Classification: C11, C15, G1, G2
Keywords: Bayesian methods, MCMC, Particle filters, stochastic volatility, jumps

# Sequential Gibbs Particle Filter Algorithm with Applications to Stochastic Volatility and Jumps Estimation*

Jiří WITZANY - University of Economics in Prague, Faculty of Finance and Accounting, Prague, Czech Republic (jiri.witzany@vse.cz) *corresponding author*

Milan FIČURA - University of Economics in Prague, Faculty of Finance and Accounting, Prague, Czech Republic

*Abstract*

*The aim of this paper is to propose and test a novel Particle Filter method called Sequential Gibbs Particle Filter allowing to estimate complex latent state variable models with unknown parameters. The framework is applied to a stochastic volatility model with independent jumps in returns and volatility. The implementation is based on a new design of adapted proposal densities making convergence of the model relatively efficient as verified on a testing dataset. The empirical study applies the algorithm to estimate stochastic volatility with jumps in returns and volatility model based on the Prague stock exchange returns. The results indicate surprisingly weak jump in returns components and a relatively strong jump in volatility components with jumps in volatility appearing at the beginning of crisis periods.*

## 1. Introduction

Bayesian Markov Chain Monte Carlo (MCMC) and Particle Filter (PF) algorithms have become standard tools of financial econometrics specifically in connection with asset return stochastic volatility and jumps' modeling. The algorithms generalize the popular Kalman filter applicable to linear Gaussian state space models involving a latent state variable and possibly a vector of unknown parameters that need to be estimated based on a sequence of observed variables linked to the latent ones. The Kalman filter allows recursive filtering of the state space variables' (Gaussian) distributions given on-going observations. The state variables distributions can be also estimated (smoothed-out) based on the full set of observed variables. In addition, since the marginal likelihood of the parameters can be solved analytically, the vector of unknown parameters can be estimated by the likelihood maximization.

The Bayesian MCMC and PF algorithms can be applied to estimate latent variables and parameters of non-linear and non-Gaussian state space models. The idea of MCMC algorithms is to iteratively and consistently sample individual parameters and state variables (or their blocks) conditional on the rest of the parameters and the state variables. Under certain mild conditions the chain converges to the target distributions of the latent variables and the parameters conditional on the observed variables and the model specification (see e.g. Johannes M., Polson N., 2009 for an

*This research has been supported by the Czech Science Foundation Grant 18-05244S "Innovation Approaches to Credit Risk Management" and by the VSE institutional grant IP 100040.

overview). The PF algorithms introduced first in Gordon et al. (1993) aim to represent the latent state variables distributions empirically by sets of weighted values (particles) that are recursively updated based on new observations. The main advantage of the method is that it does not rely on any local linearization or other functional approximation. The price paid for this flexibility is computational cost, but with an increase of computational power and parallelization the method has become more-and-more popular (see e.g. Doucet, Johansen, 2009 or Speekenbrink, 2016 for an overview).

The aim of the paper is twofold. Firstly, we propose and test a novel PF method that we call Sequential Gibbs Particle Filter. We will demonstrate that the method outperforms in terms of efficiency a state-of-the-art recently published PF method (Fulop and Li, 2013). Secondly, in our empirical study we apply the algorithm to estimate a stochastic volatility model with jumps in returns and volatility based on the Prague stock exchange returns. The results will allow us to asses persistence of the stochastic volatility and the degree of presence of jumps in returns and volatility. We will be able to answer the question whether in the price process jumps in volatility play a more important than jumps in returns. The possible applications of the estimated model include dynamical Value at Risk estimation, volatility forecasting, or derivatives valuation.

The PF algorithms are relatively simple to implement if the model parameters are known but becomes challenging if the parameters are unknown. One possibility how to approach the problem of unknown model parameters is to treat them in the PF algorithm as latent variables and thus implicitly introduce to them certain stochastic dynamics (Gilks, Berzuini, 2001, Chopin, 2002, Andrieu et al., 2004, Carvahlo et al., 2010, or Speekenbrink, 2016). The problem of this approach is that the stochastic dynamics is not consistent with the original assumption of constant (yet unknown) model parameters and so the resulting estimates do not have to be consistent. Liu and West (2001) use a kernel density estimate of the parameter distribution, together with a shrinkage, in order to alleviate the problem. Alternatively, MCMC step can be used to re-sample the parameters (Gilks and Berzuini, 2001, Storvik, 2010, Fearnhead, 2002, Lopes et al., 2011). Nevertheless, as shown in Chopin et al. (2010), the parameter distribution will still suffer from degeneration, unless the past evolutions of the latent states are re-sampled as well, together with the parameters. Chopin et al. (2013) and Fulop et al. (2013) propose to approximate the Bayesian parameter distributions by particles and at the same time for each parameter vector to estimate the conditional latent state variable particles. The sequentially updated weights of the state variable values can be used to obtain marginal weights of the parameters' values. In this way, the two-dimensional particle filter structure can be propagated dynamically based on new observations. While the latent variable particles can be rejuvenated relatively frequently (or at each step) using the standard resample-move method, this is not possible for the parameter particles since there is no stochastic dynamics given by the model. In addition, sampling of new parameter values means recalculation of the conditional latent variable particle filter from the very beginning if we want to stay consistent with the model assumption. In order to limit the significant computational cost of the latent particles recalculation Fulop and Li (2013) propose to control for degeneracy of the particle filter, i.e. rejuvenate and recalculate the latent variable particle filter only if the degeneracy falls under certain threshold. The new parameters

are sampled in a Metropolis-Hasting accept-reject approach based on a proposal distribution, e.g. multivariate normal, fitted to the estimated posterior distribution. This means that the costly latent state variable particle recalculation step might have to be repeated several times before the parameter value is accepted. In addition, depending on the proposal distribution, the algorithm may easily get stuck in local maxima regions of the parameter space. The marginalized resample move Fulop, Li (2013) algorithm is then illustrated on real data for a Lévy jump stochastic volatility model and a structural credit risk model. In Fulop et al. (2014) the algorithm is applied to estimate a self-exciting asset pricing model that also takes into account co-jumps between prices and volatility.

Our proposed Sequential Gibbs Particle Filter (SGPF) algorithm follows the same two-dimensional parameter-latent variable particle filter structure as in Fulop, Li (2013) but rejuvenates the parameter particle by a Gibbs sampler conditional on sampled instances of the latent state variables. I.e., the algorithm samples a parameter given the marginalized posterior probabilities and a full history of the latent variable from the respective latent state particle. The Gibbs sampling conditional on the history of latent states is usually possible, in particular for stochastic volatility and jump models. In this way we save the costly accept-reject recalculations and at the same search the parameter space in a more consistent and efficient way. Our approach should not be confused with the concept of Particle Markov Chain Monte Carlo (PMCMC) or Particle Gibbs (PG) sampler from Andrieu et al. (2010) although the theoretical results can be applied also in our case. In Andrieu et al. (2010) the particle filters play the role of subcomponents of a full MCMC algorithm. That is, instead of standard resampling of the latent variables a PF is employed. It is then used to resample the parameters using an accept-reject approach or a Gibbs sampler, and then the PF is run again etc. In our case, the perspective is opposite, we run a full marginalized resample-move PF and use a Gibbs sampler to rejuvenate the parameter particle conditional on the posterior latent variable paths' distribution.

Asset return stochastic volatility and jump models are of major interest in financial econometrics due to their close relationship to market risk modeling and derivatives valuation. Since volatility and jumps themselves are not observable while the related asset returns are (and the models are typically non-linear and non-Gaussian) the Bayesian MCMC and PF models naturally come into consideration. The first break-through application of the Bayesian methods for the analysis of stochastic volatility models has been made in Jacquier et al. (1994). The authors applied an MCMC algorithm to estimate a stochastic volatility model on the US stock return data. The estimation method is shown to outperform classical estimation approaches such as the Method of Moments. Since then extensive research has confirmed viability of the MCMC and PF methods (see e.g. Pitt, Shephard, N., 1999, Shephard, 2004, Chronopoulou, Spiliopoulos, 2018, or Johannes, Polson, 2009 for an overview). A number of papers demonstrate importance of jumps in returns and volatility asset return dynamics modeling (Eraker et al., 2003, Eraker, 2004, Witzany, 2013) or Fičura, Witzany (2016) utilizing high-frequency data and the concept of realized volatility. Particle filters with an MCMC move to update the unknown parameters have been applied to stochastic volatility models with jumps by Johannes et al. (2002) or Raggi, Bordignon (2008). For approaches incorporating realized variance estimators into

stochastic volatility models setting see e.g. Takahashi et al. (2009), Fičura, Witzany (2017), or Maneesoonthorn et al. (2017).

The rest of this paper is organized as follows. In Section 2 we introduce the general state filtering problem, the basic particle filter method and our novel Sequential Gibbs Particle Filter algorithm. Then, after setting the stochastic volatility model with jumps in returns and volatility, we provide step-by-step details of the sampling algorithm, in particular focusing on adaptation of the proposal densities in order to make the filter more efficient. In Section 3 we firstly report results of the tests of the algorithm on artificially generated data and then apply it to real data from the Prague stock exchange. Finally, in Section 4 we summarize the results and conclude.

## 2. Methodology

### State Filtering Problem

A general state-space model can be written as:

$$\begin{aligned} y_t &= H(x_t, w_t, \theta) \\ x_t &= F(x_{t-1}, v_t, \theta) \end{aligned} \tag{1}$$

Where the observation $y_t$ is assumed to be conditionally independent on the hidden state $x_t$, $w_t$ and $v_t$ are mutually independent noises, and $\theta$ is a vector of static parameters. Density $p(y_t|x_t, \theta)$ is called the *observation density*, while density $p(x_t|x_{t-1}, \theta)$ is called the *transition density* of the Markov process of the hidden state with initial distribution $p(x_0|\theta)$.

The task of state filtering and parameter learning is to estimate:

$$p(x_t, \theta|y_{1:t}) = p(x_t|y_{1:t}, \theta)p(\theta|y_{1:t}). \tag{2}$$

### Particle Filter Algorithm with Known Parameters

For now we will focus on the state filtering problem, which is the estimation of $p(x_t|y_{1:t}, \theta)$ for all $t$ assuming that $\theta$ is given. Therefore, we will further omit $\theta$ in the notation.

