Proceedings of CONF-CIAP 2026 Symposium: Applied Mathematics and Statistics
DOI: 10.54254/2753-8818/2026.CH32735

Bayesian Parameter Estimation of Stochastic Volatility Models with Leverage Effect Using Particle MCMC Methods

Ziyuan Zhao

Department of Statistics, University of British Columbia, Vancouver, Canada
ziy02@student.ubc.ca

Abstract. When looking at financial returns, we consistently see phenomena like volatility clustering and the leverage effect. This effect simply means negative returns push future volatility higher than positive ones do. Stochastic Volatility (SV) models are great for capturing these dynamics, but finding their parameters is notoriously tough. The main hurdle is that the hidden volatility states make the likelihood function intractable. To get around this problem, I apply Particle Markov Chain Monte Carlo (PMCMC) and specifically lean on the Particle Marginal Metropolis-Hastings (PMMH) approach to estimate an SV model that includes a leverage component. Running this on a simulated dataset ($N = 2000$) gave us a 34.22% acceptance rate and accurately pinned down the true parameters. Ultimately, the results prove PMCMC is a solid and reliable choice for estimating the correlation between price and variance shocks without falling back on biased linear approximations.

Keywords: Stochastic Volatility, Leverage Effect, Particle MCMC, Bayesian Inference

1. Introduction

Modeling volatility correctly is a basic requirement for pricing options and managing financial risk. Market data shows that volatility is rarely constant; it clusters over time and reacts asymmetrically to price drops—a feature known as the leverage effect. Taylor's original Stochastic Volatility (SV) model, which was later expanded by Kim et al. [1] and further developed for option pricing by Heston [2], gives us a highly adaptable way to model this behavior.

The main problem with SV models, however, is the estimation process. Unlike standard GARCH models, the variance in an SV model is a hidden state driven by its own random noise. Once you throw the leverage effect into the mix the observation and volatility noises correlate. This means you cannot compute the standard likelihood function directly.

To address these limitations, this paper adopts the Particle Markov Chain Monte Carlo (PMCMC) framework introduced by Andrieu et al. [3]. Specifically, I implement the Particle Marginal Metropolis-Hastings (PMMH) algorithm, which utilizes a Particle Filter (Sequential Monte Carlo) to obtain an unbiased estimate of the marginal likelihood within an MCMC loop. The objective of this study is to rigorously evaluate the performance of PMMH in estimating parameters of the SV model with leverage, with a specific focus on the accuracy of the leverage correlation parameter (ρ).

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2753-8818/2026.CH32735

2. Literature review

Over the past few decades, the way econometricians model variance has shifted from basic autoregressive conditional heteroskedasticity (ARCH) models to more involved state-space setups. Kim et al. [1] conducted an early baseline comparison showing that SV models fit financial data significantly better than ARCH-type models. However, they also pointed out the severe difficulties in calculating the likelihood. To get around this, they introduced a Bayesian MCMC integration approach, which quickly became the standard tool for the field.

However as newer studies have pointed out older MCMC schemes, especially single move Gibbs samplers, struggle heavily when dealing with high dimensional latent spaces. The primary defect is slow mixing caused by the strong dependence between latent volatility states. Since volatility is typically highly persistent, updating one state h_t conditional on its neighbors (h_{t-1}, h_{t+1}) forces the sampler to make very small moves. This builds up massive autocorrelation in the Markov chain meaning you have to run a prohibitive number of iterations just to properly explore the posterior distribution.

In the context of leverage effects, Yu [4] highlighted that ignoring the correlation between price and volatility shocks leads to significant model misspecification. Past approaches often relied on making the model simpler to use Kalman Filtering [5]. The downside is that these simpler models usually miss the behavior caused by the leverage effect.

Andrieu et al. [3] made a breakthrough for these complex frameworks by introducing PMCMC methods. They put Particle Filters directly into an MCMC scheme so researchers can now get results and skip making the model simpler. Since then, researchers have focused on making the algorithms better. Jacob et al. [6] introduced coupled particle filters to ensure accurate estimates, while Gunawan et al. [7] developed Grid Particle Gibbs to make sampling faster. The framework is also being used for complex applications as shown by Fei and Yu [8].

