Draft_Application_Stochastic_Volatility

1 Overview

In the following sections, we will try to replicate some of the analyses which were previously conducted on synthetic data, generated by a basic stochastic volatility model; this time, the same model specification and filters will be used on real data. However, as we will see, real data require different strategies to evaluate the effectiveness of several Sequential Monte Carlo techniques. Intuitively, such differences are due to the fact that now the latent process driving the Hidden Markov Model is not observed, nor do we know exactly its law of motion. This part of the analysis is structured as follows: section *fill with the number of the section* provides a review of recent research in the stochastic volatility literature, including extensions of the basic specification and the use of particle filters; section *fill with the number of the section* describes the data which we employ and the transformations that were applied to them. In section *fill with the number of the section* we briefly compare the series of filtered states that are obtained from different particle filters; moreover, several PFs are compared in terms of computational cost in approximating the exact filtering distribution implied by our SV model. Section *fill with the number of the section* features a forecast comparison between two SV specifications and a model that assumes constant mean and variance for the financial returns, namely a forecasting exercise that would not be possible were it not for particle filtering techniques. In section *fill with the number of the section* we conclude by employing a non-parametric measure of realised volatility as a proxy for the latent volatility process, allowing a RMSE and MAE comparison between filters which is similar to that of section *fill with the number of the section* for synthetic data.

2 SV models + Literary Review

In section *fill with the number of the section* a simple specification of a Stochastic Volatility model was presented and used as the data generating process of a simulated dataset. Denoting by y_t the log returns at time t and by x_t the latent volatility stochastic process, such state-space model was presented as follows,

$$y_t | x_t \sim N(0, e^{x_t})$$
$$x_t | x_{t-1}, \alpha, \beta, \tau^2 \sim N(\alpha + \beta x_{t-1}, \tau^2)$$

This specification - which we will employ in this section on a real dataset - can be deemed as the standard version of the SV model. Throughout the recent decades, several extensions have been proposed, mostly (but not exclusively) in the field of financial econometrics. On the one hand, for simplicity we will stick to the standard version, one that will suffice for the scope of this work on Sequential Monte Carlo methods. On the other hand, it is worth mentioning some of these extensions: indeed, one can rather safely assume that, through their higher degree of sophistication, such extensions might well improve on the performance of our standard SV model in the analyses that will follow.

Stochastic Volatility Models, whose early formulation is commonly attributed to Taylor (1982, 1986), allow to account for time-varying and autoregressive volatility in financial returns, posing themselves as a valid alternative to ARCH (Engle 1982) or GARCH (Bollerslev 1986) models in dealing with non-constant volatility. Kim, Shephard and Chib (1998) define the canonical model for regularly spaced data as

$$y_t = \psi e^{\frac{x_t}{2}} \epsilon_t$$
$$x_{t+1} = \mu + \beta(x_t - \mu) + \tau \eta_t$$
$$x_1 | \sigma, \beta \sim N\left(\mu, \frac{\sigma^2}{1 - \beta^2}\right)$$
$$\epsilon_t \sim N(0, 1)$$
$$\eta_t \sim N(0, 1)$$

where the log volatility x_t is assumed to follow a stationary process ($|\beta| < 1$), ψ is a constant scaling factor, β is the persistence in the volatility and τ is the volatility of the log-volatility.[1] Most importantly, ϵ_t and η_t, the Gaussian white noise processes that drive the canonical model, are assumed to be uncorrelated. This latter assumption justifies another definition of this specification, namely "discrete SV model without leverage".[2]

In order to accommodate for leverage effects, namely capture the increase in volatility that follows a drop in the returns, the model can extended as in Omori

---

[1] For identifiability reasons, either $\psi = 1$ or $\mu = 0$. Kim, Shephard and Chib (1998) prefer $\psi = 1$.

[2] Note that in this review we do not focus on continuous time SV models. In fact, such models have attracted a considerable amount of research in financial econometrics and mathematical finance, especially after Hull and White (1987) considered stochastic volatility for option pricing. Arguably, the most influential model was then proposed for option pricing by Heston (1993), a SV model with leverage effects and square root diffusion driving volatility. Diffusion-based SV models enjoyed increasing popularity, see for example Barndorff-Nielsen and Shephard (2001), or Christoffersen et al. (2010), who investigate alternatives to the entrenched affine square root SV model. Eraker (2004) proposed a SV model with correlated jumps in prices and volatility, extending Heston's model, while Comte and Renault (1998) extended Hull and White's model as to feature long memory properties. More recently, Gatheral et al. (2018) further built on Comte and Renault's fractional SV model to propose the popular "rough volatility models" (see also Friz et al., 2021).

et al. (2007),

$$y_t = e^{\frac{x_t}{2}} \epsilon_t$$
$$x_{t+1} = \mu + \beta(x_t - \mu) + \eta_t$$
$$\begin{pmatrix} \epsilon_t \\ \eta_t \end{pmatrix} \Big| \rho, \tau \stackrel{i.i.d.}{\sim} N_2(\mathbf{0}, \Sigma), \ \Sigma = \begin{pmatrix} 1 & \rho\tau \\ \rho\tau & \tau^2 \end{pmatrix}$$

where $\rho < 0$ captures the negative correlation. Such specification, which can be referred to as "discrete SV model with leverage", captures the asymmetric response of volatility to returns of different signs, so that similar specifications are sometimes also deemed as "asymmetric SV models" (e.g. Harvey and Shephard 1996, Mao et al. 2020).