Following the notation of Fulop, Li (2013) given $M$ particles $\left\{x_{t-1}^{(i)}; i = 1,2, \dots, M\right\}$ with weights $\widetilde{w}_{t-1}^{(i)}$ representing empirically the density $p(x_{t-1}|y_{1:t-1})$, we can approximate the density $p(x_t|y_{1:t})$ by drawing $x_t^i$ from a proposal density $g\left(x_t|x_{t-1}^i, y_t\right)$ and assigning importance weights to the sample:

$$w_t^{(i)} = \frac{p\left(y_t|x_t^i\right)p\left(x_t^i|x_{t-1}^i\right)}{g\left(x_t^i|x_{t-1}^i, y_t\right)}\widetilde{w}_{t-1}^{(i)}, \text{ for } i = 1, \dots, M, \tag{3}$$

which are then normalized by $\widetilde{w}_t^{(i)} = w_t^{(i)} / \sum_{j=1}^{M} w_t^{(i)}$.

The particles can be resampled at the end of every step or only time-to-time when the particle degenerates too much, i.e. when the effective sample size falls below certain threshold,

$$ESS = 1/\sum_{j=1}^{M}\left(\tilde{w}_t^{(i)}\right)^2 < ESS_{Thr}. \tag{4}$$

For a non-adapted filter not using the information given by the new observation $y_t$ the proposal density equals to the transition density $g\left(x_t^i|x_{t-1}^i, y_t\right) = p\left(x_t^i|x_{t-1}^i\right)$ and the weight update equation is thus simply:

$$w_t^{(i)} = p\left(y_t|x_t^i\right)\tilde{w}_{t-1}^{(i)}. \tag{5}$$

**Sequential Parameter Learning**

A possible approach to estimate the unknown parameters $\theta$ is to run the particle filter algorithm for an augmented state space variable $\langle x_t, \theta_t\rangle$ introducing a stochastic dynamics to the parameter vector $\theta$. A proposal density $g(\theta_t|\theta_{t-1})$ combined with the marginal likelihood $p(y_{1:t}|\theta)$ estimated by the particle filter can be used to sample a new $\theta_t$ using an accept-reject step. For example, the proposal density can be a simple random walk density $\theta_t \sim N(\theta_{t-1}, \Sigma)$ allowing the parameters to move to regions with higher marginal likelihood. However, as noted in Fulop, Li (2013), this approach does not necessarily lead to a successful solution due to the fact that the particle $\left\{x_{1:t}^{(i)}; i = 1,2,\dots,M\right\}$ has not been estimated with a static parameter vector $\theta$ leading to a possible inconsistency in the marginal likelihood estimation.

Further on, we elaborate the two-level particle filter proposed by Fullop, Li (2013) where we consider a set of parameter particles $\left\{\Theta_t^{(i)}; i = 1,2,\dots,M\right\}$ with normalized weights $\left\{\widetilde{W}_t^{(i)}; i = 1,2,\dots,M\right\}$ and, in addition, for each $\Theta_t^{(i)}$ a set of latent state particles $\left\{x_s^{(i,j)}; j = 1,2,\dots,N\right\}$ for $s = 1,\dots,t$ conditional on the same parameter vector $\Theta = \Theta_t^{(i)}$. We assume for simplicity that the latent particles are resampled at each step and so their weights need not be necessarily stored. However, before resampling of the latent states their weights can be used to update the parameter weights based on the following:

$$p(\Theta|y_{1:t}) = \int p(\Theta, x_{1:t}|y_{1:t})dx_{1:t} \tag{6}$$

and the recursive decomposition

$$\begin{aligned} p(\Theta, x_{1:t}|y_{1:t}) &= p(x_t|\Theta, x_{1:t-1}, y_{1:t})p(\Theta, x_{1:t-1}|y_{1:t-1}) \\ &\propto p(y_t|x_t, \Theta)p(x_t|x_{t-1}, \Theta)\, p(x_{1:t-1}|y_{1:t-1}, \Theta)p(\Theta|y_{1:t-1}). \end{aligned} \tag{7}$$

Therefore,

$$p(\Theta|y_{1:t}) = p(\Theta|y_{1:t-1}) \int p(y_t|x_t,\Theta)p(x_t|x_{t-1},\Theta)\, p(x_{1:t-1}|y_{1:t-1},\Theta)dx_{1:t}\,. \quad (8)$$

Note that the set $\left\{x_{1:t}^{(i,j)}; j = 1,2,\dots,N\right\}$ with the uniform normalized weights $w_{t-1}^{(i,j)} = \frac{1}{N}$ (due to resampling) represents the density proportional to $p(x_t|x_{t-1},\Theta)p(x_{1:t-1}|y_{1:t-1},\Theta)$ and the weights before normalization are $w_t^{(i,j)} = p\left(y_t|x_t^{(i,j)},\theta\right)\frac{1}{N}$. Hence, it follows that the parameter particle weights can be updated as follows:

$$\mathrm{W}_t^{(i)} = \widetilde{\mathrm{W}}_{t-1}^{(i)} \sum_j w_t^{(i,j)}. \quad (9)$$

As above, if the set of parameter particles degenerates too much, i.e. if

$$ESS = 1/\sum\nolimits_{j=1}^{M} \left(\widetilde{W}_t^{(i)}\right)^2 < ESS_{Thr} \quad (10)$$

where $\widetilde{W}_t^{(i)}$ are the parameter particle weights after normalization and $ESS_{Thr}$ is a threshold, the parameter particles need to be resampled. In this case we want to sample consistently a new set particles $\left\{\widehat{\Theta}_t^{(i)}; i = 1,2,\dots,M\right\}$ with initial equal weights $\widehat{\mathrm{W}}_t^{(i)} = 1/M$. Unfortunately, in order to be consistent for each $\Theta = \widehat{\Theta}_t^{(i)}$ the latent state particles $\left\{x_s^{(i,j)}; j = 1,2,\dots,N\right\}$ must be sampled again from the very beginning conditional on the new parameter vector $\Theta$ making the algorithm much more computationally demanding.

Resampling of $\Theta$ is based on the result of Del Moral (2004) according to which the likelihood $p(\Theta|y_{1:t})$ approximated by the particle filters is unbiased. Fullop, Li (2013) fit a multivariate normal distribution to the empirical distribution $\left\{\Theta_t^{(i)}; i = 1,2,\dots,M\right\}$ with normalized weights $\left\{\widetilde{\mathrm{W}}_t^{(i)}; i = 1,2,\dots,M\right\}$ (or to resampled equally weighted parameter particles) and sample from it proposals $\Theta_t^{*(i)}$.

The proposals are accepted based on the likelihood ratio $\widetilde{W}_t^{(i)}/\widetilde{W}_t^{*(i)}$ multiplied by the multivariate normal distribution likelihood ratio where $\widetilde{W}_t^{*(i)}$ is the proposed parameter vector normalized probability weight based on resampling of the latent state particles. The accept-reject algorithm (for $i = 1,2,\dots,M$) might be necessary to repeat more times if the acceptance ratio is too low making the algorithm even more computationally demanding.

**Sequential MCMC Particle Filter Algorithm**

Our algorithm is based on the fact that (under certain mild conditions) the particle filters with fixed parameters deliver unbiased estimates of the true density $p\left(x_{1:t}|y_{1:t},\Theta_t^{(i)}\right)$ and, according to Del Moral (2004), the likelihood $p(\Theta|y_{1:t})$

approximated by the particle filters is also unbiased. It follows that the empirical distribution obtained as a mix of the particle filters $\left\{x_{1:t}^{(i,j)}; j = 1,2, \dots, N\right\}$ with weights $\widetilde{W}_t^{(i)}$ is an unbiased approximation of the density $p(x_{1:t}|y_{1:t})$ unconditional on the parameters. Therefore, for $k = 1, \dots, M$ we can sample paths $x_{1:t}^{*(k)}$ from the mixed distribution and a new parameter value $\Theta_t^{*(k)}$ from $p\left(\Theta|x_{1:t}^{*(k)}, y_{1:t}\right)$. Practically, we firstly sample a parameter block $i$ from the discrete probability distribution $\left\{i, \widetilde{W}_t^{(i)}\right\}$ and then a path from the equally weighted set of particles $\left\{x_{1:t}^{(i,j)}; j = 1,2, \dots, N\right\}$. It should be noted that the sequence $\left\{x_s^{(i,j)}; s = 1,2, \dots, t\right\}$ is not a path in the sense of the transition relationship $x_t = F(x_{t-1}, v_t, \theta)$ due to the effect of resampling. Following the notation of Andrieu et al. (2010) we need to store the indices $j_0 = A(i, j_1, s)$ representing the parent $x_{s-1}^{(i,j_0)}$ of $x_s^{(i,j_1)}$ where the index $j_0$ changed due to resampling. These variables allow us to keep track of the genealogy of the particle and reconstruct the ancestral lineage $\{B(i, j, s); s = 1,2, \dots, t\}$ given $B(i, j, t) = j$ and going backward by

$$B(i, j, s-1) = A(i, B(i, j, s), s) \text{ for } s = t, \dots, 2. \tag{11}$$

Thus, given $i$ we sample $j \in \{1, \dots, N\}$ and the path

$$x_{1:t}^{(i,j)} = \left\{x_s^{(i,B(i,j,s))}; s = 1,2, \dots, t\right\}. \tag{12}$$

The point is that the move can be usually, e.g. in case of stochastic volatility or stochastic volatility with jumps model, done using a Gibbs sampler. However, the MCMC step can be used even if a Gibbs sampler is not known for example using an accept-reject approach where we accept a newly proposed parameter or keep the old one. In any case, after sampling (and accepting) a new parameter $\Theta = \widehat{\Theta}_t^{(i)}$ we still have to resample the latent state particles $\left\{x_s^{(i,j)}; j = 1,2, \dots, N, s = 1, .. t\right\}$. The advantage of this parameter sampling approach is that it does not rely on an ad hoc parameter proposal distribution as in Fulop, Li (2013) and prevents repeating of computationally costly accept-reject rounds.

