Latent State and Parameter Estimation of Stochastic Volatility/Jump Models via Particle Filtering

Andrew Soane

A dissertation submitted to the Faculty of Commerce, University of Cape Town, in partial fulfilment of the requirements for the degree of Master of Philosophy.

June 8, 2018

MPhil in Mathematical Finance,
University of Cape Town.

Declaration

I declare that this dissertation is my own, unaided work. It is being submitted for the Degree of Master of Philosophy to the University of Cape Town. It has not been submitted before for any degree or examination to any other university.

Andrew Soane

June 8, 2018

Abstract

Particle filtering in stochastic volatility/jump models has gained significant attention in the last decade, with many distinguished researchers adding their contributions to this new field. Golightly (2009), Carvalho et al. (2010), Johannes et al. (2009) and Aihara et al. (2008) all attempt to extend the work of Pitt and Shephard (1999) and Liu and Chen (1998) to adapt particle filtering to latent state and parameter estimation in stochastic volatility/jump models. This project will review their extensions and compare their accuracy at filtering the Bates Stochastic volatility model. Additionally, this project will provide an overview of particle filtering and the various contributions over the last three decades. Finally, recommendations will be made as to how to improve the results of this paper and explore further research opportunities.

Contents

1. Introduction . . . 1
- 1.1 Filtering Problem . . . 1
- 1.2 Particle Filtering . . . 1
  - 1.2.1 Sequential Importance Sampling . . . 3
  - 1.2.2 Sequential Importance Resampling . . . 6
  - 1.2.3 Likelihood Factorisation Method . . . 8
- 1.3 Stochastic Volatility/Jump Models . . . 9
  - 1.3.1 Particle Filtering in Stochastic Volatility/Jump Models . . . 10
- 1.4 Aim . . . 11

2. Model and Methodology . . . 13
- 2.1 Tested Models . . . 13
  - 2.1.1 Algorithm Overview . . . 14
- 2.2 Latent State Estimation . . . 15
- 2.3 Parameter Learning . . . 16
  - 2.3.1 Uniform Initial Parameter Distribution . . . 17
  - 2.3.2 Bayesian Filtering Distribution . . . 18

3. Results . . . 19
- 3.1 Latent State Estimation . . . 19
- 3.2 Parameter Learning . . . 22
  - 3.2.1 Uniform Initial Parameter Distribution . . . 22
  - 3.2.2 Bayesian Filtering . . . 24

4. Conclusion and Further Research . . . 27

Bibliography . . . 27

A. Introductory Bayesian Inference . . . 31

B. Bayesian Filtering Distributions . . . 32
- B.1 Prior Distributions . . . 32
- B.2 Distribution of μ . . . 32
- B.3 Distribution of σ_z^2 . . . 33
- B.4 Distribution of κ . . . 33
- B.5 Distribution of θ . . . 34
- B.6 Sufficient Statistic and Parameter Propagation . . . 34

List of Figures

1.1 Filtered estimate of Z_t. 6
1.2 Filtered estimate of Z_t. 7

3.1 Filtered Z_t using the Uniform Initial Parameter method 20
3.2 Filtered Z_t in Bayesian filtering method. 21
3.3 Filtered Jump Process in Bayesian filtering method. 22
3.4 Filtered estimates of Θ using the Uniform Initial Parameter method . 24
3.5 Filtered estimates of parameter set 25
3.6 Estimate of filtering distributions of parameter set. 26

List of Tables

3.1 Model Parameters. 19
3.2 Filtered estimates of parameter set. 26

Chapter 1

Introduction

1.1 Filtering Problem

Latent state estimation and parameter learning is a common problem in many different spheres including GPS navigation, signal processing and stock market analysis. Filtering attempts to remove unwanted noise or extract drivers from an observed random process. More formally, the goal of filtering is to extract a set of latent states Z_t and sequentially estimate a set of parameters Θ from an observed process X_t, given known information about the process X_t and Z_t, namely the transition distribution $p(Z_t|Z_{t-1}, \Theta)$ and the measurement equation $p(X_t|Z_t, \Theta)$.

The well known Kalman Filter (Kalman et al.; 1960) has been used extensively in the context of quantitative finance, with many extensions being developed since its first introduction, namely the Extended Kalman Filter (Anderson and Moore; 1979) and the Unscented Kalman Filter (Julier and Uhlmann; 1997). These extensions have been developed to address the inability of the Kalman Filter to estimate latent states in non-linear, non-Gaussian models. Since the introduction of sequential Monte Carlo methods for latent state and parameter estimation such as the particle filter, researchers have shifted their attention from deterministic filters such as the Kalman filter and its extensions to these more flexible algorithms.

1.2 Particle Filtering

Particle filtering is a sequential Monte Carlo method for estimation of latent states and parameter learning in non-linear, non-Gaussian models. Gordon et al. (1993) introduce the first application of sequential Monte Carlo methods for latent state estimation in non-linear stochastic models. This was later improved upon by Liu and Chen (1998) and their work become known as particle filtering. It has gained a lot of attention in recent years due to its ability to accurately and efficiently estimate filtering distributions in complex models that its counter-part, the Kalman Filter

and its extensions, cannot do. Due to its increase in popularity, many authors have released extensions they hypothesize to increase its accuracy or efficiency. This dissertation will try to address these extensions and present a practical comparison.

convergence properties. The reader can refer to Crisan (2001) or Crisan (2009) for a rigorous introduction to stochastic filtering and the theoretical reasons behind using them.

At its core, particle filtering has four main steps, with various algorithms differing in the ordering of these steps: propagation, measurement, prediction and update. Given a time series of observed data these four steps allow for the posterior distribution of latent states to be derived. These latent states can include, for example, the stochastic volatility and jump processes, and the model parameters in a stochastic volatility/jump model.

The filtering problem can be formalised as follows:
Suppose we have a probability space $(\Omega, \mathcal{F}, \mathbb{P})$ and measurable random variables X and Z, the observable and signal/latent processes respectively. The goal of particle filtering is to determine the filtering distribution $p(Z_t|\sigma(X_{0:t}))$ in order to evaluate expectations of the form

$$\mathbb{E}[f(Z_t)|\sigma(X_{0:t})]$$

for some Borel function f where
$X_{0:t} = \{X_0, X_1, ..., X_t\}$ and $\sigma(X_{0:t})$ is a measure of information of the process X up to time t.

A degree of known information is needed at the outset of the algorithm, in particular

- The initial distribution of Z_0.
- The state transition equation $p(Z_t|Z_{t-1}, \Theta)$.
- The measurement equation $p(X_t|Z_t, \Theta)$.

The procedure of determining or approximating the filtering distribution is through a sequential Monte Carlo scheme designed to give an empirical approximation of the filtering distribution. The usefulness of particle filtering comes from showing that this approximation converges in distribution to the true filtering distribution as the number of particles tends to infinity. This is shown for a particular case of particle filtering algorithm, the Sequential Importance Sampling algorithm in section 1.2.1. The reader can refer to Crisan (2001) or Crisan (2009) for a more rigorous introduction to stochastic filtering and the theoretical reasons behind using it.

An example of an application of particle filtering could be in stock price identification where option prices are observable and the underlying stock prices are not. The observed process, X_t, could be a set of option prices, and Z_t could be the underlying stock price. This example is expanded on in S ection 1.2.1 and 1.2.2. One may also want to analyse the underlying drivers of the stock price, once observed, particle filtering can be applied to extracting latent states from the stock price such as the volatility and jump processes assumed in a stochastic volatility/jump model. This is the key focus of this project, as well as attempting to estimate the parameters in such models.

1.2.1 Sequential Importance Sampling

Given the previously mentioned prior knowledge, the simplest particle filtering algorithm, the Sequential Importance Sampling (SIS) algorithm (Geweke; 1989) can be developed. This algorithm is primarily used for latent state estimation and not parameter learning; therefore the presentation of the algorithm will assume a known set of parameters and omit the dependency on the parameter set Θ for notational ease. It relies on a choice of importance function, $\pi(Z_t|Z_{t-1}, X_t)$ in the measurement stage. Analogous to the importance function used in Monte Carlo importance sampling, this function serves as an approximation to the filtering distribution $p(Z_t|Z_{0:t-1}, X_{0:t})$ which in most cases will not have a closed form. An overview of the SIS algorithm is as follows:

- The algorithm is initialised by generating M samples of Z_0 from its initial distribution
- The given time horizon $[0, T]$ is discretised into a partition, $\{0 = t_0 < t_1 < t_2 < ... < t_N = T\}$ of length $N + 1$ with equidistant time intervals Δt, for the purpose of this paper, although not necessary. Then for each discrete time point, t_i, filtering is performed as follows:
  - Propagation: Generate M samples of Z_i according to $\pi(Z_i|Z_{i-1}, X_i)$ (referred to as particles).
  - Measurement: Evaluate $w_i^j \propto \frac{p(X_i|,Z_j^i)p(Z_j^i|Z_{i-1}^j)}{\pi(Z_i|Z_{i-1},X_i)}$, such that $\sum_{j=1}^{M} w_i^j = 1$.
  - Prediction: The filtering distribution can be approximated as

$$\hat{p}_{Z_t}(z|Z_{t-1}, X_t) = \sum_{j=1}^{M} w_t^j \delta_{Z_i^j}(z),$$

    where $\delta_{Z_i^j}(z)$ is the Dirac delta distribution.

