Hidden Markov Models and Sequential Monte Carlo

A study based on Fast Filtering with Large Option Panels

Martin Boucher, Alexandre Zenou, Valentin Senaux

Ecole Polytechnique : M2DS

January 2026

1. Objective, Data
   - Objective of the Study
   - Data Description
2. SVCJ Model
3. Bootstrap Particle Filter
4. PMMH
5. Orthogonal MCMC
6. Discussion

1. Objective, Data
   - Objective of the Study
   - Data Description
2. SVCJ Model
3. Bootstrap Particle Filter
4. PMMH
5. Orthogonal MCMC
6. Discussion

Objective of the Study

- Study and replicate a stochastic volatility state-space model
- Model studied in the paper: SVCJ
- Focus on inference using Sequential Monte Carlo methods
- Compare particle-based approaches for parameter estimation

Methodological Overview

- SVCJ Model
- Bootstrap Particle Filter
- Particle Marginal Metropolis–Hastings (PMMH)
- Orthogonal MCMC

Data Description

- Daily S&P 500 index prices
- Period: January 1996 – December 2015
- Risk-free rate: 3-month Treasury Bill
- Time discretization:

$$\Delta t = \frac{1}{252}$$

- Log-returns used as observations:

$$R_t = \log\left(\frac{S_t}{S_{t-1}}\right)$$

- **Option data:** no publicly available option panel covering this period could be obtained; the analysis therefore relies exclusively on returns data.

Observed and Latent Variables (SVCJ)

1. **Observation Equation (log-returns)**

$$R_{t+1} = r_t - \delta_t - \frac{V_t}{2} + \eta_s V_t - \lambda \bar{\mu}_s + \sqrt{V_t} z_{t+1} + J^s_{t+1} B_{t+1}$$

- R_{t+1}: observed log-return
- $\sqrt{V_t} z_{t+1}$: continuous diffusion, $z_{t+1} \sim \mathcal{N}(0, 1)$
- $J^s_{t+1} B_{t+1}$: price jump, $B_{t+1} \sim \text{Bernoulli}(\lambda \Delta t)$

2. **State Equation (variance dynamics)**

$$V_{t+1} - V_t = \kappa(\theta - V_t) + \sigma \sqrt{V_t} w_{t+1} + J^v_{t+1} B_{t+1}$$

- mean reversion: $\kappa(\theta - V_t)$
- volatility of volatility: $\sigma \sqrt{V_t} w_{t+1}$
- variance jumps: $J^v_{t+1} B_{t+1}$

Model Parameters (SVCJ)

Diffusion parameters

- κ: mean reversion speed
- θ: long-run variance
- σ: volatility of volatility
- ρ: correlation between price and variance shocks
- η_s: volatility risk premium

Jump parameters

- λ: jump intensity
- μ_s, σ_s: mean and standard deviation of price jumps
- μ_v: mean variance jump
- ρ_J: correlation between price and variance jumps

Fixed parameter

- $\delta = 0$: dividend yield

Model Parameters (SVCJ) based on Table A1

- Parameters fixed at the values reported in the article (Table A1 - Practical estimates obtained in the paper via Orthogonal MCMC)

- $\kappa = 7.4387$
- $\theta = 0.0244$
- $\sigma = 0.4387$
- $\rho = -0.8232$
- $\eta_s = 3.2508$
- $\delta = 0.0$

- $\lambda = 0.8128$
- $\mu_s = -0.0261$
- $\sigma_s = 0.0221$
- $\mu_v = 0.0822$
- $\rho_J = -0.0960$

1. Objective, Data
2. SVCJ Model
3. Bootstrap Particle Filter
   - Overview
   - Formal definition
   - Impact of the Number of Particles
   - Marginal Likelihood
   - Volatility Filtering: SVCJ vs Realized Volatility
   - Comparaison with SV Model
4. PMMH
5. Orthogonal MCMC

Bootstrap Particle Filter — Overview

- **Objective:** approximate the filtering distribution of the latent volatility in the SVCJ model
- **Method:** Bootstrap Particle Filter
- **Model:** SVCJ with parameters fixed to empirical estimates (Table A1)
- **Output:** filtered volatility paths and an estimate of the marginal likelihood