**Stochastic Volatility Model with Jumps in Returns and Volatility**

We are going to consider the stochastic volatility model with independent jumps in returns and volatility

$$\begin{aligned} y_t &= \sigma_t \varepsilon_t + Z_t J_t \\ h_t &= \alpha + \beta h_{t-1} + \gamma \varepsilon_{V,t} + Z V_t J V_t \end{aligned} \tag{13}$$

Where $\varepsilon_t \sim N(0,1)$; $\varepsilon_{V,t} \sim N(0,1)$; $h_t = \log(V_t)$; $V_t = \sigma_t^2$, $J_t \sim Bern(\lambda)$, $Z_t \sim N(\mu_J, \sigma_J)$, and in addition $JV_t \sim Bern\left(\lambda_{JV}\right)$, $ZV_t \sim N(\mu_{JV}, \sigma_{JV})$.

Here, the observed values $y_t$ represent a time series of log-returns of an asset with zero mean, i.e. net of a long-term mean return if needed. In order to implement the sequential Gibbs generally described above PF we need to specify sequential resampling of the state space variables $\boldsymbol{x}_t = (h_t, J_t, Z_t, JV_t, ZV_t)$ and Gibbs resampling of the parameters $\Theta = (\alpha, \beta, \gamma, \lambda, \mu_J, \sigma_J, \lambda_{JV}, \mu_{JV}, \sigma_{JV})$.

Given a path $\boldsymbol{x}_{1:t}$ based on an ancestral lineage defined above the Gibbs sampling is relatively standard, for details see e.g. Witzany (2013):

Sample $\lambda$ and $\lambda_{JV}$ from the posterior beta distribution given by $J_{1:t}$ and $JV_{1:t}$ and appropriate prior distributions.

Sample $\mu_J, \sigma_J$ from the posterior normal and inverse gamma distributions given $Z_{1:t}$ with wide suitable prior distributions. Note that here we use only those $Z_s$ for which the corresponding jump indicator $J_s = 1$.

Similarly, sample $\mu_{JV}, \sigma_{JV}$ from the posterior normal and inverse gamma distributions given $ZV_{1:t}$.

In order to resample the stochastic volatility process parameters $\alpha, \beta, \gamma$ we use the Bayesian linear regression model:

$$\widehat{\boldsymbol{\beta}} = (\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}\boldsymbol{z}, \hat{\boldsymbol{e}} = \boldsymbol{z} - \boldsymbol{X}\widehat{\boldsymbol{\beta}} \tag{14}$$

where $\boldsymbol{z}$ is the column vector $\{h_s - ZV_s JV_s; s = 2, \dots, t\}$ and $\boldsymbol{X}$ has two columns, first with ones and the second with the corresponding "explanatory" factors $\{h_{s-1}; s = 2, \dots, t\}$. Then

$$\begin{aligned} (\gamma^*)^2 &\propto IG\left(\frac{n-2}{2}, \frac{\hat{\boldsymbol{e}}'\hat{\boldsymbol{e}}}{2}\right), \\ (\alpha^*, \beta^*)' &\propto N\left(\widehat{\boldsymbol{\beta}}, (\gamma^*)^2(\boldsymbol{X}'\boldsymbol{X})^{-1}\right). \end{aligned} \tag{15}$$

As usual, the distributions can be multiplied with suitable conjugate prior distributions.

Regarding the latent state variables $\boldsymbol{x}_t$ sampled based on the particles $\boldsymbol{x}_{1:t-1}$ and a new observation $y_t$, in order to build an efficient PF algorithm, it is important to design proposal densities adapted to the information whenever possible. Given the jump in volatility indicator $JV_t$ and its size $ZV_t$, it is straightforward to resample the latent volatility from the normal distribution $p(h_t|h_{t-1}, JV_t, ZV_t)$ given by (13). Next, given $h_t$ it is relatively simple to adapt the jump in return occurrence $J_t$ proposal probability since the likelihood density of $y_t$ is normal conditional on $J_t$. Similarly, if $J_t$=1 the jump in return size can be Gibbs sampled from a normal distribution given by the first equation in (13). Unfortunately, we cannot use the same approach to adapt $JV_t, ZV_t$ since $h_t$ on the left hand side of the equation is itself latent and not observed.

**Adapted Jumps in Volatility**

The key idea of our novel approach is to adapt $ZV_t$ taking into account the observed realized log-variance $\log(y_t^2)$. Let us firstly assume there is no jump in return, $J_t = 0$. To obtain a consistent normal proposal $ZV_t \sim N(\mu_{Z,pr}, \sigma_{Z,pr})$ conditional on $JV_t = 1$ we can use the equation

$$\log(y_t^2) = hNJ_t + ZV_t + \gamma\varepsilon_{V,t} + \log(\varepsilon_t^2)\,, \tag{16}$$

where $hNJ_t = \alpha + \beta h_{t-1}$ and approximate $\log(\varepsilon_t^2)$ by $N(c_1, c_2^2)$ where $c_1 = -1.27, c_2 = 2.22$ (as $\varepsilon_t \sim N(0,1)$ ). Therefore $ZV_t$ can be proposed from the normal distribution

$$\begin{aligned}\varphi\left(ZV_t; \mu_{Z,pr}, \sigma_{Z,pr}\right) \\ \propto \varphi\Bigg(ZV_t; \log(y_t^2) - hNJ_t \\ - c_1, \sqrt{\gamma_t^2 + c_2^2}\Bigg)\varphi(ZV_t; \mu_{JV,t}, \sigma_{JV,t})\end{aligned} \tag{17}$$

Where

$$\begin{aligned}\mu_{Z,pr} = \frac{(\log(y_t^2) - hNJ_t - c_1)\sigma_{JV,t}^2 + \mu_{JV,t}(\gamma_t^2 + c_2^2)}{\sigma_{JV,t}^2 + \gamma_t^2 + c_2^2}, \\ \sigma_{Z,pr} = \frac{\sigma_{JV,t}\sqrt{\gamma_t^2 + c_2^2}}{\sqrt{\sigma_{JV,t}^2 + \gamma_t^2 + c_2^2}}.\end{aligned} \tag{18}$$

Now, we can adapt $JV_t$ by estimating the two probabilities

$$p(JV_t|y_t, h_{t-1}) \propto \int p(y_t\,|h_t)p(h_t|h_{t-1}, JV_t)dh_t \times p(JV_t) \tag{19}$$

for $JV_t = 0,1$. In fact, we can evaluate analytically the integral

$$p(JV_t|\log(y_t^2), h_{t-1}) \propto \int p(\log(y_t^2)\,|h_t)p(h_t|h_{t-1}, JV_t)dh_t \times p(JV_t) \tag{20}$$

using the approximation of $p(\log(y_t^2)\,|h_t)$ by a normal density with known parameters based on the $\log(y_t^2) = h_t + \log(\varepsilon_t^2)$. Since $p(h_t|h_{t-1}, JV_t)$ is also normal given $JV_t$, we can apply the following general identity:

**Lemma**[1]**:** $\int_{-\infty}^{+\infty} \varphi(x; \mu_1, \sigma_1)\,\varphi(x; \mu_2, \sigma_2)dx = \frac{1}{\sqrt{2\pi(\sigma_1^2+\sigma_2^2)}} \exp\left(\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)$.

[1] *Proof*: The product of two normal densities is proportional to a normal density: $\varphi(x; \mu_1, \sigma_1)\varphi(x; \mu_2, \sigma_2) = \frac{1}{2\pi\sigma_1\sigma_2}\varphi(x; \tilde{\mu}, \tilde{\sigma}) \exp\left(\frac{(\mu_1-\mu_2)^2}{2(\sigma_1^2+\sigma_2^2)}\right)\tilde{\sigma}\sqrt{2\pi}$, where $\tilde{\mu} = \frac{\mu_1\sigma_2^2+\mu_2\sigma_1^2}{\sigma_1^2+\sigma_2^2}$ and $\tilde{\sigma} = \frac{\sigma_1\sigma_2}{\sqrt{\sigma_1^2+\sigma_2^2}}$ . The lemma then follows from $\int_{-\infty}^{+\infty} \varphi(x; \tilde{\mu}, \tilde{\sigma})\,dx = 1$ .

Therefore, using the notation of the lemma, we can set $\mu_1 = \log(y_t^2) - c_1, \sigma_1 = c_2$, and $\mu_2 = \alpha + \beta h_{t-1}$, $\sigma_2 = \gamma$ if $JV_t = 0$, and $\mu_2 = \alpha + \beta h_{t-1} + \mu_{JV}$, $\sigma_2 = \sqrt{\gamma^2 + \sigma_{JV}^2}$ if , $JV_t = 1$.

So far, we have assumed $J_t = 0$. Provided that $J_t = 1$ we base our proposal the equation

$$\log\left(y_t - \mu_J\right)^2 = h_t + \log(\varepsilon_t^2), \tag{21}$$

where the jump in returns is estimated by its mean. Thus we again apply the lemma setting $\mu_1 = \log\left(y_t - \mu_J\right)^2 - c_1$, $\sigma_1 = c_2$, and $\mu_2 = \alpha + \beta h_{t-1}$, $\sigma_2 = \gamma$ if $JV_t = 0$, and $\mu_2 = \alpha + \beta h_{t-1} + \mu_{JV}, \sigma_2 = \sqrt{\gamma^2 + \sigma_{JV}^2}$ if , $JV_t = 1$.

To evaluate consistently the four proposal probabilities $q(JV_t, J_t)$ we have to take into account that we have been in fact replacing $p(y_t|h_t)$ by $p(\log(y_t^2)|h_t)$ or $p\left(\log\left(y_t - \mu_J\right)^2|h_t\right)$. Generally, if $y = y(x)$ the transformed density satisfies $p(x)dx = p(y)|dy|$ and so $p(x) = p(y)|\frac{dy}{dx}|$. In this case:

$$p(y_t|h_t) = p(\log(y_t^2)|h_t) \times 2/|y_t|, p(y_t|h_t) = p\left(\log\left(y_t - \mu_J\right)^2|h_t\right) \times \frac{2}{|y_t - \mu_J|}.$$

It means that we have to adjust the proposal adapted probabilities as follows:

$$q(JV_t, J_t = 0) = p(JV_t|\log(y_t^2), h_{t-1}, J_t = 0) \times \frac{1-\lambda}{|y_t|} \times p(JV_t),$$

$$q(JV_t, J_t = 1) = p\left(JV_t\middle|\log\left(y_t - \mu_J\right)^2, h_{t-1}, J_t = 1\right) \times \frac{\lambda}{|y_t - \mu_J|} \times p(JV_t).$$

Finally, the proposal jump in volatility probability is

$$\lambda_{JV}^* = \frac{q(1,0) + q(1,1)}{q(1,0) + q(1,1) + q(0,0) + q(0,1)} \tag{22}$$

and $JV_t$ is sampled from $Bern(\lambda_{JV}^*)$.