Note that so far the assumptions on ϵ_t and η_t implied that the returns are conditionally normally distributed. The discrete time SV models can also be extended to allow for heavy-tailed or asymmetric conditional returns distributions: symmetric or skewed Student-t, Generalised Hyperbolic (GH) distribution, Generalised Error Distribution (GED) and scale mixtures of normals feature as popular choices (Kim and Stoffer 2008, Nakajima and Omori 2012, Mao et al. 2020). In fact, SV with heavy tailed return distributions were shown to better meet empirical regularities like the leptokurtic distribution of the returns and slowly decaying autocorrelation functions of the squared returns (Liesenfeld and Jung 2000).

Assuming that ϵ_t follows a Student-t distribution, and exploiting the fact that ϵ_t can then be written as $\lambda_t^{-1/2} \zeta_t$, where $\zeta_t \sim N(0,1)$ and $v\lambda_t \sim \chi_v^2$ (Harvey et al. 1994, Chib et al. 2002), we have the following SV model with both fat tails and leverage effect (Jacquier et al., 2004),

$$y_t = e^{\frac{x_t}{2}} \lambda_t^{-1/2} \zeta_t$$
$$x_{t+1} = \alpha + \beta x_t + \eta_t$$
$$\begin{pmatrix} \zeta_t \\ \eta_t \end{pmatrix} \Big| \rho, \tau \stackrel{i.i.d.}{\sim} N_2(\mathbf{0}, \Sigma), \ \Sigma = \begin{pmatrix} 1 & \rho\tau \\ \rho\tau & \tau^2 \end{pmatrix}$$
$$v\lambda_t \sim \chi_v^2$$

To capture other elements of the behaviour of financial data, several other extensions have been proposed. For example, SV models have been extended to include conditional heteroskedasticity in the mean returns (Koopman and Hol Uspensky 2002) to capture potential volatility feedback effects, or to feature autoregressive moving average innovations (Chan 2013, Zhang et al. 2020), allowing better goodness of fit and out-of-sample forecasts.[3] [4] The model

---

[3]Dimitrakopoulos and Kolossiatis (2020) note that "the moving average component, the leverage effect and the conditional heteroscedasticity in mean have been considered separately in the stochastic volatility literature" and provide two specifications, one featuring an MA component and leverage effects, the other an MA component and conditional heteroskedasticity in mean.

[4]Other extensions, though ones for which we do not report the specifications, feature

proposed by Koopman and Hol Uspensky is specified as

$$y_t = \nu_t + \psi e^{\frac{x_t}{2}} \epsilon_t$$
$$\nu_t = a + b y_{t-1} + d \psi^2 e^{x_t}$$
$$x_t = \beta x_{t-1} + \tau \eta_t$$
$$\begin{pmatrix} \epsilon_t \\ \eta_t \end{pmatrix} \overset{i.i.d.}{\sim} N_2 \left(\mathbf{0}, \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \right)$$

while the state space representation of the ARMA(p,q)-SV framework, as in Zhang et al. (2020), reads

$$y_t = \nu_t + \gamma_t$$
$$\gamma_t = \phi_1 \gamma_{t-1} + \dots + \phi_p \gamma_{t-p} + u_t + \varphi_1 u_{t-1} + \dots + \varphi_q u_{t-q}$$
$$u_t | x_t \sim N(0, e^{x_t})$$
$$x_t = x_{t-1} + \eta_t$$
$$\eta_t | \tau \sim N(0, \tau^2)$$

where the error terms u_t and η_t are independent across all leads and lags, while ν_t follows an unspecified time-varying process.

Interestingly enough, the path along which the SV models evolved coincides with that suggested by optimal portfolio findings. Johannes, Korteweg and Polson (2014) found that, in order to generate statistically significant portfolio improvements in a Bayesian learning problem, the model employed by the investor should incorporate both time-varying expected returns and stochastic volatility: indeed, either of these features alone did not lead to statistically significant gains with respect to employing models with time-constant expected returns and volatility.[5]

Finally, before moving to the application of Sequential Monte Carlo techniques to a SV model, we conclude by going through some references for the Bayesian analysis proposed for such models. Starting from the seminal work of Jacquier et al. (1994), the use of MCMC methods has become increasingly popular for parameter estimation and smoothing exercises in SV models (e.g. Kastner 2019, presenting the R package *stochvol* for Bayesian parameter estimation,

---

modelling the latent volatility process x_t as an ARFIMA process (Long Memory Stochastic Volatility model, Breidt et al. 1998) or as governed by a first-order Markov process (Markov Switching Stochastic Volatility model, So et al. 1998). Recently, Luo et al. (2018) incorporated neural networks in the stochastic volatility model (Neural Stochastic Volatility Model), while Xu and Chen (2021) employ deep learning models (Deep Stochastic Volatility Model).