More formally, the algorithm is represented in pseudo-code as

1. Initially $Z_0 \sim p(Z_0)$;

2. For $i = 1 : N$, $j = 1 : M$
   $Z_i^j \sim p(Z_i|Z_{i-1}^j)$
   $w_i^j \propto \frac{p(X_i|Z_i^j)p(Z_i^j|Z_{i-1}^j)}{\pi(Z_i|Z_{i-1},X_i)}$;

3. $\hat{p}_{Z_t}(z|Z_{t-1}, X_t) = \sum_{j=1}^{M} w_t^j \delta_{Z_i^j}(z)$.

It remains to show that the estimator of the filtering distribution $\hat{p}_{Z_t}(z|Z_{t-1}, X_t)$ converges to the true filtering distribution as the number of particles increases ($M \to \infty$). $\hat{p}_{Z_t}(z|Z_{t-1}, X_t)$ is called an optimal estimator of $p(Z_t|Z_{t-1}, X_t)$ (Chen et al.; 2003) as it optimally approximates the expectations of Borel functions of Z_t. Let $f : \mathbb{R} \to \mathbb{R}$ be a Borel function, with

$$\mathbb{E}[f(Z_t)|Z_{t-1}, X_t] = \int_{\mathbb{R}} f(Z_t)p(Z_t|Z_{t-1}, X_t)dZ_t. \tag{1.1}$$

Equation 1.1 can be evaluated exactly as follows:

$$\begin{aligned}\int_{\mathbb{R}} f(Z_t)p(Z_t|Z_{t-1}, X_t)dZ_t &= \int_{\mathbb{R}} f(Z_t)\frac{p(Z_t|Z_{t-1}, X_t)}{\pi(Z_t, Z_{t-1}, X_t)}\pi(Z_t, Z_{t-1}, X_t)dZ_t \\ &= \int_{\mathbb{R}} f(Z_t)\frac{w(Z_t)}{p(X_t)}\pi(Z_t|Z_{t-1}, X_t)dZ_t,\end{aligned}$$

where

$$\begin{aligned}w(Z_t) &= \frac{p(Z_t|Z_{t-1}, X_t)p(X_t)}{\pi(Z_t|Z_{t-1}, X_t)} \\ &\propto \frac{p(X_t|Z_t)p(Z_t|Z_{t-1})}{\pi(Z_t|Z_{t-1}, X_t)}\end{aligned}$$

and

$$\begin{aligned}p(X_t) &= \int_{\mathbb{R}} p(X_t|Z_t, Z_{t-1})p(Z_t|Z_{t-1})dZ_t \\ &= \int_{\mathbb{R}} w(Z_t)\pi(Z_t|Z_{t-1}, X_t)dZ_t \\ \Rightarrow \int_{\mathbb{R}} f(Z_t)p(Z_t|Z_{t-1}, X_t)dZ_t &= \frac{\mathbb{E}_\pi[f(Z_t)w(Z_t)]}{\mathbb{E}_\pi[w(Z_t)]}.\end{aligned}$$

Generating M samples of Z_t according to $\pi(Z_t|Z_{t-1}, X_t)$ allows a Monte Carlo approximation of the expectations as

$$\int_{\mathbb{R}} f(Z_t)p(Z_t|Z_{t-1}, X_t)dZ_t \approx \sum_{j=1}^{M} f(Z_t^j)\hat{w}^j(Z_t),$$

where $\hat{w}^j(Z_t) = \frac{w(Z_t^j)}{\sum_{j=1}^{M} w(Z_t^j)}$.

Alternatively, the optimal estimator, $\hat{p}(Z_t|Z_{t-1}, X_t)$ can be used to approximate the expectation:

$$\int_{Z_t} f(Z_t)p(Z_t|Z_{t-1}, X_t)dZ_t \approx \int_{Z_t} f(Z_t) \sum_{j=1}^{M} w_t^j \delta_{Z_i^j}(Z_t)dZ_t$$

$$= \sum_{j=1}^{M} w_t^j f(Z_t^j).$$

An important concept that has significant attention paid to it in the literature on particle filtering, that is the degeneracy of an algorithm. This is defined as the fast convergence of a small group of particles to a particular value leaving the majority of importance weights equal to zero. While this may appear beneficial, it is not, as the algorithm needs to collect information about the hidden processes through time and if hidden states are already essentially fixed, no added information can be added to the system and the estimates will be inaccurate.

One method of degeneracy reduction is by the choice of importance function, Zaritskii et al. (1975) propose the use of the optimal importance function, the distribution that yields minimum variance of the importance weights, w_t. This optimal distribution is found to be exactly the state transition equation $p(Z_t|Z_{t-1})$ (proof stated in Zaritskii et al. (1975)), reducing the measurement of prediction weights to

$$w_i^j \propto p(X_i|X_{i-1}, Z_{i-1}^j).$$

An example of a filtering problem, as previously mentioned, is in identifying the level of a stock price given a series of option prices over time. The SIS algorithm can be applied to this problem in the following way: The observed process X_t is a series of option prices with fixed strike K and maturity T. The latent state variable Z_t is the log stock price underlying the set of options, which for this example, is unobservable. The measurement equation is the Black-Scholes option pricing formula which is denoted by the function $\mathcal{C}$. The transition density, $p(Z_i|Z_{i-1})$ is the Euler transition density of geometric Brownian Motion parametrised as:

$$Z_i = Z_{i-1} + \left(r - \tfrac{1}{2}\sigma^2\right)\Delta t + \sigma\sqrt{\Delta t}\epsilon \tag{1.2}$$

where $\epsilon \sim \mathcal{N}(0, 1)$

$Z_0 = log(S_0)$

Assuming call prices follow a random walk process with Gaussian noise and centred around their Black-Scholes price. The measurement equation equates to evaluating the Gaussian density at the each particle. For the purpose of this examples,

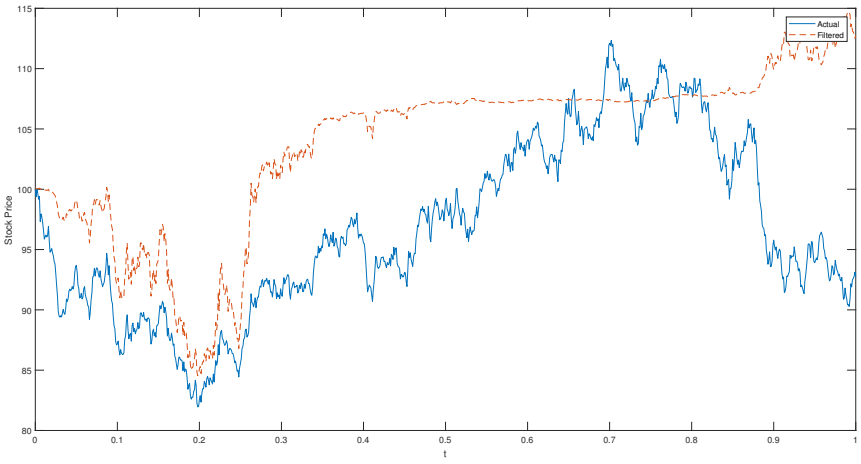
the following parameters were used:
$r = 0.05; \sigma = 0.2; S_0 = 100; \Delta t = \frac{1}{365}; K = 100; T = 1$

1. Initialise: $Z_0 = \log(100)$ and $C_0 = \mathcal{C}(Z_0; K, T)$,
2. For $i = 1 : N$, $j = 1 : M$
   $Z_i^j \sim p(Z_i | Z_{i-1}^j)$;
   $C_i^j = \mathcal{C}(Z_i^j; K, T)$;
   $w_i^j \propto \phi(C_i; C_i^j, 1)$.

The filtered stock price is estimated by taking the expectation of the particles according to their empirical distribution

$$Z_i = \sum_{j=1}^{M} w_i^j Z_i^j.$$

Fig. 1.1: Filtered estimate of Z_t.