The jump in volatility size $ZV_t$ is sampled from the mixed normal density

$$g(ZV_t|h_{t-1}, y_t) = (1-\lambda)\varphi\left(ZV_t; \mu_{Z,pr}^0, \sigma_{Z,pr}^0\right) + \lambda\varphi\left(ZV_t; \mu_{Z,pr}^1, \sigma_{Z,pr}^1\right) \tag{23}$$

where $\mu_{Z,pr}^0, \sigma_{Z,pr}^0$ are given by (18) in case $JV_t = 0$ and analogously $\mu_{Z,pr}^1, \sigma_{Z,pr}^1$ for $JV_t = 1$.

**Adapted Jumps in Returns**

As noted above, the adaptation of jumps in returns is much easier compared to adaptation of jumps in volatility. If $J_t = 0$, then $Z_t$ and $y_t$ are independent, and thus

$$p(y_t|h_t, J_t = 0) = \varphi(y_t; \mu, \sigma_t). \tag{24}$$

If $J_t = 1$, then $y_t$ is the sum of two independent normally distributed variables with distributions $N(0, V_t)$ and $N(\mu_J, \sigma_J^2)$, and so

$$p(y_t|h_t, J_t = 1) = \varphi\left(y_t; \mu_J, \sqrt{\sigma_J^2 + V_t}\right). \tag{25}$$

Based on the relationship $p(J_t|h_t, \lambda_t, y_t) \propto p(y_t|h_t, J_t)p(J_t|\lambda_t)$ we can easily compute the normalizing constant, as $J_t$ is only binary. Therefore,

$$p(J_t|h_t, \lambda_t, y_t) \sim Bern(\lambda_t^*), \text{ where}$$

$$\lambda_t^* = \frac{\varphi\left(y_t; \mu_J, \sqrt{\sigma_J^2 + V_t}\right)\lambda_t}{\varphi\left(y_t; \mu_J, \sqrt{\sigma_J^2 + V_t}\right)\lambda_t + \varphi(y_t; 0, \sigma_t)(1 - \lambda_t)}. \tag{26}$$

Given $J_t = 0$ the jump size is Gibbs sampled from $Z_t \sim N(\mu_J, \sigma_J)$. If $J_t = 1$ then

$$p(Z_t|h_t, y_t, J_t = 1) \propto \varphi(y_t; Z_t, \sigma_t)\varphi\left(Z_t; \mu_J, \sigma_J\right) \tag{27}$$

and so $p(Z_t|h_t, y_t, J_t = 1) \sim \varphi\left(Z_t; \mu_J^*, \sigma_J^*\right)$, where $\mu_J^* = \frac{y_t\sigma_J^2 + \mu_J V_t}{\sigma_J^2 + V_t}$, $\sigma_J^* = \frac{\sigma_J \sigma_t}{\sqrt{\sigma_J^2 + V_t}}$.

Once the state variables are resampled the weight of the respective particle must be updated according to (3), i.e.

$$w_t = \frac{p\left(y_t|h_t, Z_t, J_t\right)p(Z_t)p(ZV_t)(\lambda_t)^{J_t}(1 - \lambda_t)^{1-J_t}\left(\lambda_{JV,t}\right)^{JV_t}\left(1 - \lambda_{JV,t}\right)^{1-JV_t}}{g(Z_t|h_t, y_t, J_t)g(ZV_t|h_{t-1}, y_t)(\lambda_t^*)^{J_t}(1 - \lambda_t^*)^{1-J_t}\left(\lambda_{JV,t}^*\right)^{JV_t}\left(1 - \lambda_{JV,t}^*\right)^{1-JV_t}}\tilde{w}_{t-1}. \tag{28}$$

**Prior Distributions**

We are going to use standard parameter conjugate prior distributions characterized by their approximate mean and standard deviations given in Table 1. The second column shows the initial uniform distributions from which the step zero parameter particle values are drawn. The relatively wide intervals correspond to known stock returns empirical results where jumps in returns are usually negative while jumps in volatility are positive. It is customary to report the long-term volatility parameter $Ltv = \alpha/(1 - \beta)$ transforming the stochastic volatility equation (13) into the mean-reverting form:

$$h_t - h_{t-1} = (1 - \beta)(Ltv - h_{t-1}) + \gamma\varepsilon_{V,t} + ZV_t JV_t. \tag{29}$$

For example, the annualized long-term volatility around 25% corresponds to $Ltv = -8.3$.

Besides the initial distribution, we do not use any prior distributions for $Ltv, \beta,$ and $\gamma$. The intensity of jumps and returns distributions are standard conjugate Beta

with parameters corresponding to the mean and standard deviation indicated in Table 1. The means of sizes of jumps in returns are conjugate normal with parameters corresponding to the wide regions where the values are expected. The variances of the jump size conjugate priors are the inverse gamma distributions again with parameters corresponding to the mean and standard deviation in the table. Note that we show the square roots of the parameters in order to indicate where $\sigma_J$ and $\sigma_{JV}$ are expected to lie.

**Table 1 Prior Distributions**

| *Parameter* | *Initial dist.* | *Prior dist.* | *Mean* | *Standard dev.* |
|---|---|---|---|---|
| $Ltv = \alpha/(1-\beta)$ | U[-10, -6] | - | - | - |
| $\beta$ | U[0.8, 0.995] | - | - | - |
| $\gamma$ | U[0.1, 0.3] | Non-informative | - | - |
| $\lambda$ | U[0.001, 0.1] | Beta | 5% | 2.2% |
| $\mu_J$ | U[-0.1, 0.02] | Normal | -5% | 10% |
| $\sigma_J$ | U[0.05, 0.1] | Inverse Gamma | 10% | 8% |
| $\lambda_{JV}$ | U[0.001, 0.1] | Beta | 5% | 2.2% |
| $\mu_{JV}$ | U[0.5, 1.5] | Normal | 1 | 0.5 |
| $\sigma_{JV}$ | U[0.2, 0.8] | Inverse Gamma | 1 | 0.85 |

## 3. Simulated Dataset Results

In order to test the sequential Gibbs PF algorithm described above we have simulated a return process following (13) and given the (true) parameters shown in Table 2 over 4000 (daily) periods. We have run the particle filter with the estimates and Bayesian 95% confidence intervals that are reported in Table 2. Figure 1 demonstrates the estimated latent log-variance (mean values from the first run) fitting very well the true log-variance. The size of the parameter particles was set to $M = 200$, the size of latent state particles to $N = 200$, and the effective sample size threshold to $ESS_{Thr} = 100$. The first parameter recalculation is allowed after 10 steps in order to avoid possible issues with matrix inversion during the Gibbs resampling. The number of periods $T = 4000$ corresponds to the length of the real world dataset we are going to analyze in Section 4 and the relatively small number of particles was set at $200 \times 200$ with respect to memory capacity and computational time limitations. Note that the algorithm still works with several very large latent state matrices of the size $40\,000 \times 4\,000$. [2]

The results shown in Table 2 are satisfactory since the true parameters do fall into the estimated 95% Bayes confidence intervals in all cases. The estimated mean values are based on the last 2000 periods (i.e., the first 2000 days are considered as a burnout period). It should be noted that the quantiles are obtained from the mixed estimated particle densities also over the last 2000 periods. For some parameters such as $\gamma$ and the jump intensities the wide confidence intervals indicate uncertainty of the parameter inference. Since, in the simulation, we know the true latent variables, we can estimate directly the sample parameters that may differ slightly from the true data generating parameters and should be, in fact, estimated by the algorithm in an ideal

[2] The algorithm has been implemented in Matlab and run in parallel on 16 Core i7-5960X 4.3 GHz CPUs/ 64GB RAM desktop computer. One run with 200x200 particles and 4 000 steps took around 40 minutes.

situation. Again, in all cases the parameters inferred from the sampled log-variances and returns belong to the estimated confidence intervals.

**Table 2 True, Sampled and Estimated Parameters (SGPF, 200x200 particles)**

| *Parameter* | *True value* | *Sampled value* | *Estimated value* | *95% confidence intervals* | |
|---|---|---|---|---|---|
| $\mu_J$ | -0.08 | -0.0754 | -0.0766 | -0.1072 | -0.0457 |
| $\sigma_J$ | 0.04 | 0.0369 | 0.0435 | 0.0306 | 0.0610 |
| $Ltv$ | -8 | -7.8043 | -7.9794 | -10.6834 | -5.1749 |
| $\beta$ | 0.98 | 0.9795 | 0.9742 | 0.9574 | 0.9895 |
| $\gamma$ | 0.2 | 0.1999 | 0.1811 | 0.1370 | 0.3054 |
| $\lambda$ | 0.06 | 0.0513 | 0.0529 | 0.0267 | 0.0834 |
| $\mu_{JV}$ | 1 | 1.0142 | 0.7865 | -0.2074 | 1.4259 |
| $\sigma_{JV}$ | 0.4 | 0.3880 | 0.4341 | 0.2914 | 0.7398 |
| $\lambda_{JV}$ | 0.04 | 0.0375 | 0.0611 | 0.0251 | 0.1044 |

Next, Figure 2 shows posterior (estimated) jump probabilities and mean sizes. The true values are plotted above the x-axis (light grey) and the estimated values below the x-axis (dark grey) with artificially set negative signs for the sake of a visual comparison. The algorithm appears to estimate jumps in returns quite well. In order to calculate the estimated probability and mean of jumps in volatility we have used 15 days lag perspective. As the algorithm can recognize a (positive) jump in volatility only after a period of sustained relatively higher realized volatility, it had difficulties in identifying jumps in volatility at the exact time of their occurrence, as shown in the last two plots in Figure 2. Nevertheless, a closer inspection reveals that true jumps in volatility are usually followed by several days with higher estimated jump probability, i.e. the algorithm recognizes the increased volatility level but is not able to identify exactly the day when it happened. In spite of that the filter has estimated the jump size in volatility distribution parameters according to Table 2 relatively well.