[5]One could then argue that caution is needed when employing basic specifications of stochastic volatility models. For instance, Poon and Granger (2005) found that historical volatility and ARCH models both achieved better volatility forecasting performance than SV models. Similarly, Allen and McAleer (2020, see also Allen, 2020) found that, using realised volatility as benchmark, neither the canonical SV model or a GARCH(1,1) specification could forecast better than a simple form of historical volatility model.

and Chopin and Papaspiliopoulos 2020, who use MCMC to sample from the smoothing distribution of a SV model). As regards filtering exercises, the adoption of particle filters was rather rapid:[6] indeed, latent volatilities in Kim et al. (1998) were already filtered by employing the particle filter suggested in Pitt and Shephard (1999), paving the way for subsequent applications (*inter alia*) in Chib et al. (2006), Omori et al. (2007), Kim and Stoffer (2008) and Nakajima and Omori (2012).

3 Data Description

As previously mentioned, we will analyze the behaviour of the described filtering tools associated to a simple stochastic volatility model. In particular, we are evaluating the performance of such model on real data. For the observable process, namely financial returns in the SV model, we consider the continuously compounded daily returns (also called logarithmic returns) of three indices, S&P500, DOW JONES and STOXX50, in a time interval from June 1st, 2017 to May 30th, 2021. From these, we estimate the daily volatility, as computed by the model.

The indices have been selected as representatives of the global economic trends in the US and EU markets. Specifically, the S&P500 is a market-capitalization-weighted stock-price index tracing the performance of the 500 largest companies listed on US stock exchanges (NYSE and Nasdaq Exchange). The DOW JONES, instead, is a price-weighted stock-market index and accounts for the 30 major companies listed on US stock exchanges, characterized for being "blue-chip". Also the EURO STOXX 50 follows bluce-chip stocks representing leading firms in regions of the Eurozone.

The three plots below represent the time series of the log returns calculated for the three indices during our period of interest. In particular, each series is a sequence of daily observations representing the logarithm of the ratio between the closing price of the index for a given day and the closing price of the day before.

---

[6]Actually, SV models are now often used as straightforward applications of particle filters on non-linear state space models, see for example Andrieu et al. (2010), Douc et al. (2014) or Chopin and Papaspiliopoulos (2020).
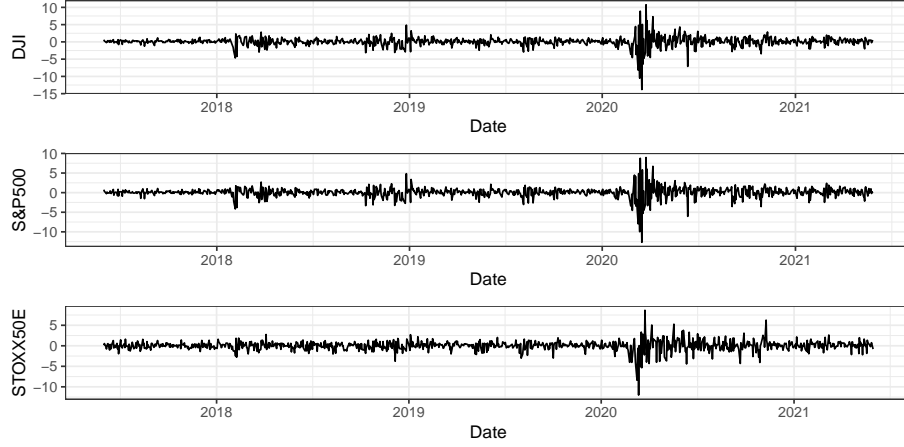
Figure 1: Close to close percentage log returns, 3 indices

At a first visual inspection, we can see a common behaviour in the volatility of all three indices. Especially, it is worth pointing out the similarity between the pattern of DJI and that of S&P500 (especially in terms of peaks), which both differ from the STOXX50E, presumably because the first two describe the US stock market, while the last index describes the European one. For example, a difference that can be seen by having a glance at the plots is that the US indices log returns displays differences in fluctuation magnitude across different periods that are more marked than for the EU index, for which log returns display, in general, wider fluctuations. Anyways, what stands out the most in all three plots is the very wide fluctuations present from February - March 2020 to around September 2020, which, very intuitively, are connected with the Covid-19 Pandemic Crisis.

Table 1: Summary statistics, close to close percentage log returns

	Dow Jones	S&P500	STOXX50E
Mean	0.05581	0.05002	0.01229
Standard Deviation	1.32564	1.38510	1.22951

As regards the series that we will employ in section *fill with the number of the section*, namely the series of realised volatilities which we will employ as a proxy for the true latent volatilities, the data were retrieved from the Oxford-Man Institute's Realized Library, which provides several different daily non-parametric measures of past volatility.[7] We consider a proper rescaling of the

---

[7]Such class of measures will be better presented in section *fill with the number of the section*, when reviewing the use of realised volatility estimators as proxies for latent volatilities.