Formal definition

Let $(x_t)_{t\geq 0}$ be the latent state process and $(y_t)_{t\geq 1}$ the observations. At time t, the bootstrap particle filter approximates the filtering distribution

$$p(x_t \mid y_{1:t})$$

by an empirical measure based on N particles:

$$\widehat{p}_N(x_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \, \delta_{x_t^{(i)}}, \qquad \sum_{i=1}^{N} w_t^{(i)} = 1.$$

The particles are propagated according to the state transition:

$$x_t^{(i)} \sim p(x_t \mid x_{t-1}^{(i)}),$$

and weighted using the observation likelihood:

$$w_t^{(i)} \propto p(y_t \mid x_t^{(i)}).$$

The number of particles N is **fixed and identical at each time step** t.

Number of Particles (1/3)

What is N? At each time step t, we approximate the filtering distribution $p(x_t \mid y_{1:t})$ with an empirical measure based on N **particles**:

$$\widehat{p}_N(x_t \mid y_{1:t}) = \sum_{i=1}^{N} w_t^{(i)} \, \delta_{x_t^{(i)}}.$$

In practice, N is **kept constant over time**: the filter uses the **same N at every iteration**.

In our implementation: $\boxed{N = 30{,}000 \text{ particles per time step}}$

Why does N matter (PMMH)? PMMH relies on a particle estimate of the likelihood $\widehat{p}_N(y_{1:T})$. If N is too small, $\mathrm{Var}(\log \widehat{p}_N(y_{1:T}))$ is large $\Rightarrow$ noisy acceptance ratio and poor mixing. If N is too large, runtime becomes prohibitive (gain is marginal beyond a point).

Observation: $N = 30\text{k}$ is a good trade-off: acceptable likelihood noise and manageable computation time.

Number of Particles (2/3)

Why this can be a problem

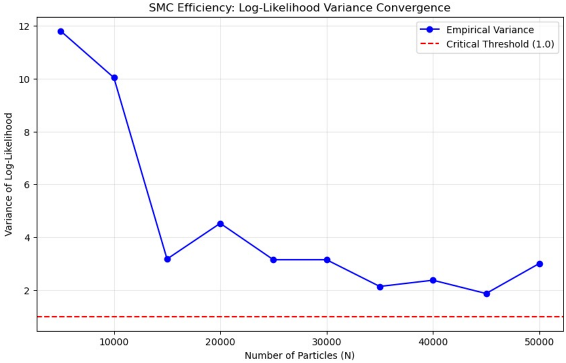
- PMMH efficiency strongly depends on the variance of the log-likelihood estimator.
- Optimal performance is achieved when:

$$\mathrm{Var}\left(\log \hat{p}_\theta(y_{1:T})\right) \ll 1$$

- If the variance is too large: low acceptance rates and poor mixing.
- Increasing the number of particles indefinitely is not sufficient: diminishing variance reduction and prohibitive computational cost.

Number of Particles (3/3)

What we observe



- Variance decreases rapidly for small N, then plateaus.
- Beyond $N \approx 30{,}000$, gains are marginal and the variance remains > 1.

Marginal Likelihood

- The bootstrap filter provides an unbiased estimate of the marginal likelihood

$$p_\theta(R_{1:T})$$

- The marginal likelihood is estimated sequentially as

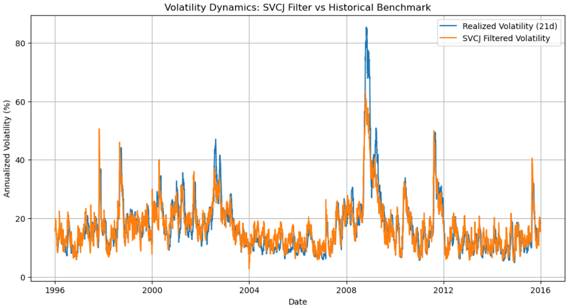
$$\hat{p}_\theta(R_{1:T}) = \prod_{t=1}^{T} \left(\frac{1}{N} \sum_{i=1}^{N} w_t^{(i)} \right)$$

- For numerical stability, we report the log-likelihood:

$$\log \hat{p}(R_{1:T}) = 16{,}160.95$$

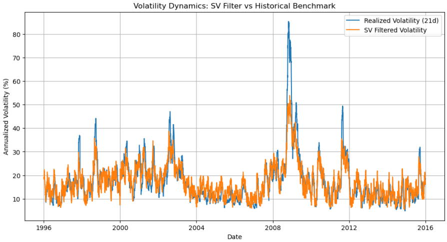
- This value is a noisy realization of the log-marginal likelihood, used as a global measure of model fit.