The above figure shows the filtered estimate of Z_t in orange while the blue line indicates the true value. As can be seen the filtered estimate tracks the true value until a quarter of the way through, at which point the estimate no longer gathers information from the observed process and degenerates to an incorrect estimate.

1.2.2 Sequential Importance Resampling

The sequential importance resampling (SIR) algorithm was first developed by Rubin (1988), this algorithm was invented to deal with one major flaw in the SIS algorithm, particle degeneracy. The SIR algorithm adds an additional step to the SIS

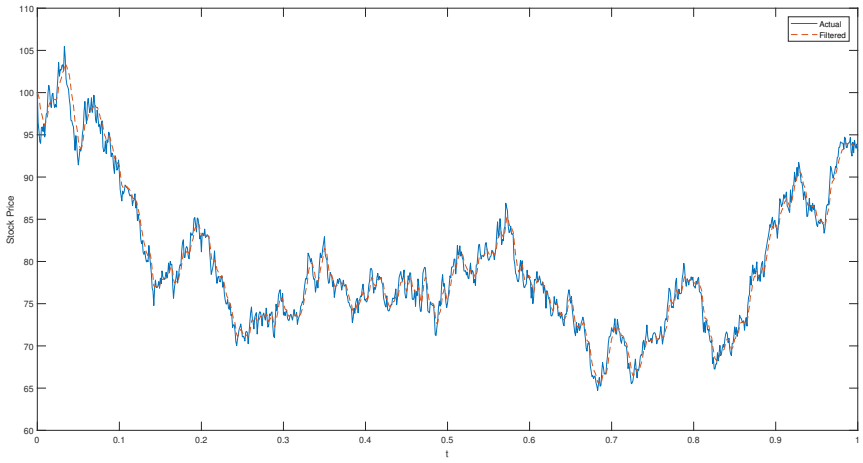
algorithm, a re-weighting of the particles according to the prediction weights, completing the four particle filtering steps. This then causes the algorithm to discard particles with little contribution to the likelihood. This implies that latent state estimation is done on a less variable and more accurate swarm of particles. Essentially the particles are directed to their correct distribution, as in the SIS algorithm, the entire swarm would converge to a single estimate without any further knowledge of the observable process. In order to illustrate the difference in the SIS and SIR algorithm, the example of filtering the log stock price Z_t is used. The SIR algorithm is as follows

1. Initialise: $Z_0 = \log(100)$ and $C_0 = \mathcal{C}(Z_0)$;

2. For $i = 1 : N$, $j = 1 : M$
   $Z_i^j \sim p(Z_i|Z_{i-1}^j)$;
   $C_i^j = \mathcal{C}(Z_i^j)$;
   $w_i^j \propto \phi(C_i; C_i^j, 1)$;
   Resample $\{Z_i^j\}_{j=1}^M$ according to $\{W_i^j\}_{j=1}^M$.

The filtered stock price calculated by taking a mean of the particles at each time step,

Fig. 1.2: Filtered estimate of Z_t.



The figure above shows the filtered estimate of Z_t in orange with the true value being in Blue. As can be seen, the filtered estimate tracks the true stock price more accurately.

1.2.3 Likelihood Factorisation Method

Johannes et al. (2006) and Carvalho et al. (2010) extend the work of Storvik (2002) who used the likelihood factorisation theorem to decompose the likelihood function to a function of model parameters and latent states, the contribution of the former authors will be developed in later sections. More formally:

$$p(\Theta, s_t, Z_{0:t}|X_{0:t}) = p(\Theta|s_t)p(s_t, Z_t|X_t),$$

Where s_t is a set of sufficient statistics for Θ based off (X_t, Z_t).
This allows the joint learning problem to be decomposed into two problems, latent state estimation and parameter learning. It is noted that the entire history of Z_t is no longer needed when utilising sufficient statistics, by definition. This reduces the dimensionality of the filtering problem as the algorithm is now designed to track just the current value of latent state processes and not the entire history.

Latent State Estimation According to Pitt and Shephard (1999) and Storvik (2002) with enhancement from Johannes et al. (2006) and Johannes et al. (2009), latent state estimation is optimised when resampling is performed pre-propagation of latent states. The motivation for this, as opposed to the resampling step of the SIR algorithm, is that latent states can be propagated from a distribution with more informative conditions. This method of particle filtering is further motivated by Carvalho et al. (2010), asserting that sufficient statistics will better represent the parameter set in this method as opposed to Storvik's filter.

Parameter Learning Parameter learning in the context of particle filtering remains a difficult task and has had substantial attention paid to it by experts in the field of filtering and machine learning. Sequential estimation of model parameters in a Monte Carlo framework relies on parameter particles to be propagated in such a way that they explore the majority of their sample space and are then directed towards their true value. This implies that model parameters need to be perturbed at each iteration of the algorithm and then resampled according to their likelihood. There are three popular methods used in the literature of sequential parameter estimation to perturb these parameters. They are:

- Assuming a suitable sample space for the parameter set, the initial parameter estimates can then be sampled uniformly inside this space. The particle filter then sequentially updates the parameter set based on its updating information. The parameter set is not propagated during the algorithm, it is simply

updated with the intention of the updating information causing a convergence in the parameter set to their true values. This concept is developed in more detail in Section 2.3.1.

- Perturbation of parameters can be done via a kernel smoothing density. This method is explored in detail in Liu and West (2001), however it is outside of the scope of this project. A benefit to using a smoothing density is that it restricts parameter estimates from exploding in the filtering algorithm. This can cause issues when estimating sensitive parameters such as the mean reversion rate and volatility of variance parameters in the Heston model. Golightly (2009) proposes a simulation filter prior to his work on Bayesian filtering distributions. The simulation filter makes use of the parameters' kernel smoothing densities. The author's results show that the Bayesian filtering method, although more sensitive to the amount and accuracy of prior information, did provide more accurate estimates of model parameters in a stochastic volatility/jump model.

- The third and most viable method (Golightly (2009), Jacquier et al. (2016), Johannes et al. (2009)) is the method of deriving the Bayesian posterior distributions of the parameter set given a set of prior distributions. This method allows propagation of parameters with knowledge of the entire history of both the latent and observable states, in the form of a set of sufficient statistics. This method still allows the propagated particles to explore the parameter sample space but with greater density in the higher likelihood areas.

1.3 Stochastic Volatility/Jump Models

Consider the SVJJ model of Duffie et al. (2000)

$$\begin{aligned} dX_t &= a(X_t, Z_t, t)dt + b(X_t, Z_t, t)dW_t^1 + dq_t^1, \\ dZ_t &= c(Z_t, t)dt + d(Z_t, t)dW_t^2 + dq_t^2, \end{aligned} \tag{1.3}$$

where W_t^1 and W_t^2 are correlated Brownian motions, and q_t^1 and q_t^2 represent the jump processes.
These can be compound Poisson processes with stochastic or deterministic intensities.
The Bates stochastic volatility model of the log-stock price can be represented by the following parametrisation of Equation 1.3

$$a(X_t, Z_t, t) = (\mu - \lambda m^j - \tfrac{1}{2}Z_t),$$

$$b(X_t, Z_t, t) = \sqrt{Z_t},$$
$$c(Z_t, t) = \kappa(\theta - Z_t),$$
$$d(Z_t, t) = \sigma_Z \sqrt{Z_t},$$

$$q_t^1 = \sum_{i=0}^{N(t)} V_i,$$

where $N(t)$ is a Poisson process with intensity λ and $V_i \sim \mathcal{N}(\mu, \sigma^2)$. This model will be considered throughout this project.

1.3.1 Particle Filtering in Stochastic Volatility/Jump Models

Due to a particle filter's flexibility and efficiency in latent state estimation and parameter learning, it has recently gained substantial attention in the literature for its use in stochastic volatility/jump diffusion models. Golightly (2009) attempts to filter a jump model with the following parametrisation according to Equation (1.3)

$$a(X_t, Z_t, t) = \mu,$$
$$b(X_t, Z_t, t) = \sqrt{Z_t},$$
$$c(Z_t, t) = \kappa(\theta - Z_t),$$
$$d(Z_t, t) = \sigma_Z \sqrt{Z_t},$$
$$q_t^1 = \sum_{i=0}^{N(t)} V_i^x, \quad N(t) \sim \mathcal{P}(\lambda), \quad V_i^X \sim \mathcal{N}(\mu_x, \sigma_x^2),$$
$$q_t^2 = \sum_{i=0}^{N(t)} V_i^Z, \quad V_i^Z \sim \exp(\mu_z).$$

The author uses a Sequential Importance Resampling method coupled with Bayesian filtering distributions to estimate the latent state Z_t and parameters $\Theta = \{\mu, \mu_X, \sigma_x, \lambda, \kappa, \theta, \sigma_z, \mu_z\}$. Note, the paper made no attempt at filtering the jump processes q_t^1 and q_t^2. Instead, the jump processes were marginalised from the measurement equation, meaning the importance weights had no dependence on the jump times and sizes. This method is discussed in later sections. Storvik (2002) developed the now well-known Storvik's filter as an extension of the auxiliary particle filter developed by Pitt and Shephard (1999). This filter was later applied to a stochastic volatility framework in Johannes et al. (2006). An appraisal of the authors' results is carried out in later sections with there being subtle nuances in the algorithm developed in this paper.