rv5 series, *i.e.* model-free daily volatility estimates based on 5 min intraday return intervals. Also in this case, the chosen interval spanned from June 1st, 2017 to May 30th, 2021. The plots below represent the realized volatility in the period of interest for all the three indices.[8] As usual, they show a rather similar pattern across all three indices, identifying peaks corresponding to periods of higher uncertainty (in particular referring to the 2020 Covid-19 Crisis outbreak and its impact over time). Once more, there is great similarity between the series for the DJI and the one for the S&P500, while the series of the STOXX50E appears slightly different, in terms of peaks.
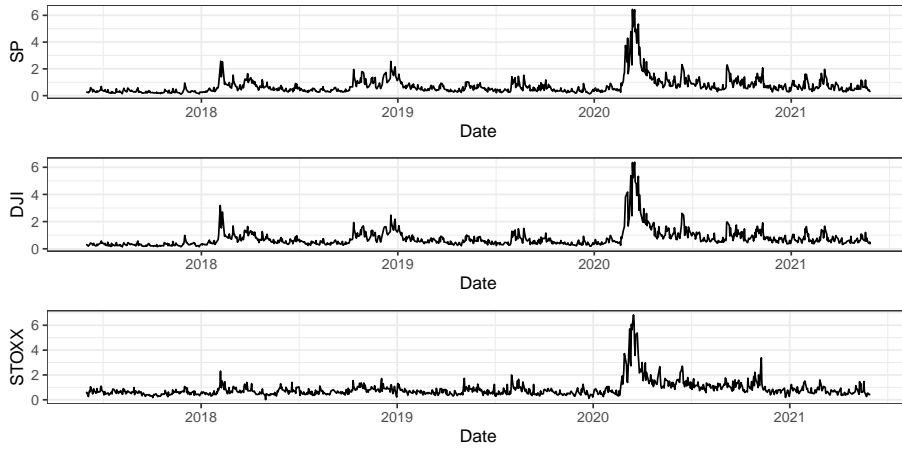

Figure 2: Rescaled realised daily volatility, 3 indices

3.1 Rescaling the data

When specifying the parameters of a stochastic volatility model, one should clarify whether the series of observed financial returns is one of log returns or of percentage log returns. Indeed, let y_t denote the returns at time t, so that a basic SV specification would prescribe $y_t|x_t \sim N(0, e^{x_t})$: clearly, for each period t, e^{x_t} needs to be higher to replicate the observed *percentage* log return rather than the observed log return. This implies that the latent volatility x should fluctuate around higher values when using percentage log returns as y_t. This feature, when adopting the aforementioned canonical SV model as defined in Kim et al. (1998), would be represented by a higher μ. For example, as noted in Kastner (2019), daily log returns often have a variance of 0.0001 or less, which implies that μ should lie somewhere around $\ln(0.0001) \approx -9$; instead, when considering daily percentage log returns, one then has a variance of 1, so that $\mu = \ln(1) = 0$. In other terms, one might end up with a basic specification of

---

[8]In the graph, the series *rv5* is rescaled as specified in the upcoming section: we took the square root of each value, which is a daily variance measure, and multiplied it by 100.

the SV model when (roughly) calibrating the parameters of the canonical model on a wider set of financial returns.

Note that rescaling the log returns has analogous implications for the section in which we employ measures of realised volatility as a proxy for the latent volatility process. Let us denote by $rv5_t$ the daily realised variance at time t, which we obtain from the *rv5* series provided in the Oxford-Man database. Now, note that in a SV framework one should not directly compare $rv5_t$ to x_t, as $rv5_t$ can be rather seen as an estimate of the variance of the log returns at time t, *i.e.* of e^{x_t} in the SV model. Moreover, by construction of $rv5_t$,[9] the comparison should feature a SV model where the log returns did *not* undergo the percentage transformation. Thus, in order to achieve a common scale with the percentage log returns, we need to multiply $rv5$ by 100^2. To ease the interpretation and representation of the comparison between differently approximated filtering distributions, we finally take the square root of the rescaled $rv5$, so that we compare it to $e^{x_t/2}$ (*In realtà detto così è poco preciso, perchè in realtà andiamo a confrontare con la trasformazione della media dello stato filtrato*).

4 Comparing the filters

In this section the different particle filter techniques we discussed in earlier sections will be employed to approximate the filtering distribution that springs from a simple SV model applied to a real dataset. First, we provide a qualitative comparison between the filters, basing on visual analysis of the (approximated) filtering distributions. Subsequently, we move to a more quantitative approach, in order to assess the quality of the approximations as the number of particles increases. In a way, while in the first part most filters can be rather confidently assumed to be close to convergence, the second subsection will hint at possibly different computational costs across the competing filters. Before moving to such analyses, it is worth mentioning that not all the filters we employ approximate the same filtering distribution $p(x_t|y_{1:t}, \psi)$, where ψ is a vector containing the parameters of the model. Indeed, the Liu and West filter actually approximates the distribution of a slightly different SV model, one that assumes an additional hierarchical level,

$$y_t|x_t \sim N(0, e^{x_t})$$
$$x_t|x_{t-1}, \alpha, \beta, \tau^2 \sim N(\alpha + \beta x_{t-1}, \tau^2)$$
$$x_0 \sim N(0, 100)$$
$$\alpha \sim N(\gamma, \zeta)$$
$$\beta \sim N(\pi, \phi)$$
$$\tau^2 \sim IG(\nu/2, \lambda\nu/2)$$

---

[9]Nota sulla formula, che considera returns, non returns in percentuale. Spiegare qui perchè possa avere senso paragonare questa varianza a quella dei log returns, come fa Guidolin

where $\gamma, \zeta, \pi, \phi, \nu$ and λ are assumed to be known.[10].