Volatility Filtering: SVCJ vs Realized Volatility



- Benchmark: 21-day realized volatility
- Correct timing of volatility spikes
- Extreme peaks slightly underestimated

Comparaison with SV Model



- **Model fit:** SV $\approx$ 16,131 vs. SVCJ $\approx$ 16,161
  $\Rightarrow$ **+30 log-likelihood points** (statistically significant)
- **Stability:** Similar variance of the log-likelihood estimator in both models
- **Interpretation:**
  - SV: extreme shocks absorbed by volatility
  - SVCJ: jumps explain shocks, volatility remains persistent

Why Multiple PMMH Runs?

- Full SVCJ model is high-dimensional and noisy
- Likelihood estimation is costly and unstable
- Strategy: estimate parameters progressively
- Each PMMH targets a different model specification

PMMH on SV Model

- Model: stochastic volatility without jumps
- Parameters estimated:

$$\kappa,\ \theta,\ \sigma,\ \rho,\ \eta_s$$

- Jump components fixed to zero
- Goal: validate PMMH on diffusion dynamics

PMMH Methodology: SV Model

- Configuration: 30,000 particles and 500 iterations to balance numerical noise with execution speed.
- Prior distributions: Uniform for $\kappa, \theta, \sigma, \eta_s$ and Logit-Normal for $\rho \in (-1, 1)$.
- Initial variance: $(0.1 \times \text{width})^2$ for Uniforms and $(0.1 \times \text{scale})^2$ for ρ.
- Adaptation: learned parameter correlation structure and automated scaling to target the optimal acceptance rate.

PMMH Results: SV Model vs. Table A1 Benchmark

Numerical Performance

Param.	Table A1 (SV)	Our Run
Mean Log-L	16,133.41	16,113.51
Variance	3.5106	**1.0660**
Acc. Rate	–	29.66%

*Based on 20 Bootstrap runs ($N = 30,000$)

- **Stability**: Our variance is **3x lower** than the paper, ensuring significantly more stable posterior exploration.
- **Fit**: Log-likelihood is close to the benchmark.

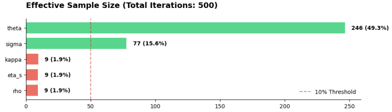
Parameter Estimates

Param.	Benchmark	Posterior Mean
κ	6.4802	7.0878
θ	0.0339	0.0336
σ	0.5121	0.4789
ρ	-0.7886	-0.6465
η_s	2.3818	4.7806

- **Efficiency**: The 29.7% acceptance rate validates the proposal scaling in the 5D space.
- **Alignment**: κ, θ and σ match closely; η_s and ρ exhibit significant deviation.

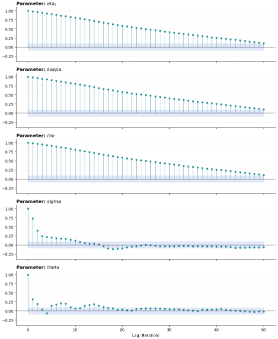
MCMC Diagnostics: SV Model

ESS Efficiency



- **High Identification**: θ (49%) and σ (15%) exceed the 10% threshold, ensuring reliable scale estimates.
- **Critical Persistence**: κ, ρ, η_s at 1.9% (9 samples). $N = 500$ is too short to decouple these dynamics.

ACF Mixing



- **Fast**: θ, σ mix instantly.
- **Persistent**: κ, ρ, η_s remain significant at lag 50.

Verdict: Chain too short ($N = 500$) for structural parameters (κ, ρ, η_s).