Jacquier et al. (2016) attempts to use Bayesian filtering to extract the entire latent state set and estimate parameters in Merton's jump model. This paper, unlike Golightly (2009) and Johannes et al. (2006) attempts to filter the jump processes using the full likelihood function. These results indicate particle filtering is more than capable of identifying jumps in a series of stock prices. This implies that further research could be conducted on the effectiveness of particle filtering in predicting jumps and its accuracy in comparison with other jump detection tests.

Cited areas of weakness

It is common in particle filtering literature (Golightly (2009), Jacquier et al. (2016), Johannes et al. (2009), Aihara et al. (2008)) for the mean reversion rate parameter and the volatility of variance parameter in the Heston and Bates model to be difficult to estimate due to their large contribution to the likelihood and transition distributions. Intuitively, it is expected that the parameter particles will be highly volatile in the early stages of the algorithm, regardless of parameter learning method chosen. If, for example, the volatility of variance particles remain directionless for a certain number of iterations, this will have a knock-on affect as it will cause inaccurate estimates of the hidden volatility process which will cause inaccuracies in the sufficient statistic set. It is therefore hypothesized that when estimating latent states and parameters simultaneously, the algorithm will either need a large number of observable variables to infer the hidden states or a large amount of data for the parameter estimates to have time to slow down and converge to their true values.

The first suggestion above can be done by incorporating option prices into the algorithm as a form of observable process. This implies the observable set of processes will consist of a series of option price data with varying strikes and maturities as well as the underlying stock price. It is expected that option price data will enhance the extraction of the volatility process as option prices give better indicators of the underlying's volatility than the underlying itself.

The second suggestion will mean fewer instruments would need to be tracked to make inference about an underlying stock price, however, the prices would need to be observed over a longer period of time.

1.4 Aim

The aim of this dissertation is to perform a simulation study on the effectiveness of particle filtering in stochastic volatility/jump models. In particular, to compare the effectiveness of the various particle filtering algorithms. This will be done in

a chronological fashion, testing methods first proposed in the literature and comparing them with their more optimised, developed algorithms which have recently been introduced.

Section 2 will lay out a foundation for the particle filtering algorithms to be used as well as present the model which will be the primary focus of the dissertation. Section 3 will present the results of the simulation study and attempt to tie the findings together such that the research questions can be answered. Section 4 will give a discussion with backing from the literature based on the results and propose further research areas.

Chapter 2

Model and Methodology

2.1 Tested Models

The model tested in this paper is the Bates stochastic volatility model, first introduced by Bates (1996). The choice of model is justified by a vast library of literature arguing its validity in modelling stock prices (Bates (2003), Bakshi et al. (1997), Duffie et al. (2000), to cite a few).

This model extends the commonly used Heston model by adding a random jump component to the stock price. To remain consistent with the notation in 1.2, the model is presented as follows

$$dX_t = (\mu - \lambda m^j - \tfrac{1}{2}Z_t)dt + \sqrt{Z_t}dW_t^1 + dq_t \tag{2.1}$$

$$dZ_t = \kappa(\theta - Z_t)dt + \sigma_z\sqrt{Z_t}dW_t^2, \tag{2.2}$$

where X_t is the log-stock price.

$$q_t = \sum_{i=0}^{N(t)} V_i,$$

$$N(t) \sim \mathcal{P}(\lambda),$$

$$V_i \sim \mathcal{N}(\mu_x, \sigma_x^2),$$

$$m^j = e^{\mu_x + \frac{1}{2}\sigma_x^2} - 1,$$

where it is assumed that W_t^1 and W_t^2 are uncorrelated. This assumption, although restricting, is required as it is well documented (Golightly (2009), Johannes et al. (2009)) that estimation of the correlation coefficient between Brownian Motions is difficult in the context of Sequential Monte Carlo estimation.

Aihara et al. (2008) attempt to estimate the volatility and model parameters, conditional on jumps occurring. An implementation of their algorithm is presented in Section 3.2.1. Their algorithm, by assuming prior uniformity of parameter values sacrifices more accuracy than is desirable, as evident in their results.

Golightly (2009) implements a simulation filter as a well as a particle filter on an extended Bates model. Their algorithm will inform the results presented in this dissertation, with there being subtleties in the model formulation and Bayesian parameter distributions. The simulation filter extends the work of Aihara et al. (2008) by perturbing the parameter values using a smoothed kernel density form of $p(\Theta, Z_{0:t})$. The purpose of this replacement is to reduce particle degeneracy in the filtering algorithm. Their simulation filter performs optimally with large sample sizes, this is not the case in particle filtering as accurate results can be achieved with relatively small sample sizes due to the efficient use of prior and posterior information utilised by the Bayesian parameter distributions. Storvik (2002) motivates the use of Bayesian parameter distributions as it improves the convergence rate of the filtered parameters to their true value. This suggests that both the simulation filter of Golightly (2009) and the particle filter of Aihara et al. (2008) could have been improved by adding a measure of dependence on previous parameter values, latent states or observed states to the parameter transition distributions.

Furthermore, Storvik (2002) uses the property of sufficient statistics to factorise the likelihood function in order to break-down the filtering problem into two components: updating the sufficient statistics and latent states based on observed data and then updating parameter values based on these updated variables. Storvik's filter has been given a lot of attention, in particular Johannes et al. (2006) modify Storvik's filter with substantial motivation as to its improvement. In particular, they amend the resampling step in the particle filtering algorithm to occur at the start of the algorithm at pre-propagation as opposed to the Sequential Importance Resampling algorithm, which resamples after propagation of latent states and parameter values. This idea was originally discussed by Pitt and Shephard (1999). Its motivation is that it will generally only propagate high likelihood particles as opposed to blindly propagating particles and then measuring their likelihood. Secondly, Johannes et al. (2006) coerce dependence in the transition distribution of latent states, that is; use $p(Z_{t+1}|Z_t, X_{t+1}, \Theta)$ as opposed to $p(Z_{t+1}|Z_t, \Theta)$ to propagate latent state particles. This is to create a lower variance estimate of the latent state variable as particles are moved towards values that are most likely to have generated X_{t+1}.

2.1.1 Algorithm Overview

The goal of particle filtering is to estimate a vector of latent states and parameters (Z_t, Θ) given an observed process X_t over a certain time period. Using the foundation set out in Johannes et al. (2006), this algorithm will rely on updating a set of sufficient statistics at each time step in order to fix the dimensionality of the algo-

rithm, rendering it more efficient, furthermore; the resampling of particles will be done at the start of the algorithm for reasons stated above.

2.2 Latent State Estimation

In order to derive transition and likelihood distributions, the discretised form of the Bates model is considered, that is, discretise the time interval $[0, T]$ into a partition $\{0 = t_0, t_1, ..., t_N = T\}$ with N equidistant spaces, Δt.
Let $(X_i, Z_i) = (X_{i\Delta t}, Z_{i\Delta t}) \quad t \in [0, T], \quad i \in \{1, 2, .., N\}$.
The discretised Bates model has the following form

$$X_{i+1} = X_i + (\mu - \lambda m^j - \tfrac{1}{2}Z_i)\Delta t + \sqrt{Z_i \Delta t} W^1 + J_{i+1} V_{i+1} \tag{2.3}$$

$$Z_{i+1} = Z_i + \kappa(\theta - Z_i)\Delta t + \sigma_z \sqrt{Z_i \Delta t} W^2 \tag{2.4}$$

where it is assumed that only one jump can occur in a short time interval of Δt. Meaning the probability of a jump occurring in a sufficiently small interval of length Δt is $\lambda \Delta t$. The jump time variable is modelled by an independent Bernouli random variable with jump probability $\lambda \Delta t$,

$$\Rightarrow J_i \sim \text{Ber}(\lambda \Delta t)$$

$$W^j \sim \mathcal{N}(0, 1) \quad j = 1, 2.$$

This then allows the transition density and measurement equations to be computed as

$$p(X_{i+1}|Z_i, J_{i+1}, V_{i+1}, X_i, \Theta) = \phi(X_i + (\mu - \lambda m^j - \tfrac{1}{2}Z_i)\Delta t + J_{i+1}V_{i+1}, Z_i \Delta t)$$

$$p(Z_{i+1}|Z_i, \Theta) = \phi(Z_i + \kappa(\theta - Z_i)\Delta t, \sigma_z^2 Z_i \Delta t),$$

where $\phi(a, b)$ is the Gaussian distribution with mean and variance a and b respectively. To maintain consistency with current literature on latent state estimation in stochastic volatility models, there are two points that need to be discussed at this stage:

- Firstly, in accordance with Aihara et al. (2008), the filtering of the stochastic volatility state variable Z_t can be optimised in terms of accuracy and computation time if the likelihood is marginalised with respect to the jump times and sizes. Consider Equation (2.3), then

$$p(X_{i+1}|Z_i, J_{i+1} = 1, \Theta) = \phi(X_i + (\mu - \lambda m^j - \tfrac{1}{2}Z_i)\Delta t + \mu_x, Z_i \Delta t + \sigma_x^2),$$

$$p(X_{i+1}|Z_i, J_{i+1} = 0, \Theta) = \phi(X_i + (\mu - \lambda m^j - \tfrac{1}{2}Z_i)\Delta t, Z_i \Delta t).$$

Therefore the likelihood function can be marginalised of the jump times and sizes. Therefore restricting latent state estimation to a one-dimensional case rather than a three-dimensional problem. This know as Rao-Blackwallization in the literature. This marginalised likelihood will be considered when testing the uniform initial parameter distribution method for latent states estimation and parameter learning.

- Secondly, in accordance with Johannes et al. (2006), the transition distribution is coerced to have dependence on the current value of the observed process. This requires the transition equation of Z_{t+1} to have dependence on $X_{t:t+1}$; which can be accomplished by modifying the discretisation in Equation (2.3) to use Z_{t+1} in the drift term instead of Z_t, as is common practice in the literature on particle filtering (Johannes et al. (2009), Aihara et al. (2008)). That is

$$p(Z_{t+1}|Z_t, X_{t:t+1}, J_{t+1}, V_{t+1}, \Theta) \propto p(Z_{t+1}|Z_t, \Theta)p(X_{t+1}|X_t, Z_t, J_{t+1}, V_{t+1}, \Theta),$$

$$\implies p(Z_{t+1}|Z_t, X_{t:t+1}, J_{t+1}, V_{t+1}, \Theta) \propto$$

$$\exp\left(-\frac{1}{2}\frac{\left(Z_{t+1} - \frac{Z_t+\kappa(\theta-Z_t)\Delta t+\frac{1}{2}\Delta t\sigma_z^2(\Delta X_t-(\mu-\lambda m^j)\Delta t-J_{t+1}V_{t+1})}{1+\frac{1}{4}\sigma_z^2(\Delta t)^2}\right)^2}{\frac{\sigma_z^2 Z_t\Delta t}{1+\frac{1}{4}\sigma_Z^2(\Delta t)^2}}\right).$$

It is noted from the above posterior distribution, with the assumption of zero correlation that the reduction in variance caused by the coercion of dependence on $X_{t:t+1}$ is of an order of $(\Delta t)^2$. Therefore, for the sake of computational efficiency, this reduction will be taken as negligible and the unconditional transition density $p(Z_{t+1}|Z_t, \Theta)$ will be used in the propagation step of the algorithm.

2.3 Parameter Learning

In order to estimate the model parameters sequentially, the parameters need a component of noise to them, allowing them to explore their sample space and gain information as to where they are most likely to occur. Given this, there are a number of methods for accurately estimating this parameter set.

The parameter learning is done via three methods proposed in the literature on stochastic volatility models. These methods are:

1. Uniform Initial Parameter Distribution;
2. Maximum Likelihood Update Method; and

3. Bayesian Filtering Distribution Method.

The second method will be omitted from this dissertation as it can be shown that the Bayesian posterior parameter estimates are a function of the maximum likelihood estimates.

2.3.1 Uniform Initial Parameter Distribution

This method is proposed in Aihara et al. (2008) and relies on the measurement and prediction stage of the particle filter to solely estimate the parameter values. An overview of the algorithm is as follows, with N time steps and M particles as before:

- Initially simulate M samples of $\Theta \sim \mathcal{U}$, over a reasonable sample space centred around a parameter estimate[1];
- Then at each discrete time step:
  - Propagate M samples of the latent state Z_t according to $p(Z_t|Z_{t-1}, \Theta)$
  - Measure the influence of each particle by evaluating the likelihood of observing each particle. That is, calculate the importance weights
  $$w_t \propto p(X_t|Z_t, \Theta);$$
  - Resample Z_t and Θ according to the importance weights w_t.

More formally, the algorithm is presented as pseudo-code:

1. At $i = 0$, $\Theta \sim \mathcal{U}$;
2. For $j = 1 : N$, $i = 1 : M$

$$\begin{aligned} Z_i^j &\sim p(Z_j, Z_{i-1}^j, \Theta^i); \\ w_i^j &\propto p(X_i|Z_i^j, \Theta^i); \\ u^j &\sim \text{multi}\{\{w_i^j\}_{j=0}^M\}; \\ (Z_i^j, \Theta^i) &= (Z_i^{u^j}, \Theta^{u^j}). \end{aligned}$$

where multi$\{w_i\}$ is the multinomial distribution with vector of probabilities w_i. This algorithm, although easy to implement and computationally inexpensive, does not capture the variability that random parameters have on the process, while it performs well in estimating the stochastic volatility state variable, there is no convergence present in the estimates of the model parameters. This will be evident when the results of this algorithm are presented in the next section.

[1] For this simulation study, the true values are used as the centre.

2.3.2 Bayesian Filtering Distribution

This method requires a set of prior distributions for Θ, conjugate prior distributions are used to allow the maintenance of distributional form from prior to posterior. The posterior distributions are then calculated with respect to the entire observed process and state processes. This gives rise to the method of sufficient statistics proposed by Storvik (2002). This method is developed as follows:

1. Initially, parameters are propagated using their prior distributions;
2. Proceeding parameter estimates are then propagated using their posterior distribution, $p(\Theta, s_t)$ where s_t is a set of sufficient statistics for Θ;
3. s_t is updated at each time step according to a recursive relation of the form $s_{t+1} = \mathcal{S}(s_t, X_{t+1}, Z_{t+1}, J_{t+1}, V_{t+1}, \Theta)$.

The presentation of conjugate prior distributions, posterior distributions and sufficient statistics is included in Appendix B.

Furthermore resampling latent states and parameters as the first stage of the algorithm will be utilised in the final results. This suggests a final algorithm as

1. At $i = 0$: $\Theta \sim p(\Theta)$;
2. For $j = 1 : N$, $i = 1 : M$

$$w_j^i \propto p(X_j | Z_{j-1}^i, X_{j-1}, J_j^i, V_j^i, \Theta^i);$$

$$u^j \sim \text{multi}\{\{w_i^j\}_{j=0}^M\};$$

$$Z_j \sim p(Z_j | Z_{j-1}^{u^i}, \Theta^{u^i});$$

$$J_{j+1}^i \sim p(J_j | \Theta^{u^i}) \qquad V_{j+1}^i \sim p(V_j | \Theta^{u^i});$$

$$s_j^i = \mathcal{S}(s_{j-1}^{u^i}, X_j, Z_j^i, J_j^i, V_j^i);$$

$$\Theta^i \sim p(\Theta | s_j).$$

Chapter 3

Results

3.1 Latent State Estimation

The requirements for estimating the latent states in the Bates model are:

- The transition distribution of (Z_t, J_t, V_t)
- The likelihood function, $p(X_t|Z_{t-1}, X_{t-1}, J_t, V_t, \Theta)$

As shown in the previous section, both the transition distribution of Z_t and the likelihood distribution have Gaussian distributions. Due to the independence of jump times and sizes with their history as well as the latent and observable process, these states are simply propagated with an updated set of parameters and then resampled according to their likelihood. The results of these estimates are included in the proceeding figures. It is noted that latent state estimation is performed concurrently with parameter learning in the uniform initial parameter algorithm, causing the effects of the inaccuracies in parameter learning to present themselves in the results of latent state estimation. The simulation studies below were based on the following mesh parameters $T = 5$, $M = 1000$, $\Delta = 0.001$ and the model parameters in Table 3.1:

Tab. 3.1: Model Parameters.