In order to test stability of the sequential algorithm (SGPF) and compare it to the Fulop-Li algorithm, both versions of the algorithm have been run independently ten-times for $M = 100,\ N = 100$ and $ESS_{Thr} = 50$, with convergence results shown in Figure 3 and in Table 4. The relatively large dispersion of the estimated values in the different runs (for both algorithms), e.g. for $Ltv$ or $\gamma$, corresponds well to the wide confidence intervals shown in Table 2. In terms of the deviations of the estimated parameters with respect to the true values, the two approaches provide comparable results. The efficiency and precision of the algorithms is comprehensively compared in Table 3 showing R2 of the (log)volatility estimates and the discrimination power of the jumps in returns and jumps in volatility estimates (Bayesian probabilities) measured by the Accuracy Ratio (AR). SGPF gives better results compared to Fulop-Li in terms of volatility R2 and jumps in volatility AR, and comparable performance in terms of jumps in returns AR. Most importantly, SGPF significantly outperforms the Fulop-Li algorithm in terms of computational efficiency (SGPF using only 46% of time needed by Fulop-Li). The inefficiency of the Fulop-Li algorithm is caused mainly by the decreasing probability of acceptance and increasing number of runs in the accept-reject step of the algorithm as shown in Figure 4.

To compare the computational efficiency over longer time horizons, we have run the SGPF and Fulop-Li algorithms on a simulated 8000-day time series. Figure 5 shows that the computational time of the Fulop-Li relative to the SGPF algorithm increases exponentially as the acceptance probability gradually approaches zero. The

two algorithms still provide comparable results with SGPF slightly outperforming Fulop-Li in terms of R2 and AR, but the computational time needed by Fulop-Li is more than 33-times the time required by SGPF over the 8000-day time horizon (Table 5). It is apparent that the Fulop-Li algorithm computational cost becomes prohibitive for longer time series while SGPF is still able to provide feasible results.

**Figure 1 Simulated (light grey) and Estimated (dark grey) Latent Log-Variance $h_t$ (left) and Variance $V_t$ (right)**

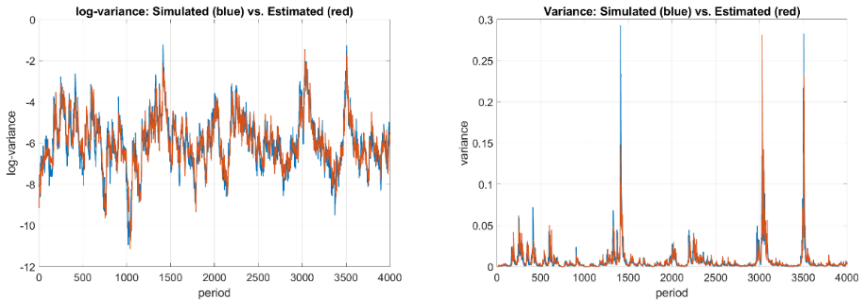



**Figure 2 Simulated (light grey) Versus Estimated Jumps (dark grey) in Returns and Volatility in Terms of Probability and Estimated Size**

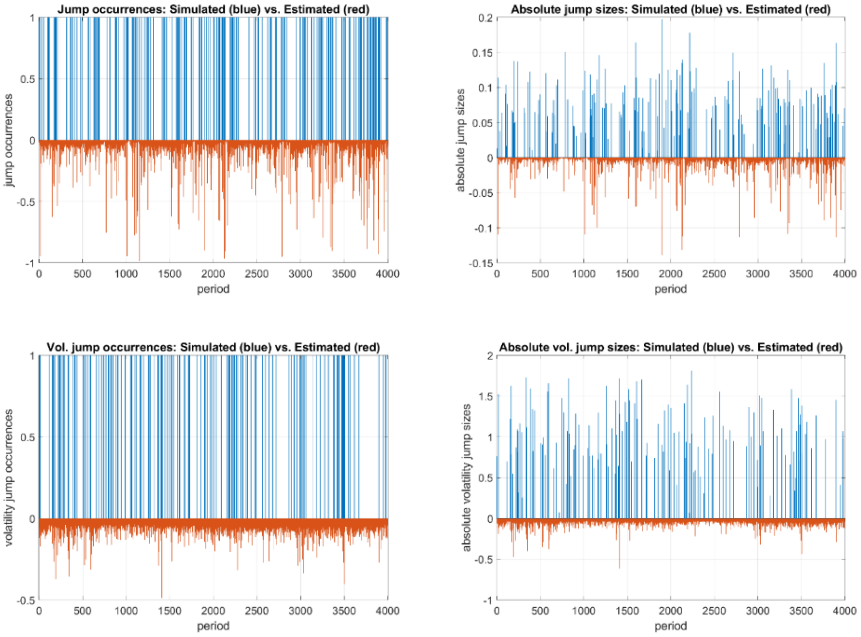
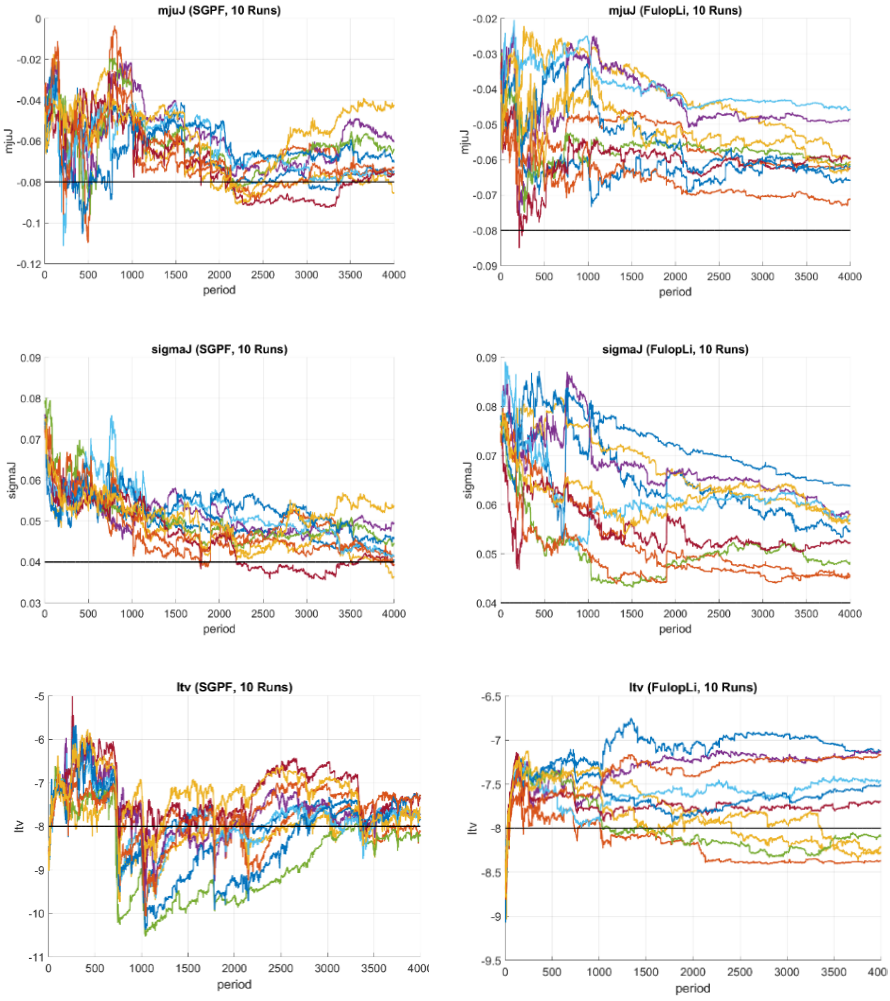
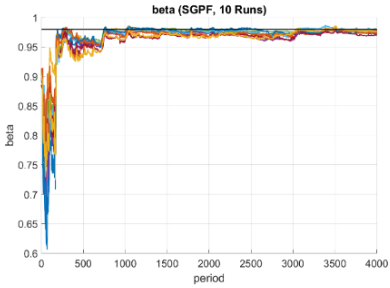
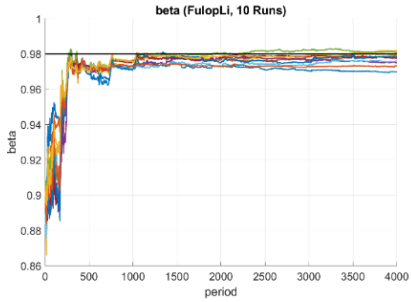
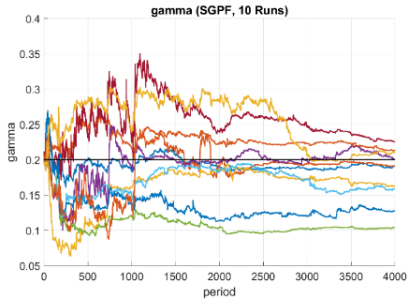


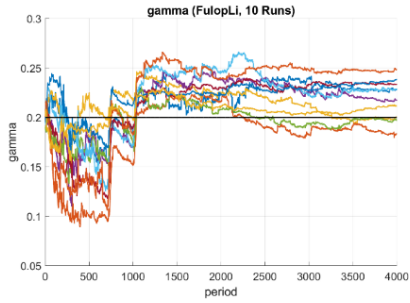
**Figure 3 Parameters Estimated by the Particle Filter Run Ten-Times (the horizontal black indicate the true values of the estimated parameters) (SGPF vs. Fulop-Li, 100x100 particles)**