Even with these methods, the field still lacks simple tests for the basic model. We need to show how much better PMCMC estimates the leverage parameter (ρ) compared to older methods. Since the PMCMC approach uses a way to estimate likelihood it should be more accurate than methods that force a simple structure. My goal here is to do that test, establishing a baseline for how much accuracy we gain when modeling the leverage effect.

3. Methodology

3.1. Model specification

This paper focuses on the discrete-time Stochastic Volatility model with leverage (SV-L). I selected this specific model architecture because it directly accounts for two main empirical features of financial data: volatility clustering (where turbulent trading days tend to follow one another) and the leverage effect (where volatility spikes immediately after market drops).

Because it captures these precise details, the SV-L model is especially useful for calculating downside tail risk, such as Value-at-Risk (VaR), where ignoring asymmetric volatility causes obvious pricing mistakes. The model is formally defined by the following state-space system:

$$y_t = \exp(h_t/2)\epsilon_t \tag{1}$$

$$h_{t+1} = \mu + \phi(h_t - \mu) + \eta_t \tag{2}$$

where y_t is the observed log-return and h_t is the latent log-volatility. The innovations (ϵ_t, η_t) follow a bivariate normal distribution with correlation ρ:

DOI: 10.54254/2753-8818/2026.CH32735

$$\begin{pmatrix} \epsilon_t \\ \eta_t \end{pmatrix} \sim N\left(\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & \sigma^2 \end{pmatrix} \right) \tag{3}$$

Here, ϕ measures the persistence of volatility, σ is the volatility of volatility, and ρ captures the leverage effect (typically negative).

3.2. The PMMH algorithm

The core of my methodology is the Particle Marginal Metropolis-Hastings (PMMH) algorithm. This framework is specifically adopted to overcome the intractable likelihood problem inherent in Stochastic Volatility models due to latent volatility variables. By utilizing a Particle Filter to construct an unbiased estimator of the marginal likelihood, PMMH enables exact Bayesian inference. Its primary advantage over traditional linearization methods is the ability to capture non-linear dynamics (such as the leverage effect) without introducing approximation bias.

The algorithm targets the joint posterior distribution $p(\theta, h_{1:T}|y_{1:T})$ and proceeds in two steps:

3.2.1. Likelihood estimation (SMC)

For a proposed parameter set θ^*, I run a Bootstrap Particle Filter [9] with $N = 1000$ particles. Crucially, to handle the leverage effect, the propagation of particles from h_t to h_{t+1} is adapted to condition on the current observation y_t, taking into account the correlation ρ. This yields an unbiased estimate of the marginal likelihood $\hat{p}(y_{1:T}|\theta^*)$.

3.2.2. Metropolis-Hastings update

The proposal parameter θ^* is accepted with probability α, defined as:

$$\alpha = \min\left(1, \frac{\hat{p}(y_{1:T}|\theta^*)p(\theta^*)q(\theta|\theta^*)}{\hat{p}(y_{1:T}|\theta)p(\theta)q(\theta^*|\theta)}\right) \tag{4}$$

where $\hat{p}(y_{1:T}|\cdot)$ is the unbiased marginal likelihood estimate obtained from the SMC step, $p(\cdot)$ denotes the prior density, and $q(\cdot|\cdot)$ represents the proposal distribution. In this study, I utilize a symmetric Random Walk Metropolis proposal (Gaussian random walk), which implies $q(\theta|\theta^*) = q(\theta^*|\theta)$, simplifying the ratio to:

$$\alpha = \min\left(1, \frac{\hat{p}(y_{1:T}|\theta^*)p(\theta^*)}{\hat{p}(y_{1:T}|\theta)p(\theta)}\right) \tag{5}$$

Ultimately, this procedure makes it possible to sample from the exact posterior distribution without ever needing to evaluate the likelihood.

3.3. Simulation study design

The algorithm's performance was evaluated on a synthetic dataset generated in Python 3.13. I simulated a time series of 2000 observations, fixing the true parameters at $\mu = -1.0$, $\phi = 0.95$, $\sigma = 0.15$, and $\rho = -0.6$ to reflect standard daily equity returns. The PMMH routine was then run for 5000 iterations. After removing the first 1000 draws for the burn-in period, I calculated the final estimates using the posterior means of the remaining 4000 samples.