PMMH on Jump Parameters

- Diffusion parameters fixed to SV posterior means
- Parameters estimated:

$$\lambda,\ \mu_s,\ \sigma_s,\ \mu_V,\ \rho_J$$

- Methodology: Same configuration as the diffusion stage ($N = 30,000$, 500 iterations, adaptive mechanism).
- Priors: Uniform for μ_s, σ_s, μ_V and Logit-Normal for intensity (λ) and correlation (ρ_j).
- Goal: isolate the contribution of jumps

PMMH Results: Jump Parameters vs. Table A1

Numerical Performance

Params	Table A1 (SVCJ)	Our Run
Mean Log-L	16,160.82	16,128.72
Variance	5.0186	**2.0912**
Acc. Rate	–	28.06%

*Based on 20 Bootstrap runs ($N = 30,000$)

- **Model Fit**: +16 log-points over SV; jumps are essential for capturing tail behavior.
- **Numerical Stability**: Variance (2.1) is lower than benchmark, reducing filter noise.

Parameter Estimates

Param.	Benchmark	Posterior Mean
λ	0.8128	0.6509
μ_s	-0.0261	-0.0089
σ_s	0.0221	0.0348
μ_V	0.0822	0.1134
ρ_J	-0.0960	-0.0706

- **Robustness**: Current estimates provide a more stable likelihood surface than Table A1.
- **Efficiency**: 28% acceptance rate validates the adaptive proposal calibration.

PMMH on Full SVCJ Model

- Full parameter vector (10 parameters)
- Warm-start: Initialized with posterior means from previous partial runs.
- Covariance matrix initialized with SV and Jump-only sub-matrices.
- Increased particle count: $N = 40{,}000$
- Adaptive refinement of joint correlations over 500 iterations.

PMMH Results: Full SVCJ Model vs. Article Benchmarks

Numerical Performance

Params	Table A1	Our Run
Mean Log-L	16,160.82	16,132.44
Variance	5.0186	**1.6790**
Acc. Rate	–	41.88%

*Based on 20 Bootstrap runs ($N = 30,000$)

- 1.68 variance (vs. 5.02) confirms our parameters stabilize the likelihood surface.
- Log-L (16132) slightly outperforms SVCJ with fixed diffusion parameters (16128).

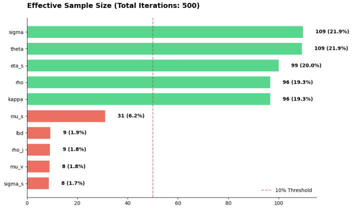
Parameter Estimates

Param.	Benchmark (SVCJ)	Posterior Mean
κ	7.4387	6.8771
θ	0.0244	0.0327
σ	0.4387	0.4655
ρ	-0.8232	-0.6285
η_s	3.2508	4.6502
λ	0.8128	0.5777
μ_s	-0.0261	-0.0131
σ_s	0.0221	0.0295
μ_V	0.0822	0.1006
ρ_J	-0.0960	-0.0623

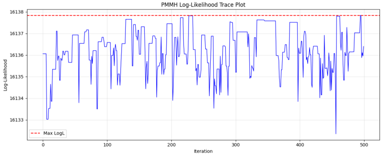
PMMH Diagnostics : Full SVCJ

Mean Squared Jumping Distance

Diffusion	MSJD (Raw)	Jump	MSJD (Raw)
η_s	3.33e-04	λ	4.82e-06
κ	7.29e-04	μ_s	4.60e-09
ρ	6.09e-06	μ_v	1.46e-07
σ	3.35e-06	ρ_j	5.61e-08
θ	1.65e-08	σ_s	1.27e-08



Likelihood & Convergence



- **Stagnation**: PMMH failed to converge to a likelihood above initialization.
- **Mixing Failure**: Jump block mobility is negligible compared to the Diffusion block.
- **Verdict**: $N = 40k$ and 500 iterations are insufficient for full SVCJ.
- **Requirement**: Big increase in particles or/and iterations.

Orthogonal MCMC (O-MCMC): Concept

- **Problem**: Standard PMMH chains often get stuck in local modes due to the high dimensionality of the SVCJ model.
- **Solution**: Run a population of parallel chains that interact to share information.
- **Algorithm Structure**:
  1. **Vertical Step (Local Exploration)**: Each chain runs independent PMMH steps (M_v iterations) to explore its local neighborhood using the Particle Filter.
  2. **Horizontal Step (Global Interaction)**: Chains exchange information using a **Differential Evolution** proposal:

$$\theta_{\text{prop}} = \theta_i + \gamma(\theta_a - \theta_b) + \epsilon$$

     Allows "bold jumps" across the parameter space based on population dispersion.