μ	θ	κ	σ_z	λ	μ_x	σ_x
0.1	0.2	5	0.4	4	-0.1	0.1

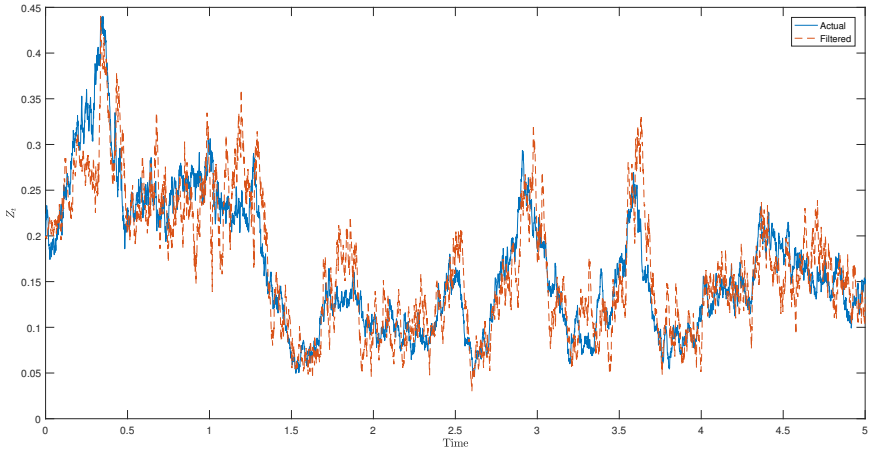


Fig. 3.1: Filtered Z_t using the Uniform Initial Parameter method

Firstly the uniform initial parameter distribution method is tested. Figure 3.1 above shows the filtered estimate of the state variable Z_t. As opposed to the Bayesian filtering method, it is clear that the uniform initial parameter method shows convergence in the latent state variables very rapidly. This is consistent with the design of the two algorithms. The parameters are perturbed uniformly at the start of the algorithm and then directed towards their true value. The direction comes from the measurement step of the algorithm, which, in the case of the uniform initial parameter, is purely based off of the likelihood of the latent states with respect to the observed state. That is, the perturbed parameters have very little contribution to the likelihood, meaning that, in essence, the algorithm first corrects the filtered estimate of the latent states and then attempts to correct the parameters. In contrast, as will be shown, the Bayesian filtering method attempts to correct both latent states and parameters simultaneously, causing highly volatile estimates in the initial stages of the algorithm.

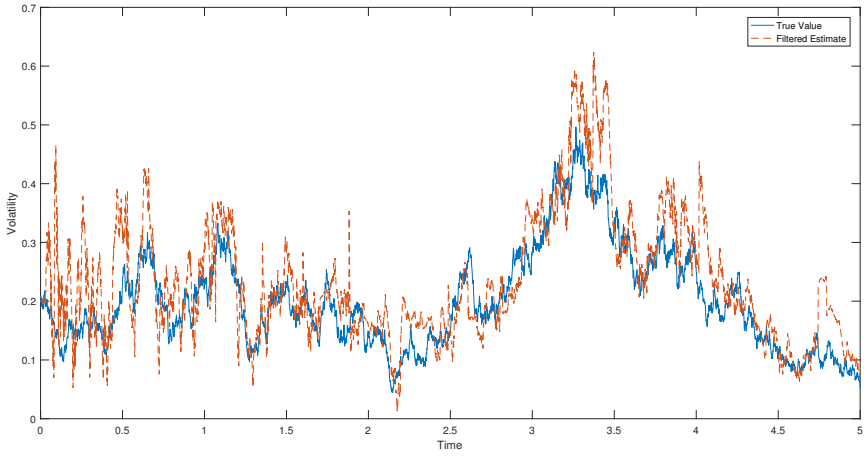


Fig. 3.2: Filtered Z_t in Bayesian filtering method.

Figure 3.1 shows the filtered estimate of Z_t using the Bayesian filtering method. As is mentioned above, the estimate of Z_t is extremely volatile in the early stages of the algorithm, however it does begin to converge over time. This is due to the algorithm attempting to estimate parameters and latent states concurrently with only one source of information. At its core, the algorithm, perturbs both latent states and parameters according to some reasonable sample space. Due to the lack of information in the beginning, these sample spaces are extremely large causing the estimates to bare no resemblance of their true value. However as more information enters the system, these sample spaces shrink and the estimates begin to converge.

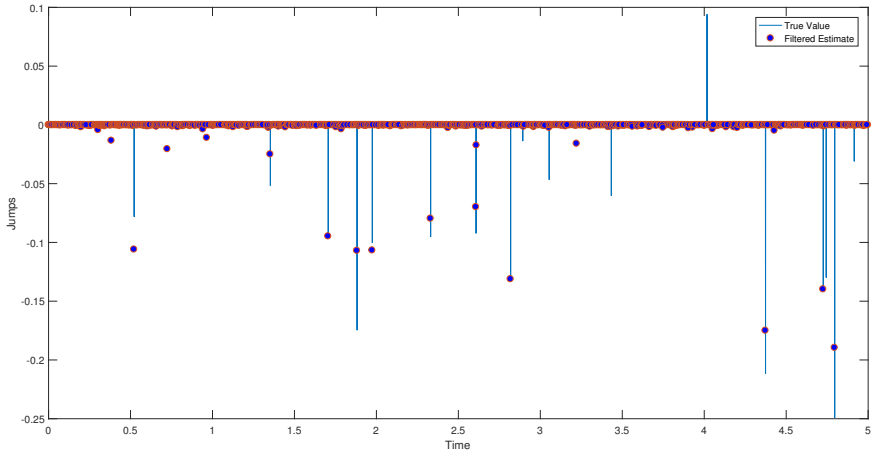


Fig. 3.3: Filtered Jump Process in Bayesian filtering method.

Finally, Figure 3.1 shows the filtered estimate of the jump process. The thing to note is that the algorithm identifies the jump times accurately, this is evident by the position of the dots on top of the solid lines for almost all solid lines. The jump sizes are somewhat less accurate as they quite often under or over estimate the true size. It is noted, again, that the jump times and sizes in the discretised Bates model do not have a transition distribution with dependence on the previous values, latent states or observed states of the process. In essence, the estimate of the jump times and sizes is achieved in a similar way as the estimate of the parameters is done in the uniform initial parameter method in that these latent variables are propagated using a distribution independent of all other latent variables (including themselves). This lack of dependence in the propagation stage may explain the disparity in the filtered jump sizes and the true sizes.

3.2 Parameter Learning

3.2.1 Uniform Initial Parameter Distribution

The particles are initially scattered uniformly over a reasonable sample space, centred around an empirical estimate of their true value (eg. maximum likelihood estimate) and then directed to their true value. This method relies on the measurement step accurately quantifying the likelihood of observing each parameter and latent state. This method has two main drawbacks which affect its accuracy, as will be apparent with the results.

The first drawback is that, the propogation of parameters is done by simply using the estimate of the posterior distribution of latent states at the previous time step's. More formally: At time t, Θ is propagated using $\hat{p}(Z_{t-1}, \Theta|X_{0:t-1}) = \sum_{j=0}^{t-1} w_{t-1}^{j} \delta(Z_{t-1}^{j}, \Theta^{j})$. This implies that no information about the current value of the observable or latent processes is being added to the propagation of parameters. This is brought up as a complication to parameter learning via particle filtering in Carvalho et al. (2010) as it is asserted that the approximation of $p(\Theta|X_{0:t})$ will be a degenerate distribution with few particles contributing to its estimation. This is evident in the plots of filtered parameters as the straight line indicates that particles have degenerated towards that value and the estimated posterior distribution has no variance.

The second drawback occurs when the observable process consists of a single observation of either a stock price or derivative price. The likelihood function will then be unable to detect a clear path for the swarm of particles due to the non-linearities in the measurement equation.

It is further noted that the posterior distribution of latent states may be multi-modal in complex models such as the Bates model, making estimation more cumbersome, especially in a Monte Carlo Markov Chain context. Particle swarms may tend towards an area of high likelihood due to the multi-modal behaviour of the posterior. Figure 3.2.1 shows the estimates of the parameter set $\Theta = \{\theta, \kappa, \sigma_z, \mu_x, \sigma_x, \lambda\}$. These results are far from satisfactory as they show no convergence to their true value. The parameters do appear to converge to a value, different to their true value, this is due to the non-linearity of the measurement equation causing multi-modal posterior distributions. The particle swarms of latent states and parameters are potentially being directed to an area of high likelihood. It is further noted that due to the multi-modality of the filtering distributions, certain runs of the algorithm could produce near perfect parameter values.