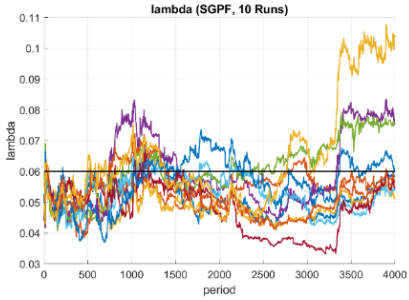


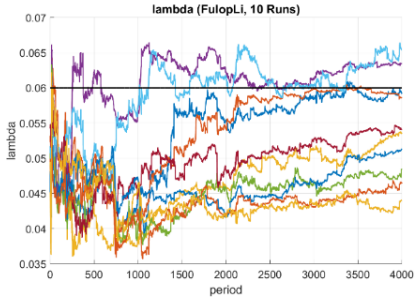

beta (SGPF, 10 Runs)
beta
period

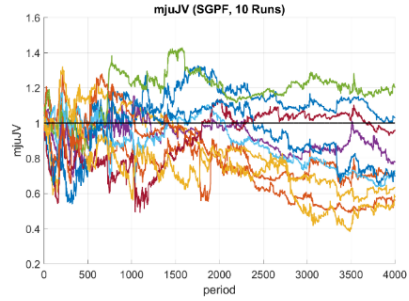

beta (FulopLi, 10 Runs)
beta
period

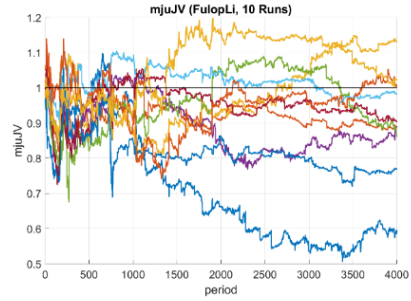

gamma (SGPF, 10 Runs)
gamma
period


gamma (FulopLi, 10 Runs)
gamma
period


lambda (SGPF, 10 Runs)
lambda
period


lambda (FulopLi, 10 Runs)
lambda
period


mjuJV (SGPF, 10 Runs)
mjuJV
period


mjuJV (FulopLi, 10 Runs)
mjuJV
period

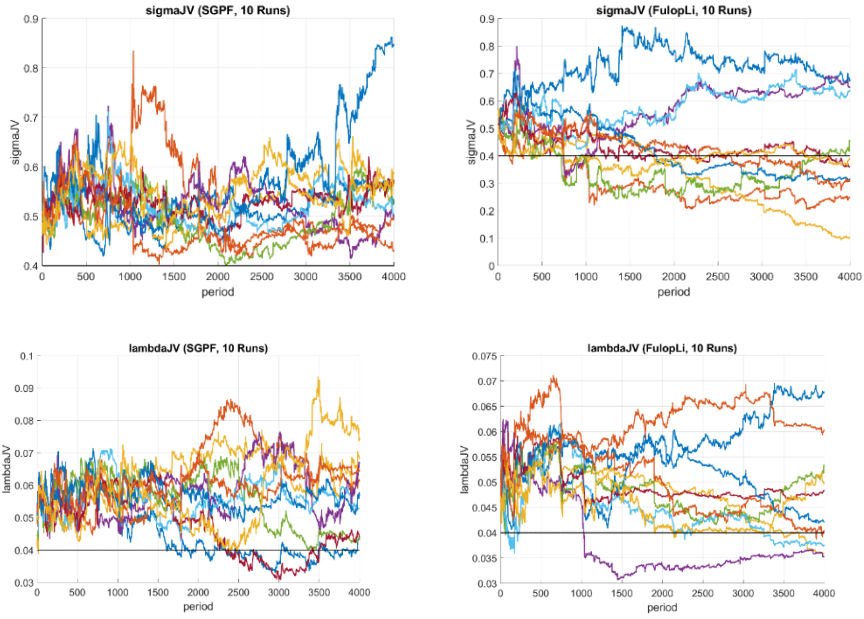



**Table 3 Average Performance Metrics of the Filtered Latent States by SGPF and Fulop-Li Algorithms (10 runs with 100x100 particles)**

| *Measure (average)* | *SGPF* | *FulopLi* | *Ratio (SGPF/FulopLi)* |
|---|---|---|---|
| Computational time | 878.76 s | 1906.22 s | 0.4610 |
| R2 (log-variance) | 0.7818 | 0.7822 | 0.9995 |
| R2 (variance) | 0.4695 | 0.4796 | 0.9789 |
| AR (jumps in ret.) | 0.6581 | 0.6565 | 1.0025 |
| AR (jumps in vol.) | 0.1980 | 0.2186 | 0.9057 |

**Table 4 Average Values and Standard Deviations of the Estimated Parameters by SGPF and Fulop-Li Algorithms (10 runs with 100x100 particles)**

| *Parameter* | *True value* | *Sampled value* | *avg(SGPF)* | *avg(FulopLi)* | *std(SGPF)* | *std(FulopLi)* |
|---|---|---|---|---|---|---|
| $\mu_J$ | -0.08 | -0.0754 | -0.0725 | -0.0572 | 0.0078 | 0.0079 |
| $\sigma_J$ | 0.04 | 0.0369 | 0.0463 | 0.0569 | 0.0033 | 0.0074 |
| $Ltv$ | -8 | -7.8043 | -7.7754 | -7.7051 | 0.4537 | 0.4635 |
| $\beta$ | 0.98 | 0.9795 | 0.9744 | 0.9767 | 0.0023 | 0.0031 |
| $\gamma$ | 0.2 | 0.1999 | 0.1860 | 0.2209 | 0.0460 | 0.0184 |
| $\lambda$ | 0.06 | 0.0513 | 0.0554 | 0.0524 | 0.0089 | 0.0072 |
| $\mu_{JV}$ | 1 | 1.0142 | 0.8500 | 0.9203 | 0.2080 | 0.1542 |
| $\sigma_{JV}$ | 0.4 | 0.3880 | 0.5244 | 0.4317 | 0.0518 | 0.1760 |
| $\lambda_{JV}$ | 0.04 | 0.0375 | 0.0565 | 0.0474 | 0.0107 | 0.0091 |

**Figure 4 Fulop-Li algorithm Accept-Reject Step Acceptance Rates and Numbers of Runs until 50% Acceptance in One Run of the Algorithm**

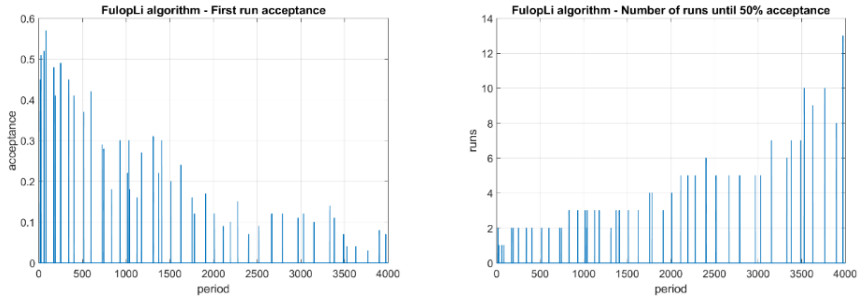
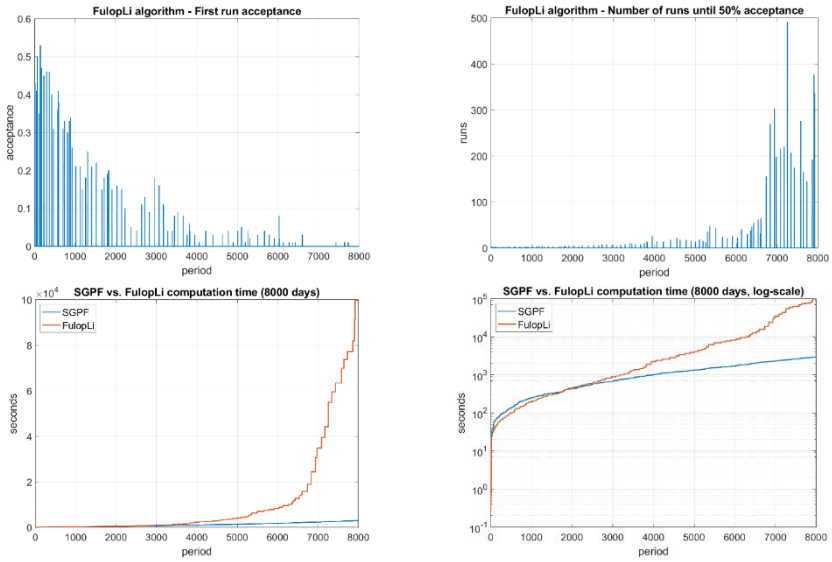



**Table 5 Performance Metrics of the Filtered Latent States by SGPF and Fulop-Li Algorithms (single run with 100x100 particles on a simulated time series of 8 000 days)**

| Measure | SGPF | FulopLi | Ratio (SGPF/FulopLi) |
|---|---|---|---|
| Computation time | 2948.38 s | 99723.89 s | 0.0296 |
| R2(log-variance) | 0.7842 | 0.7864 | 0.9972 |
| R2(variance) | 0.6546 | 0.6807 | 0.9616 |
| AR(jumps) | 0.6974 | 0.7107 | 0.9813 |
| AR(vol.jumps) | 0.3069 | 0.3185 | 0.9635 |

**Figure 5 Performance of the Fulop-Li Algorithm on an 8000-Days Long Simulated Time Series**



## 4. Prague PX Index Empirical Study

We have applied the SGPF algorithm to the Prague stock exchange index PF daily returns over the period 4.1.2002 – 25.04.2018, i.e. on a dataset with 4075 observations shown in Figure 6. We can notice the global financial crisis and the Eurozone crisis periods with increased volatility levels. The same figure shows the estimated daily volatilities obtained from the mean estimate latent log-variance $h_t$. The Sequential Gibbs PF algorithm was run with $M = 200$ and $N = 200$ particles, and with $ESS_{Thr} = 100$. The estimated mean parameters and the posterior confidence intervals are reported in Table 6. Since the parameter levels appear to stabilize after around 2000 steps of the algorithm (see Figure 7) we have set the first 2 000 days as the burnout periods and calculated the means and confidence interval based on the remaining 2075 estimates (Table 6). In order to verify robustness of the estimates we have also run the algorithm independently ten times with $100 \times 100$ particles as in the previous section. The results shown in Figure 9 are again consistent with the estimates given in Table 6.