4.1 Visual Analysis after Filtering

We now offer a preliminary comparison of the various filtering methods, based on a visual analysis of graphs representing, for each index, the time-varying volatility of the observations, obtained through the application of the different filtering techniques explored in the theoretical analysis. Plots that are compared represent the sequence of percentage log returns for a given index (the observations, in gray) and the sequence of filtered volatilities calculated using filtered states (in red) obtained through a specific filtering technique indicated in the caption of the plot.[11]

First, a general point is that the sequence of percentage log returns are different across the three indices. The reasons why this holds can be associated with the index-construction method and the characteristics of the economy and industrial setting it represents (SEE SECTION X). In particular, as previously noticed, there are extensive similarities between the DOW JONES Index and the S&P500, which can be visualized when looking at the percentage of log returns in the corresponding plots, since both are describing the US Economy. However, the STOXX50E differs quite extensively, especially in the different sensitivity to the Covid-19 Crisis and its consequences, representing the European Economy. Aside from this note, the considerations expressed below on the sequences "filtered volatilities"[12] for the various techniques hold almost uniformly for all the three indices. Therefore, in the next paragraph we will be referring to the set of plots pertaining to one specific index (which could be any of the three).

Generally speaking, note that, the sequence of "filtered volatilities" obtained from any of the filtering techniques follows roughly similar paths across all the plots. Unsurprisingly, the filtered volatility appears wider in periods corresponding to wider fluctuations of the percentage log returns and lower in periods corresponding to smaller such fluctuations. For example, all sequences of filtered volatilities show peaks corresponding to the higher fluctuations due to the 2020 Covid-19 Crisis outbreak and the consequent increases in uncertainty for the later periods.

---

[10]In these empirical applications we keep the values of the parameter we employed in the synthetic data part, see section *fill with the number of the section*

[11]Such filtered volatilities are obtained as follows. For each t, let us denote by μ_t the mean of the generated particles; the filtered volatility is then defined as e^{μ_t}. Note that the number of particles was set at 10000 for each filter: the conclusion of the next section will come back to such a choice.

[12]Per ora nei grafici c'è scritto "Filtered States", verrà corretto.

When focusing the attention on specific filters, first we can discuss the effect of resampling. Indeed, in the plot referring to the Sequential Importance Sampling procedure, the sequence of filtered volatilities is not able to reproduce the peaks as in with other filters, but also is characterized by collapsing confidence intervals. This is because :::::::

Then, if we instead look at the filtered volatilities obtained with resampling, we can appreciate how the filtered sequences for the Basic,[13] Bootstrap, Auxiliary and Optimal Kernel Guided particle filters give rather analogous results, with no differences appearing at a first visual inspection.[14]

However, the sequence of filtered volatilities obtained by applying the Liu and West filter is clearly different from the others. Indeed, it shows a "more detailed" sequence of states (in the sense of the sequence of filtered volatilities being less smooth) and also a higher peak for the wide fluctuation corresponding to the 2020 Covid Crisis outbreak. Therefore, we may say that, compared to other filtering techniques, it offers a better representation of the data. This is because ... *Potremmo parlare di maggiore sensibilità alle oscillazioni dei prezzi, o simili*

4.2 Quality of approximation

Since we do not have access to the exact filtering distribution of the SV model, when assessing how rapidly different particle filters converge to it, we actually need to approximate even such target distribution as to have a benchmark. Fortunately, convergence results about particle filters ensure that when the number of particles diverges, the approximation converges to the target distribution.[15] Thus, we run one of the particle filters with a very large number of particles, and assume that such approximation is close enough to the target to be used itself as the target. With such a benchmark, we can then compare how different filters approach it as the number of particles increases. We employ the Root Mean Squared Error and the Mean Absolute Error to measure discrepancies from the benchmark. Note that the Liu and West filter can not be included in such an analysis: trivially, the comparison assumes that the competing algorithms are approximating the same target distribution.[16]

We start by running the Bootstrap particle filter with N=50000, setting it as the benchmark. In table XX we report the RMSE and MAE measures one obtains

---

[13]By this name, we refer to a particle filter that features no adaptive resampling; rather, resampling is conducted at each step of the algorithm.

[14]Except for the Auxiliary Particle filtered sequence of volatilities, showing a slightly higher peak corresponding to the greatest volatility fluctuation in 2020, if compared to other filters in this subset. This is still valid for all three indices.