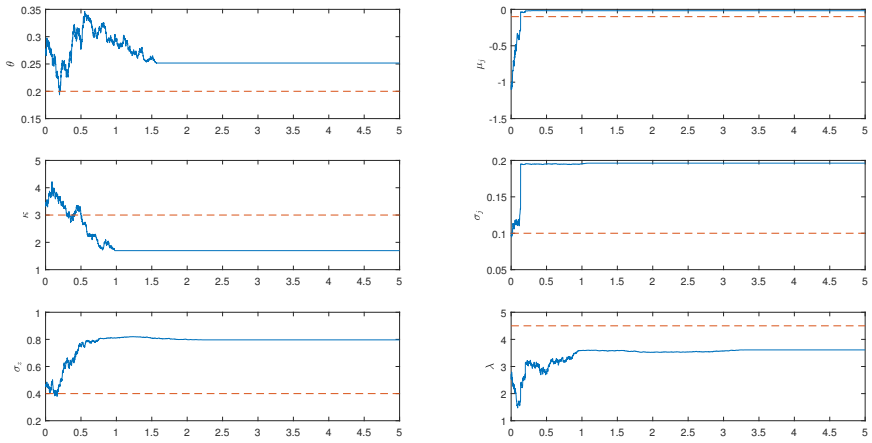


Fig. 3.4: Filtered estimates of Θ using the Uniform Initial Parameter method

It is clear from the above figures that the particle filter is finding parameter values of high likelihood, however they are incorrect. It can be concluded that for non-linear measurement equations, propagation of parameters needs to have some form of posterior knowledge of the observed and latent processes.

3.2.2 Bayesian Filtering

The Bayesian filtering framework makes up for one of the drawbacks of the uniform initial parameter distribution methods. By deriving the posterior distributions of each parameter given information about observable and latent process, this implies a more informative propagation stage of the algorithm. Prior knowledge of the parameter set is needed to determine the prior distributions. This paper makes use of conjugate prior distributions[1] with low variance so as to control the dispersion of the particle swarms in the initial stages of the algorithm. Care needs to be taken with the choice of initial prior parameters as there is a trade-off between low dispersion of particle swarms and the swarms exploring their entire sample space.

The following is an overview of the steps required before performing latent state estimation and parameter learning using the Bayesian filtering distribution method:

- Prior distributions of the parameter set, Θ are determined;
- Posterior distributions of Θ, given the observed and latent data, are derived;

[1] See Appendix A for introductory Bayesian Inference

- Identify the sufficient statistic for each parameter;
- Derive a recursive relationship for each sufficient statistic relating to the statistic from the previous time step.

These steps are performed in appendix B.

The aim is to estimate a set of parameters $\Theta = \{\mu, \theta, \kappa, \sigma_z\}$. It is assumed that the jump parameters, μ_x, σ_x, λ are fixed at their true value throughout this simulation study, furthermore, due to time constraints and to illustrate the techniques of parameter learning in a Bayesian Filtering context, the latent states Z_t and q_t are assumed to be known, the purpose being to show the degree of accuracy in the particle filtering algorithm as a parameter estimator. The figures below show the mean filtered estimate of the four parameters. The blue line attempts to track the straight, red line. It is clear that the estimates of θ and σ show fast convergence to their true value while μ has a significantly longer convergence time. The estimate of κ appears to not track its true value in the simulation study. It is well known in the literature on particle filtering (Golightly (2009), Johannes et al. (2006),Johannes et al. (2009), Carvalho et al. (2010)) that the mean reversion rate parameter in a given stochastic volatility model is particularly difficult to estimate. It is expected that with a greater number of observed processes; for example, augmenting the observed stock price with a set of associated derivative prices will increase the accuracy of the algorithm and will allow for simultaneous estimation of latent states and parameters in the Bayesian Filtering method. This extension is left as further research.

Fig. 3.5: Filtered estimates of parameter set

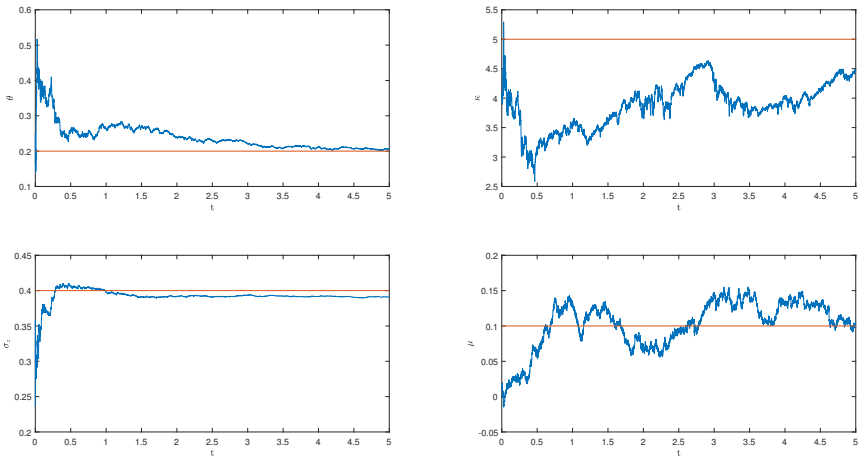
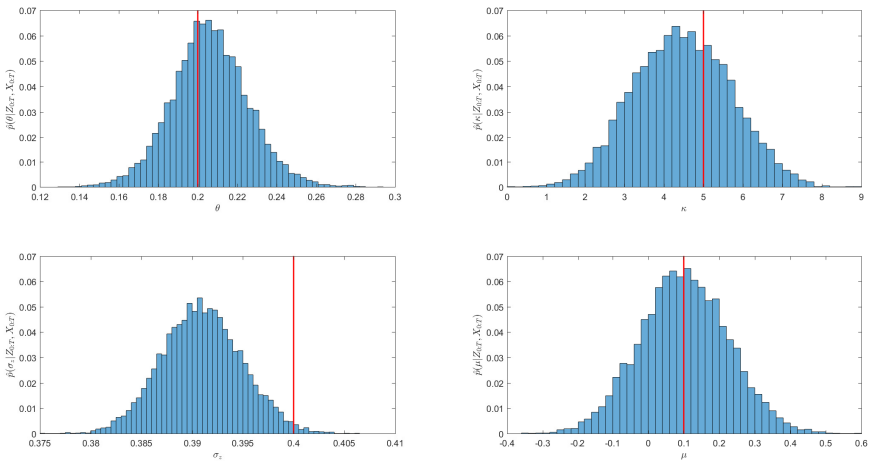


Figure 3.2.2 below show the empirical distributions of the terminal estimates of

the parameter set, with the vertical line representing the true estimate. These figures show a low variance distribution at termination of the algorithm with three of the distributions being approximately centred around the true parameter values. It is noted how low the variance is of the distribution of σ, causing a misrepresentation of the figure's portrayal of the estimate of σ. The posterior mean of σ for this study is 0.3910 which is not too significantly different to its true value 0.4. Table 3.2.2 gives the particle filter's estimate of the four parameters with comparison to their true value.

Fig. 3.6: Estimate of filtering distributions of parameter set.



Tab. 3.2: Filtered estimates of parameter set.

Parameters	μ	θ	κ	σ_z
True Value	0.1	0.2	5	0.4
Filtered Estimate	0.1006	0.2062	4.4625	0.3910

Chapter 4

Conclusion and Further Research

This dissertation has attempted to introduce particle filtering and investigate some of the numerous algorithms that have recently been developed. A detailed comparison was given of each of these algorithms with examples to illustrate their differences.

The aim of the simulation study was to separate latent state estimation and parameter learning to develop the Uniform Initial Parameter and Bayesian filtering algorithms. Although in practice the filtering problem requires simultaneous estimation to occur, the aim was to show how the two sub-problems differ from each other and how one can achieve results if one problem were solved. Further research can be undertaken to merge the two methods to achieve accurate results for simultaneous estimation of latent states and parameters.

It has been shown that not only can particle filtering identify the stochastic volatility in a jump diffusion model but can, with relative accuracy, estimate the jump times and sizes too. This opens the door for further research into using particle filtering as a method for identifying jumps in stock prices, as opposed to the classic statistical tests used to identify jumps.

Bibliography

Aihara, S. I., Bagchi, A. and Saha, S. (2008). Estimating volatility and model parameters of stochastic volatility models with jumps using particle filter, *IFAC Proceedings Volumes* **41**(2): 6490–6495.

Anderson, B. D. and Moore, J. B. (1979). Optimal Filtering, *Englewood Cliffs* **21**: 22–95.

Bakshi, G., Cao, C. and Chen, Z. (1997). Empirical performance of alternative option pricing models, *The Journal of finance* **52**(5): 2003–2049.