Regarding the results, the long-term log-variance parameter $Ltv \cong -9.9$ corresponds to the annualized long-term volatility level around 11.2%. The volatility persistence parameter $\beta \cong 0.97$ corresponds well other studies (e.g. Eraker et al., 2003 or Witzany, 2013), while the estimated volatility of volatility parameter $\gamma \cong 0.15$ appears slightly lower probably due to the jump in volatility component. What comes as a rather surprising result is a very low estimate of the intensity of jump in returns parameter $\lambda \cong 1.2\%$ with the posterior 95% confidence interval (0.36%, 2.6%). In addition the mean size of the jumps of returns has been estimated at $\mu_J \cong -1.35\%$ not significantly different from zero (while jumps in stock returns are expected to be negative) and the standard deviation $\sigma_J \cong 3.3\%$ around five times the average daily volatility 0.74%. Our conclusion is that the jump in return component is quite weak just slightly symmetrically fattening the normal distribution tails. On the other hand, the jump in volatility component appears to be rather strong with the jump intensity $\lambda_{JV} \cong 2.9\%$, relatively large and significant mean jumps size $\mu_{JV} \cong 0.98$, and its standard deviation $\sigma_{JV} \cong 0.56$. It is also worth noting (Figure 8) that the jumps in returns are identified rather in the normal volatility periods while the jumps in volatility tend to appear at the beginning of crisis periods. In this case, we cannot show the true jump indicators as in Figure 2 but we do show the return series and the estimated log-variance series to visually locate possible jumps in returns and volatility.

**Figure 6 PX Index Daily Returns (light grey bars, 4.1.2002 – 25.04.2018) and the Volatility Estimated by the Particle Filter Algorithm, i.e. $\sigma_t = \exp(h_t/2)$.**

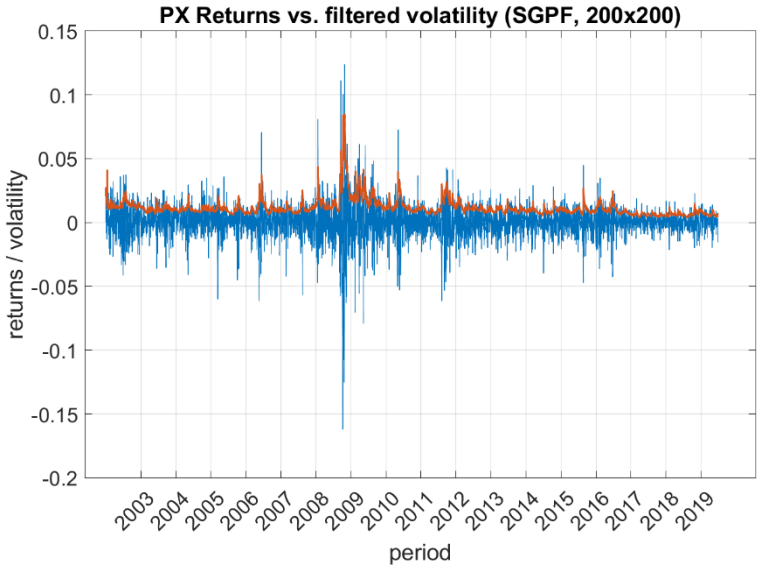
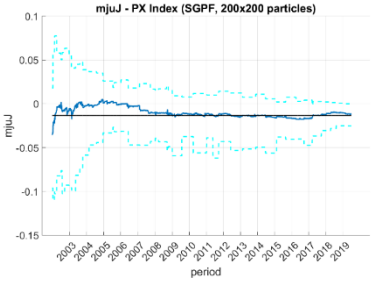
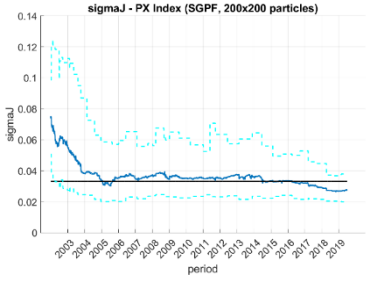


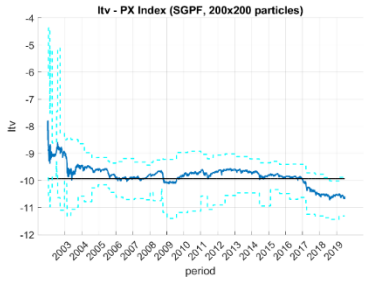

**Table 6 The Stochastic Volatility Model (13) Parameters Estimated for the PX Index Daily Returns Data**

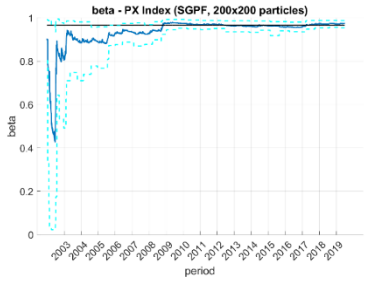
| *Parameter* | *Estimated value* | *95% confidence intervals* | |
|---|---|---|---|
| $\mu_J$ | -0.0135 | -0.0433 | 0.0079 |
| $\sigma_J$ | 0.0332 | 0.0220 | 0.0559 |
| $Ltv$ | -9.9328 | -11.1375 | -9.1716 |
| $\beta$ | 0.9661 | 0.9407 | 0.9876 |
| $\gamma$ | 0.1545 | 0.1344 | 0.1791 |
| $\lambda$ | 0.0122 | 0.0036 | 0.0260 |
| $\mu_{JV}$ | 0.9677 | 0.4806 | 1.5373 |
| $\sigma_{JV}$ | 0.4979 | 0.3147 | 0.8204 |
| $\lambda_{JV}$ | 0.0291 | 0.0113 | 0.0555 |

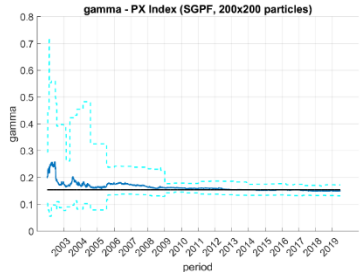
**Figure 7 Convergence of the Model Parameters and the 95% Confidence Intervals Estimated by the Particle Filter Where the Horizontal Black Lines Indicated the Estimated Mean Values**