[15]Bisognerebbe fare riferimento alla sezione sulla convergenza. Inoltre, va chiarito il punto della "doppia convergenza" per T e per N (ossia, se il nostro T sia sufficientemente alto da parlare già di convergenza quando N è molto grande)

[16]For an analogous reason, we also conduct this analysis only on one of the three indices, the SP500. Indeed, what matters in this section is not how well the model fits the data, but rather how different algorithms approach the same distribution yielded by the same model on a given set of data.

by comparing the mean of the particles generated at each step by the Bootstrap PF with N=50000 and the mean[17] of the particles generated by other particles filters with N equal to 10, 100, 1000, 5000 or 10000.[18]

Table 2: RMSE and MAE, Bootstrap PF 50000 particles

	N	BPF	GPFOPT	APF	BAPF	SIS
RMSE	10	0.42331	0.36853	0.54784	0.55959	0.91686
RMSE	100	0.15689	0.15094	0.24044	0.23488	1.10551
RMSE	1000	0.06901	0.07669	0.08878	0.10295	0.62505
RMSE	5000	0.03817	0.03568	0.05742	0.06110	0.62004
RMSE	10000	0.03045	0.02709	0.04296	0.05151	0.51189
MAE	10	0.28078	0.27961	0.43103	0.43779	0.72753
MAE	100	0.09658	0.09624	0.17570	0.18394	0.91799
MAE	1000	0.03312	0.03697	0.05539	0.06726	0.49718
MAE	5000	0.01889	0.01665	0.03186	0.03448	0.44753
MAE	10000	0.01456	0.01423	0.02449	0.02726	0.40636

As we can see, almost every filter shows dramatic improvement already when passing from 10 particles to 100 particles, while the performance of the SIS algorithm actually gets worse. With N=10 or N=100, the Bootstrap PF and the Guided PF (with optimal proposal kernel) emerge quite clearly as the best performing filters; however, no conclusion should be drawn at this stage, as the large errors seem to suggest quite a wide margin from improvement even for the best performing filters.

As it turns out, such suggestion is confirmed by running the algorithms with N=1000. Passing from 100 to 1000 particles nearly halves the RMSEs across the 5 competing filters, with even more pronounced gains in the MAEs; at N=1000, the Bootstrap and the Guided particle filters still hold their lead over the rest of the field, keeping a sizable hedge over the Auxiliary PF. Such a hedge almost vanishes when increasing the number of particles to 5000: according to the RMSE, the Bootstrap PF performs slightly better than the Guided PF, which in turn yields very similar results to the Auxiliary PF.

We would argue that already for N=5000, but even more clearly for N=10000,

---

[17]As a possible limitation of our exercise, here we do not consider in detail the variance of such generated particles. Nevertheless, one can get at least an idea of the differences in the generated variances by comparing the confidence intervals that will be shown in the graphs below. Unsurprisingly, it will be apparent that including a resampling step is essential to avoid degeneracy of the weights on the particles.

[18]We consider 5 algorithms: a Bootstrap PF, a Guided PF (with the optimal proposal kernel), an Auxiliary PF, a "Basic PF" and a Sequential Importance Sampling alogrithm. By "Basic PF" we refer to a Bootstrap PF that features resampling at each step, whereas resampling never happens in the SIS algorithm: one can then consider these 2 algorithms as extreme cases of an adaptive resampling algorithm, namely one that features resampling when ESS is non negative (*i.e.* always) or when $ESS < 0$ (*i.e.* never). In this sense, we use the shorthand "particle filters" when referring to all the 5 algorithms.

the difference between the Bootstrap, the Guided and the Auxiliary particle filters is so small that none of these filters can be straightforwardly chosen over the others, a figure which visual inspection of the graphs below can confirm. Interestingly enough, no gains seem to arise from resampling at each step with respect to adaptive resampling (as the RMSE and MAE criteria only prefer the Basic PF to the APF on a couple of occasions), whereas the need of some degree of resampling is apparent, as shown by the awful performance of the SIS algorithm with any number of particles. Moreover, the RMSE and the MAE show infrequent disagreement, delivering a rather clear picture of the comparison.

Now, one might argue that the relative performance of the filters is influenced by the choice of the Bootstrap filter as the benchmark algorithm when run with N=50000. Intuitively, if the convergence phase were not reached, the Bootstrap filter would clearly find it easier to replicate the benchmark we set.[19] Such criticism should definitely be taken into account, even more so when noticing that the BT filter did in fact come out on top for several values of N.

As a robustness check, we thus conduct the same analysis employing the Auxiliary Particle Filter as the benchmark, keeping the number of particles set to 50000 as for the benchmark BT filter.

Table 3: RMSE and MAE, Auxiliary PF 50000 particles

	N	BPF	GPFOPT	APF	BAPF	SIS
RMSE	10	0.40549	0.34950	0.53237	0.54597	0.91038
RMSE	100	0.13510	0.14209	0.22243	0.22398	1.10156
RMSE	1000	0.04145	0.06874	0.07796	0.07692	0.61504
RMSE	5000	0.02847	0.04922	0.03931	0.05230	0.61176
RMSE	10000	0.03048	0.03337	0.02648	0.02411	0.50547
MAE	10	0.27266	0.26941	0.42151	0.42909	0.72112
MAE	100	0.08540	0.09034	0.16691	0.17484	0.91468
MAE	1000	0.02329	0.03532	0.05105	0.05753	0.48959
MAE	5000	0.01756	0.02410	0.02662	0.03275	0.44195
MAE	10000	0.01417	0.01718	0.01974	0.01742	0.40221

Using the APF as the benchmark does not deliver a different picture than the one we previously had. Leaving the SIS algorithm aside (whose analogous performance comes as no surprise), it is interesting to note how once again the Auxiliary particle filter actually fares worse than the Bootstrap and the Guided PFs at 10, 100 and 1000 particles.