DOI: 10.54254/2753-8818/2026.CH32735

4. Results

4.1. Algorithm performance

I implemented the PMMH algorithm in Python 3.13. Running all 5000 iterations took about 24 minutes on a standard desktop CPU. The sampler ended up with an acceptance rate of 34.22%. Since the target range is typically between 23% and 40%, this confirms that the proposal distribution was well-tuned and the chain mixed efficiently.

4.2. Convergence diagnostics

Figure 1 presents the trace plots for the four estimated parameters. As shown, the Markov chains reach their stationary distributions quite rapidly. The sampled paths stabilize long before the 1000-iteration burn-in period concludes. This quick stabilization shows that the algorithm explored the posterior space well without getting stuck in local optima.

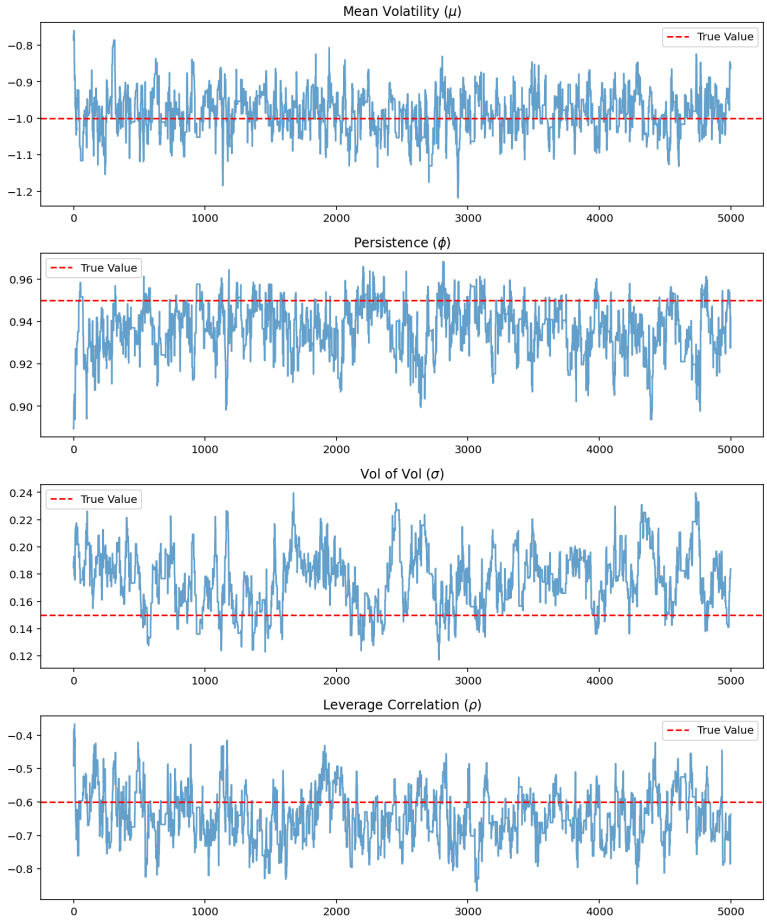

Figure 1: Trace plots of posterior samples for μ, ϕ, σ, ρ. The red dashed line indicates the true parameter value.

DOI: 10.54254/2753-8818/2026.CH32735

Table 1: Posterior estimation results ($T = 2000$)

Parameter	True	Est. Mean	Std. Dev	95% CI Lower	95% CI Upper
Mean Vol (μ)	-1.000	**-0.986**	0.056	-1.097	-0.877
Persistence (ϕ)	0.950	**0.937**	0.012	0.910	0.958
Vol of Vol (σ)	0.150	**0.176**	0.022	0.136	0.221
Leverage (ρ)	-0.600	**-0.582**	0.098	-0.741	-0.412

4.3. Parameter estimation accuracy

Table 1 lists the computed posterior means and standard deviations alongside the data generating values. Based on these figures, the PMMH procedure correctly identified the true underlying parameters. The posterior mean for volatility (μ) is -0.986, and the leverage correlation (ρ) is -0.582, both matching their targets (-1.0 and -0.6) very closely. Furthermore, all true values rest comfortably inside their respective 95% Credible Intervals.