Implementation Details

Configuration ("Fast Prototyping")

- **Population**: $Z = 10$ parallel chains.
- **Particles**: $N_x = 2,000$ (Lower than PMMH for computational speed).
- **Vertical step**: $M_v = 10$ short bursts.
- **Schedule**: 100 cycles of alternating vertical/horizontal steps.
- **Total Effort**: $\approx 10,000$ likelihood evaluations.

Objective

- Test if population-based methods can find high-probability regions faster than a single tuned chain.
- Recover latent volatility dynamics under constraints.

Model Selection: Naive vs. O-MCMC

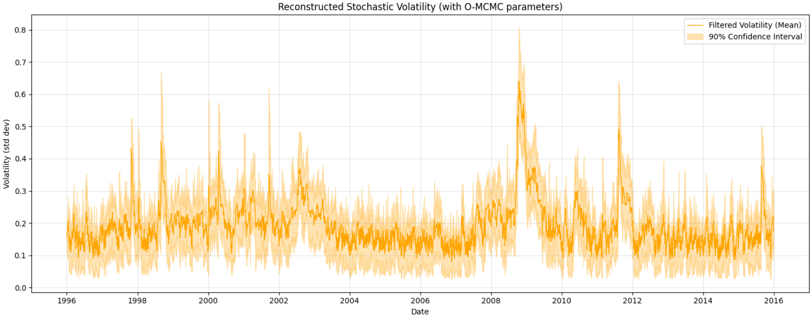
- **Naive Benchmark**: Geometric Brownian Motion (Constant Volatility).
- **O-MCMC**: Stochastic Volatility with Correlated Jumps.

Metric	**Naive Model**	**PMMH**	**O-MCMC**
Log-Likelihood	14,929.86	**16,132.44**	**16,099.53**
Fit Gain	–	**+1,202.58**	**+1,169.67**

Interpretation:

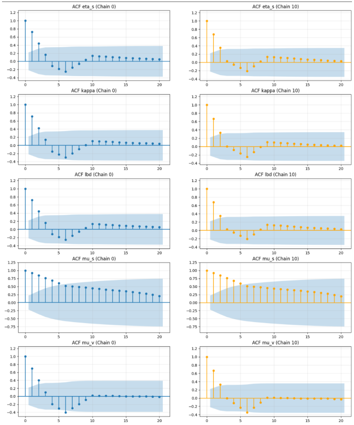
- The massive gain ($>$ 1000 points) provides "decisive evidence" (Bayes Factor) in favor of the SVCJ specification.
- Successfully captures stylized facts: Fat tails, Volatility Clustering, and Leverage Effect.

Latent State Reconstruction



- **Regime Detection**: Clearly identifies the 2008 crisis ($t \approx 3200$) with volatility peaking at $\sim 80\%$.
- **Uncertainty**: The 90% CI (orange band) naturally expands during turbulent periods, correctly quantifying risk.

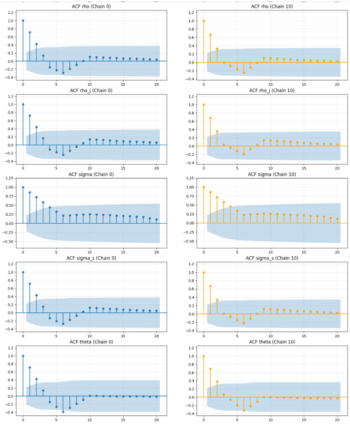
Diagnostics: The "Sticky Chain" Problem (1/2)



Observations

- **High Autocorrelation**: The visual evidence of slow decay in the ACF plots indicates "sticky" chains.
- **Root Cause**: With only $N = 2000$ particles, the estimator variance is high ($\text{Var}(\log \hat{L})$).

Diagnostics: The "Sticky Chain" Problem (2/2)



Mechanics of Failure

- **Effect**: High variance leads to a noisy acceptance ratio. The Metropolis-Hastings step rejects most proposals, causing chains to stay fixed (flat lines) for many cycles.

Global Conclusion

- The O-MCMC algorithm acted as a powerful **optimizer** (successfully finding the high-probability mode).
- However, it performed as a poor **sampler** (failed to quantify

Thanks!