Its relative performance ameliorates dramatically at N=5000, when the halving of its RMSE and MAE allows the APF to overtake the Guided PF, though

---

[19]Note that all the filters, including the BT filter run with 50000 particles, shared the same seed in the R software, ruling randomness out of these considerations.

falling short of the Bootstrap PF. At N=10000 both the Basic PF and the APF achieve lower RMSEs than the Bootstrap and the Guided PF, even though the MAE criterion still selects the Bootstrap as the closest to the target distribution. Interestingly, this time the performance of the Bootstrap filter worsens when N is increased from 5000 to 10000 (both according to RMSEs and MAEs, whose disagreements, once again, are overall infrequent). Such evidence suggests that, even at very large (yet finite) values of N, some difference between the approximated filtering distributions still persists across our particle filters.

Indeed, by comparing the two benchmark approximations with N=50000, we can see that, for example, the Bootstrap PF run with N=5000 is closer to the benchmark APF than the benchmark BT is.[20]

Table 4: Bootstrap vs Auxiliary PF, 50000 particles

	RMSE	MAE
Discrepancy	0.04408	0.01668

Nevertheless, we should not overplay tiny differences. Rather, we prefer to stress the points that emerged from both benchmark analyses and we can therefore support quite confidently.

First, if on the one hand it is clear that some degree of resampling is essential, on the other we found no gains from resampling at each period.

Second, when the number of particles is set at 5000 or more, neither benchmark could single out an algorithm that clearly outperformed the others, when restricting the analysis to procedures that feature adaptive resampling rules.

Third, when the number of particles was set to 10, 100 or 1000, the Bootstrap particle filter and the Guided particle filter with optimal proposal kernel significantly emerged as the best performing algorithms for our SV model, suggesting that their computational cost is lower than that of the Auxiliary particle filter, which needed many more particles to "get rolling".

Finally, some discrepancy between different approximations persisted even when the number of particles was set to 50000. However, the narrow extent of the RMSEs and MAEs with respect to both benchmarks supports the idea that at N=10000 the best-performing algorithms have already entered the phase of convergence. Even in light of these markedly more computationally expensive benchmarks, we then choose N=10000 as the number of particles which we routinely employ in this empirical application.

5 Realised volatility as benchmark

---

[20]According to the RMSE, also the Bootstrap with N=10000 is closer than the Bootstrap benchmark; however, in this occasion the MAE disagrees. Similarly, while according to the RMSE the APF with N=10000 is closer to the benchmark BT than the benchmark APF is, the MAE is lowest for the benchmark APF.

Table 5: RMSE and MAE, realised volatility as benchmark

	N	BPF	GPFOPT	APF	BAPF	LW	SIS
RMSE SP500	10000	0.50864	0.50926	0.50994	0.51033	0.42872	0.60288
MAE SP500	10000	0.35377	0.35315	0.35174	0.35306	0.29156	0.43197
RMSE STOXX50E	10000	0.46358	0.46345	0.47571	0.46783	0.34407	0.65253
MAE STOXX50E	10000	0.30190	0.30418	0.30422	0.30421	0.23716	0.44077
RMSE DJI	10000	0.50762	0.50622	0.50934	0.50536	0.44083	0.60467
MAE DJI	10000	0.33754	0.33827	0.33745	0.33615	0.28095	0.41563

Il fatto che con 50000 particles si performi peggio conferma che il problema sia il SV sotto, che è chiaramente battuto dal modello che sta sotto Liu West

Table 6: RMSE and MAE, realised volatility as benchmark (SP500)

	N	BPF	APF
RMSE	50000	0.51170	0.51489
MAE	50000	0.35419	0.35534

6 Bibliography

Allen, D. E., & McAleer, M. (2020). Do we need stochastic volatility and generalised autoregressive conditional heteroscedasticity? Comparing squared end-of-day returns on FTSE. Risks, 8(1), 12.

Allen, D. E. (2020). Stochastic Volatility and GARCH: Do Squared End-of-Day Returns Provide Similar Information?. Journal of Risk and Financial Management, 13(9), 202.

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.

Barndorff-Nielsen, O. E., & Shephard, N. (2001). Non-Gaussian Ornstein–Uhlenbeck-based models and some of their uses in financial economics. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 63(2), 167-241.

Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. Journal of econometrics, 31(3), 307-327.

Breidt, F. J., Crato, N., & De Lima, P. (1998). The detection and estimation of long memory in stochastic volatility. Journal of econometrics, 83(1-2), 325-348.

Chan, J. C. (2013). Moving average stochastic volatility models with application to inflation forecast. Journal of Econometrics, 176(2), 162-172.

Chib, S., Nardari, F., & Shephard, N. (2002). Markov chain Monte Carlo methods for stochastic volatility models. Journal of Econometrics, 108(2), 281-316.

Chib, S., Nardari, F., & Shephard, N. (2006). Analysis of high dimensional multivariate stochastic volatility models. Journal of Econometrics, 134(2), 341-371.

Chopin, N., & Papaspiliopoulos, O. (2020). An introduction to sequential Monte Carlo (Vol. 4). Springer.