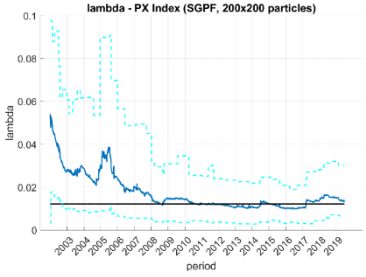


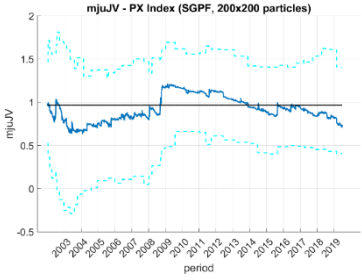



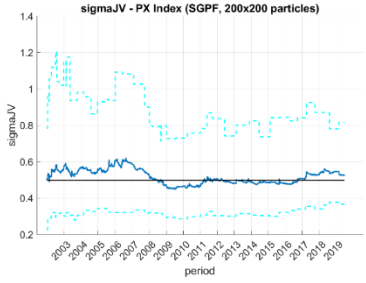

ltv - PX Index (SGPF, 200x200 particles)
ltv
period


beta - PX Index (SGPF, 200x200 particles)
beta
period


gamma - PX Index (SGPF, 200x200 particles)
gamma
period


lambda - PX Index (SGPF, 200x200 particles)
lambda
period


mjuJV - PX Index (SGPF, 200x200 particles)
mjuJV
period


sigmaJV - PX Index (SGPF, 200x200 particles)
sigmaJV
period

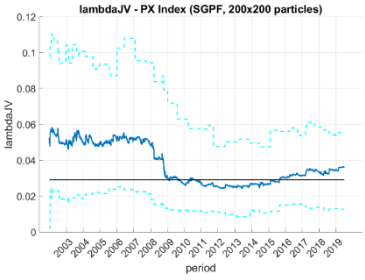

lambdaJV - PX Index (SGPF, 200x200 particles)
lambdaJV
period

**Figure 8 The Posterior Jumps in Returns and Volatility Probabilities and Sized**

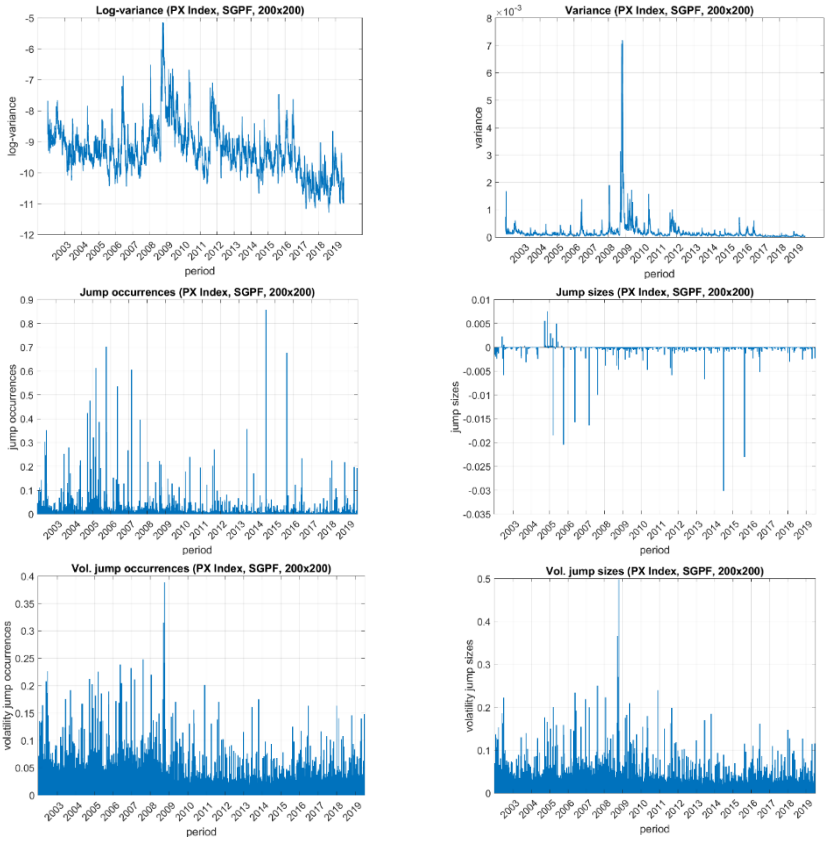
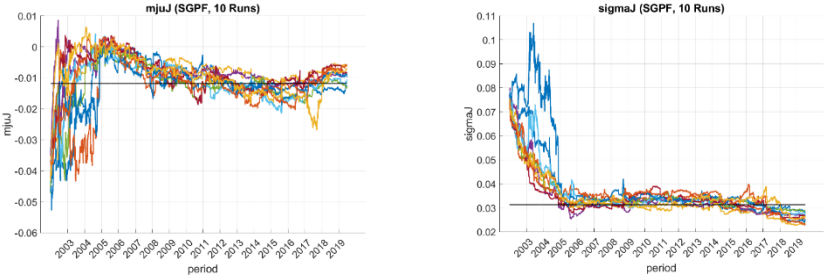
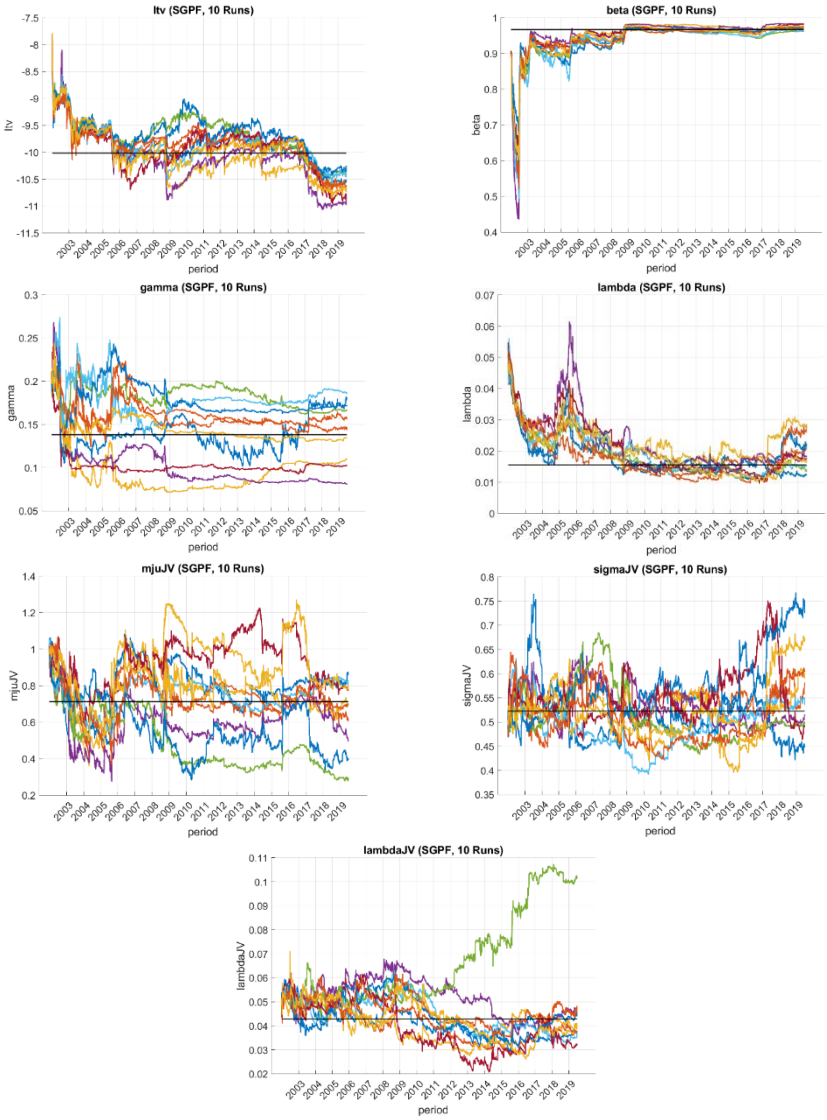



**Figure 9 Convergence of the Model Parameters Estimated by Ten Independent Runs of the Particle Filter (SGPF, 100x100 particles) Where the Black Lines Indicate the Mean Estimated Values**






## 5. Conclusions

We have proposed a new Sequential Gibbs Particle Filter algorithm allowing to estimate complicated latent state models with unknown parameters. The general framework has been applied to the stochastic volatility model with independent jumps in returns and in volatility. In order to make the algorithm more efficient in terms convergence we have designed adapted resampling steps whenever possible. The algorithm has been tested several times on an artificially generated datasets based on known true parameter with good results. The SGPF algorithm has been shown to outperform significantly the Fulop-Li algorithm in terms of computational efficiency.

Finally, we have applied the algorithm to a more than 16 years long time series of the Prague stock exchange index daily returns with some interesting results. Namely, identifying a very weak presence of jumps in returns while a strong presence of jumps in volatility taking place at the beginning of crisis periods.

Identification and timing of jumps in volatility seems to be the most serious weakness of the algorithm. In our opinion, this is caused rather by the fact that we are using only daily data and that it is impossible to identify a jump in volatility based just on one or a few observed daily returns with a higher magnitude. Therefore, we believe that, as a subject of further research performance, the algorithm can be improved by incorporating high frequency realized volatility data and possibly the leverage effect (in terms of both diffusion and jump components).

RERERENCES


Andrieu C, Doucet A, Holenstein R (2010): Particle Markov Chain Monte Carlo Methods. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 72(3):269-342.

Andrieu C, Doucet A, Singh SS, Tadic VB (2004): Particle Methods for Change Detection, System Identification, and Control. *Proceedings of the IEEE*, 92(3):423-438.

Carvalho C, Johannes M, Lopes, Polson N (2010): Particle Learning and Smoothing. *Statistical Science*, 25(1):88-106.

Chopin N (2002): A Sequential Particle Filter Method for Static Models. *Biometrika*, 89(3)539–551.

Chopin N, Iacobucci A, Marin JM, Mengersen KL, Robert CP, Ryder R, Schafer C (2010): On Particle Learning. *Ninth Valencia Meeting Discussion Paper*, Benidorm, Spain.

Chopin N, Jacob PE, Papaspiliopoulos O (2013): SMC2: An Efficient Algorithm for Sequential Analysis of State Space Models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75(3):397–426.

Chronopoulou, A, Spiliopoulos K (2018): Sequential Monte Carlo for Fractional Stochastic Volatility Models. *Quantitative Finance*, 18(3):507-517.

Del Moral P (2004): Feynman-Kac Formulae. In Del Moral P (eds): *Feynman-Kac Formulae: Genealogical and Interacting Particle Systems with Applications*, Springer, New York.

Doucet A, Johansen AM (2009): A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. *Handbook of Nonlinear Filtering*, 12(3):656-704.

Eraker B, Johannes M, Polson N (2003): The Impact of Jumps in Volatility and Returns. *The Journal of Finance*, 58(3):1269-1300.

Eraker B (2004): Do Stock Prices and Volatility Jump? Reconciling Evidence from Spot and Option Prices. *The Journal of Finance*, 59(3):1367-1403.

Fearnhead P (2002): Markov Chain Monte Carlo, Sufficient Statistics, and Particle Filters, *Journal of Computational and Graphical Statistics*, 11(4):848-862.

Ficura M, Witzany J (2016): Estimating Stochastic Volatility and Jumps Using High-Frequency Data and Bayesian Methods. *Finance a Uver*, 66(4):278.

Ficura M, Witzany J (2017): Estimation of SVJD Models with Bayesian Methods, Advanced Methods of Computational Finance, Prague: *Oeconomica*, p.161-204

Fulop A, Li J (2013): Efficient Learning Via Simulation: A Marginalized Resample-Move Approach. *Journal of Econometrics*, 176(2):146-161.

Fulop A, Li J, Yu J (2014): Self-Exciting Jumps, Learning, and Asset Pricing Implications. *The Review of Financial Studies*, 28(3):876-912.

Gilks WR, 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(1):127-146.

Gordon NJ, Salmond DJ, Smith AF (1993): Novel Approach to Nonlinear/Non-Gaussian Bayesian State Estimation. *IEE Proceedings F (Radar and Signal Processing)*, 140(2):107-113. IET Digital Library.

Jacquier E, Polson N, Rossi P: (1994): Bayesian Analysis of Stochastic Volatility Models. *Journal of Business & Economic Statistics*, 12(4):69-87.

Johannes M, Polson N, Stroud J (2002): Nonlinear Filtering of Stochastic Differential Equations with Jumps. *Working paper*, Graduate School of Business, University of Chicago.

Johannes M, Polson N (2009): MCMC Methods for Financial Econometrics. In: Ait-Sahalia Y, Hansen LP. *Handbook of Financial Econometrics 2*, Amsterdam, 1-72.

Liu J, West M (2001): Combined Parameter and State Estimation in Simulation-Based Filtering. In Doucet A, Freitas N, Gordon N (eds): *Sequential Monte Carlo Methods in Practice*, Springer, New York, 197–223.

Lopes H, Carvalho C, Johannes M, Polson N (2011): Particle Learning for Sequential Bayesian Computation, Bayesian Statistics 9. Oxford University Press, 317-360

Maneesoonthorn W, Forbes CS, Martin GM (2017): Inference on Self-Exciting Jumps in Prices and Volatility Using High-Frequency Measures. *Journal of Applied Econometrics*, 32(3):504-532.

Pitt MK, Shephard N (1999): Filtering Via Simulation: Auxiliary Particle Filters. *Journal of the American Statistical Association*, 94(446):590-599.

Raggi D, Bordignon S (2008): Sequential Monte Carlo Methods for Stochastic Volatility Models with Jumps. Preprint. URL: http://homes. stat. unipd. it/raggi.

Speekenbrink M (2016): A Tutorial on Particle Filters. *Journal of Mathematical Psychology*, 73:140-152.

Shephard N (2004): Stochastic Volatility: Selected Readings. Oxford University Press, Oxford.

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.

Takahashi M, Omori Y, Watanabe T (2009): Estimating Stochastic Volatility Models Using Daily Returns and Realized Volatility Simultaneously. *Computational Statistics & Data Analysis*, 53(6):2404-2426.

Witzany J (2013): Estimating Correlated Jumps and Stochastic Volatilities. *Prague Economic Papers*, 2:251.