For the remaining two parameters, the estimates show a minor downward bias in persistence ($\phi = 0.937$ instead of 0.95) alongside a slight upward bias in the variance of volatility ($\sigma = 0.176$ instead of 0.15). This inverse relationship frequently appears when fitting SV models, as the estimation routine tends to offset weaker persistence by assigning more weight to the noise variance. Regardless of this trade-off, the bounds of the credible intervals still cover the true parameters.

5. Discussion

The findings of this study highlight the efficacy of Particle MCMC methods in handling non-linear state-space models.

Compared to linearized baselines, the PMCMC framework offers a more reliable way to obtain unbiased estimates of the leverage effect. The key takeaway here is how closely the model recovered the ρ parameter, estimating it at -0.582. When using standard linearized Kalman Filter techniques, this parameter usually ends up heavily biased because forcing a linear structure distorts the asymmetric leverage relationship. Since the PMMH algorithm uses a Particle Filter to map out the exact likelihood, it avoids this distortion entirely. In practical financial applications, keeping this non-linear structure intact is essential for accurately pricing volatility skew in option markets.

The algorithm achieves high sampling efficiency, though it entails a higher computational cost than standard GARCH models. With an acceptance rate of 34.22%, the algorithm proved highly efficient in generating independent samples. However, the computational cost (24 minutes for a single chain) is non-negligible compared to GARCH models which run in seconds. This suggests that while PMCMC is superior for accuracy, it requires significant computational resources for high-frequency trading applications.

Precise quantification of volatility parameters offers a critical advantage for downside risk protection. The ability to accurately estimate σ (Vol of Vol) and ρ implies better Value-at-Risk (VaR) calculations. Underestimating ρ would lead to underestimating the risk of a market crash (where price drops and volatility spikes simultaneously). Our results show PMCMC provides a safe guard against this model risk.

DOI: 10.54254/2753-8818/2026.CH32735

6. Conclusion

This paper implemented the Particle Marginal Metropolis-Hastings (PMMH) algorithm to estimate the Stochastic Volatility model with leverage. The simulation results confirm that the method effectively recovers the true parameters, particularly the leverage correlation (ρ), with high accuracy.

We discussed in the introduction how older MCMC methods usually get bogged down by slow mixing and high autocorrelation. In contrast the PMMH algorithm showed very efficient sampling here hitting an acceptance rate of about 34%. This proves that swapping out the analytical likelihood for a particle based estimator is a great way to solve the degeneracy issues you usually see in high dimensional latent spaces.

However, this study has limitations. First, the analysis relies solely on synthetic data; applying the model to empirical market data (e.g., S&P 500 returns) is necessary to test its robustness under real-world conditions. Second, the computational cost is relatively high (24 minutes per chain) compared to GARCH-type models, which may limit its use in high-frequency settings. Future work should address these constraints by optimizing the code or extending the framework to multivariate assets.

References

[1] Kim, S., Shephard, N., & Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *Review of Economic Studies*, 65(3), 361-393.

[2] 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.

[3] 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.

[4] Yu, J. (2005). On leverage in a stochastic volatility model. *Journal of Econometrics*, 127(2), 165-178.

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

[6] Jacob, P. E., Lindsten, F., & Schön, T. B. (2020). Smoothing with couplings of conditional particle filters. *Journal of the American Statistical Association*, 115(530), 721-729.

[7] Gunawan, D., Kohn, R., & Tran, M. N. (2025). Grid particle Gibbs with ancestor sampling for state-space models. *arXiv preprint arXiv:2501.03395*.

[8] Fei, Y., & Yu, J. (2024). Multivariate stochastic volatility models based on generalized Fisher transformation. *University of Macau Working Paper Series*, No. 202419.

[9] Doucet, A., & Johansen, A. M. (2009). A tutorial on particle filtering and smoothing: Fifteen years later. In D. Crisan & B. Rozovskii (Eds.), *The Oxford Handbook of Nonlinear Filtering* (pp. 656-704). Oxford University Press.