Comte, F., & Renault, E. (1998). Long memory in continuous-time stochastic volatility models. Mathematical finance, 8(4), 291-323.

Christoffersen, P., Jacobs, K., & Mimouni, K. (2010). Volatility dynamics for the S&P500: Evidence from realized volatility, daily returns, and option prices. The Review of Financial Studies, 23(8), 3141-3189.

Dimitrakopoulos, S., & Kolossiatis, M. (2020). Bayesian analysis of moving average stochastic volatility models: modeling in-mean effects and leverage for financial time series. Econometric Reviews, 39(4), 319-343.

Douc, R., Moulines, E., & Stoffer, D. (2014). Nonlinear time series: Theory, methods and applications with R examples. CRC press.

Engle, R. F. (1982). Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation. Econometrica: Journal of the econometric society, 987-1007.

Eraker, B. (2004). Do stock prices and volatility jump? Reconciling evidence from spot and option prices. The Journal of Finance, 59(3), 1367-1403.

Friz, P. K., Gassiat, P., & Pigato, P. (2021). Precise asymptotics: robust stochastic volatility models. The Annals of Applied Probability, 31(2), 896-940.

Gatheral, J., Jaisson, T., & Rosenbaum, M. (2018). Volatility is rough. Quantitative finance, 18(6), 933-949.

Harvey, A. C., & Shephard, N. (1996). Estimation of an asymmetric stochastic volatility model for asset returns. Journal of Business & Economic Statistics, 14(4), 429-434.

Harvey, A., Ruiz, E., & Shephard, N. (1994). Multivariate stochastic variance models. The Review of Economic Studies, 61(2), 247-264.

Heber, Gerd, Asger Lunde, Neil Shephard and Kevin Sheppard (2009). Oxford-Man Institute's realized library. Oxford-Man Institute, University of Oxford, version 0.3. Available at: https://realized.oxford-man.ox.ac.uk/.

Heston, S. L. (1993). A closed-form solution for options with stochastic volatility with applications to bond and currency options. The review of financial studies, 6(2), 327-343.

Hull, J., & White, A. (1987). The pricing of options on assets with stochastic volatilities. The journal of finance, 42(2), 281-300.

Jacquier, E., Polson, N. G., & Rossi, P. (1994). Bayesian analysis of stochastic volatility models. Journal of Business and Economic Statistics, 12(4), 371-389.

Jacquier, E., Polson, N. G., & Rossi, P. E. (2004). Bayesian analysis of stochastic volatility models with fat-tails and correlated errors. Journal of Econometrics, 122(1), 185-212.

Johannes, M., Korteweg, A., & Polson, N. (2014). Sequential learning, predictability, and optimal portfolio returns. The Journal of Finance, 69(2), 611-644.

Kastner, G. (2019). Dealing with stochastic volatility in time series using the R package stochvol. arXiv preprint arXiv:1906.12134.

Kim, S., Shephard, N., & Chib, S. (1998). Stochastic volatility: likelihood inference and comparison with ARCH models. The review of economic studies,

65(3), 361-393.

Kim, J., & Stoffer, D. S. (2008). Fitting stochastic volatility models in the presence of irregular sampling via particle methods and the EM algorithm. Journal of time series analysis, 29(5), 811-833.

Koopman, S. J., & Hol Uspensky, E. (2002). The stochastic volatility in mean model: empirical evidence from international stock markets. Journal of applied Econometrics, 17(6), 667-689.

Liesenfeld, R., & Jung, R. C. (2000). Stochastic volatility models: conditional normality versus heavy-tailed distributions. Journal of applied Econometrics, 15(2), 137-160.

Luo, R., Zhang, W., Xu, X., & Wang, J. (2018, April). A neural stochastic volatility model. In Thirty-second AAAI conference on artificial intelligence.

Mao, X., Czellar, V., Ruiz, E., & Veiga, H. (2020). Asymmetric stochastic volatility models: Properties and particle filter-based simulated maximum likelihood estimation. Econometrics and Statistics, 13, 84-105.

Nakajima, J., & Omori, Y. (2012). Stochastic volatility model with leverage and asymmetrically heavy-tailed error using GH skew Student's t-distribution. Computational Statistics & Data Analysis, 56(11), 3690-3704.

Omori, Y., Chib, S., Shephard, N., & Nakajima, J. (2007). Stochastic volatility with leverage: Fast and efficient likelihood inference. Journal of Econometrics, 140(2), 425-449.

Pitt, M. K., & Shephard, N. (1999). Filtering via simulation: Auxiliary particle filters. Journal of the American statistical association, 94(446), 590-599.

Poon, S. H., & Granger, C. (2005). Practical issues in forecasting volatility. Financial analysts journal, 61(1), 45-56.

So, M. E. P., Lam, K., & Li, W. K. (1998). A stochastic volatility model with Markov switching. Journal of Business & Economic Statistics, 16(2), 244-253.

Xu, X., & Chen, Y. (2021). Deep Stochastic Volatility Model. arXiv preprint arXiv:2102.12658.

Zhang, B., Chan, J. C., & Cross, J. L. (2020). Stochastic volatility models with ARMA innovations: An application to G7 inflation forecasts. International Journal of Forecasting, 36(4), 1318-1328.