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

Bates, D. S. (2003). Empirical option pricing: A retrospection, *Journal of Econometrics* **116**(1-2): 387–404.

Carvalho, C. M., Johannes, M. S., Lopes, H. F., Polson, N. G. et al. (2010). Particle learning and smoothing, *Statistical Science* **25**(1): 88–106.

Chen, Z. et al. (2003). Bayesian filtering: From Kalman filters to particle filters, and beyond, *Statistics* **182**(1): 1–69.

Crisan, D. (2001). Particle filters—a theoretical perspective, *Sequential Monte Carlo methods in practice*, Springer, chapter 2, pp. 17–41.

Crisan, D. (2009). Particle filters in discrete time, *Fundamentals of Stochastic Filtering*, Vol. 3, Springer, chapter 10, pp. 257–286.

de Vos, A. F. (2004). A primer in Bayesian inference, *preprint* . Last visited 07/02/2018.
URL: *http://personal.vu.nl/a.f.de.vos/primer/primer.pdf*

Duffie, D., Pan, J. and Singleton, K. (2000). Transform analysis and option pricing for affine jump diffusions, *Econometrica* **68**: 929–952.

Geweke, J. (1989). Bayesian inference in econometric models using Monte Carlo integration, *Econometrica: Journal of the Econometric Society* pp. 1317–1339.

Golightly, A. (2009). Bayesian filtering for jump-diffusions with application to stochastic volatility, *Journal of Computational and Graphical Statistics* **18**(2): 384–400.

Gordon, N. J., Salmond, D. J. and Smith, A. F. (1993). Novel approach to nonlinear/non-gaussian Bayesian state estimation, *IEEE Proceedings F (Radar and Signal Processing)*, Vol. 140, IET, pp. 107–113.

Jacquier, E., Polson, N. and Sokolov, V. (2016). Sequential Bayesian learning for Merton's jump model with stochastic volatility, *arXiv preprint arXiv:1610.09750* .

Johannes, M., Polson, N. and Stroud, J. (2006). Exact particle filtering and parameter learning, *Technical report*, Working paper, Graduate School of Business, Columbia University.

Johannes, M. S., Polson, N. G. and Stroud, J. R. (2009). Optimal filtering of jump diffusions: Extracting latent states from asset prices, *The Review of Financial Studies* **22**(7): 2759–2799.

Julier, S. J. and Uhlmann, J. K. (1997). A new extension of the Kalman filter to nonlinear systems, *Int. Symp. Aerospace/Defense Sensing, Simul. and Controls*, Vol. 3, Orlando, FL, pp. 182–193.

Kalman, R. E. et al. (1960). A new approach to linear filtering and prediction problems, *Journal of Basic Engineering* **82**(1): 35–45.

Liu, J. S. and Chen, R. (1998). Sequential Monte Carlo methods for dynamic systems, *Journal of the American Statistical Association* **93**(443): 1032–1044.

Liu, J. and West, M. (2001). Combined parameter and state estimation in simulation-based filtering, *Sequential Monte Carlo methods in practice*, Springer, pp. 197–223.

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

Rubin, D. B. (1988). Using the SIR Algorithm to simulate posterior distributions, *Bayesian statistics* **3**: 395–402.

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.

Zaritskii, V., Svetnik, V. and Šimelevič, L. (1975). Monte Carlo technique in problems of optimal information processing, *Avtomatika i Telemekhanika* (12): 95–103.

Appendix A

Introductory Bayesian Inference

Let $X = \{x_i\}_{i=0}^{N}$ be an iid sample from pdf $f_X(x|\Theta)$ where Θ is a set or parameters defining the distribution of X. Suppose we want to infer the values of Θ after having observed the sample X.
Let $L(\Theta; X)$ be the likelihood function generated from $f_X(x|\Theta)$.
Bayes' Theorem states:

$$p(\Theta|X) = \frac{L(\Theta; X)p(\Theta)}{p(X)} \tag{A.1}$$

where $p(\Theta)$ is the prior distribution of Θ,
and $p(X)$ is the marginal distribution of X. $P(X)$ is often unknown and of little importance, therefore using the fact that $p(\Theta|X)$ is a density function, equation A.1 can be simplified to

$$p(\Theta|X) = \frac{L(\Theta; X)p(\Theta)}{\int_\Theta L(\Theta; X)p(\Theta)d\Theta} \tag{A.2}$$

Therefore by deriving $p(\Theta|X)$, the posterior mean and variance of Θ can be found. In the context of filtering, this is extremely useful as it allows for information about observed and latent process to be used in the estimation of the model parameters as well as omitting a distribution by which the model parameters can be propagated in the Monte Carlo framework.

The amount and form of prior information will inform the prior parameter distribution. A conjugate prior distribution is one whose distribution form does not change when going from prior to posterior.

Examples of these conjugate prior and posterior distributions are included in appendix B. More information on Bayesian inference can be found in de Vos (2004).

Appendix B

Bayesian Filtering Distributions

B.1 Prior Distributions

Using conjugate prior distributions for each parameter to maintain consistency of distributional form from prior to posterior. The following prior distributions are observed from the distributional form of X_t and Z_t.

$$\begin{aligned}\mu &\sim \mathcal{N}(a_0, b_0)\\ \sigma_z^2 &\sim \mathcal{IG}(c_0, d_0)\\ \kappa &\sim \mathcal{N}(e_0, f_0)\\ \theta &\sim \mathcal{N}(g_0, h_0)\end{aligned}$$

B.2 Distribution of μ

$$p(\mu) = \mathcal{N}(a_0, b_0)$$

$$p(\mu|X_{0:t}, Z_{0:t}, J_{0:t}, V_{0:t}, \Theta) \propto L_x(\Theta; X_{0:t}, Z_{0:t}, J_{1:t}, V_{1:t})p(\mu)$$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{\mu^2 - 2\mu a_0 + b_0 \sum_{i=1}^{t} \frac{((\mu\Delta t)^2 - 2\mu\Delta t y_i)}{Z_{i-1}\Delta t}}{b_0}\right)\right)$$

Where $y_i = \Delta X_i + (\lambda m^j + \frac{1}{2}Z_{i-1})\Delta t - J_i V_i$

$$\propto \exp\left(-\frac{1}{2}\left(\frac{(1 + b_0\Delta t \sum_{i=1}^{t} Z_{i-1}^{-1})\mu^2 - 2\left(a_0 + b0 \sum_{i=1}^{t} \frac{y_i}{Z_{i-1}}\right)\mu}{b_0}\right)\right)$$

$$\propto \exp\left(-\frac{1}{2}\frac{\left(\mu - \frac{\left(a_0 + b0 \sum_{i=1}^{t} \frac{y_i}{Z_{i-1}}\right)}{(1 + b_0\Delta t \sum_{i=1}^{t} Z_{i-1}^{-1})}\right)^2}{\frac{b0}{(1 + b_0\Delta t \sum_{i=1}^{t} Z_{i-1}^{-1})}}\right)$$

$$\therefore p(\mu|X_{0:t}, Z_{0:t}, J_{0:t}, V_{0:t}, \Theta) = \mathcal{N}(a_t, b_t)$$

$$a_t = \frac{\left(a_0 + b0 \sum_{i=1}^{t} \frac{y_i}{Z_{i-1}}\right)}{(1 + b_0 \Delta t \sum_{i=1}^{t} Z_{i-1}^{-1})} \qquad b_t = \frac{b0}{(1 + b_0 \Delta t \sum_{i=1}^{t} Z_{i-1}^{-1})}$$

Giving the recursive relation

$$b_t = \frac{1}{\frac{1}{b_{t-1}} + \frac{\Delta}{Z_{t-1}}}$$

$$a_t = \left(\frac{a_{t-1}}{b_{t-1}} + \frac{y_t}{Z_{t-1}}\right) b_t$$

B.3 Distribution of σ_z^2

$$p(\sigma_z^2) = \mathcal{IG}(c_0, d_0)$$

$$p(\sigma_z^2|Z_{0:t}, \Theta) \propto L_z(\Theta|Z_{0:t}) p(\sigma_z^2)$$

$$\propto (\sigma_z^2)^{-(c_0 + \frac{t}{2}) - 1} \exp\left(-\frac{1}{\sigma_z^2}\left(d_0 + \frac{1}{2}\sum_{i=1}^{t} \frac{(\Delta Z_i - \kappa(\theta - Z_{i-1})\Delta t)^2}{Z_{i-1}\Delta t}\right)\right)$$

$$\therefore p(\sigma_z^2|Z_{0:t}) = \mathcal{IG}(c_t, d_t)$$

$$c_t = c_0 + \frac{t}{2} \qquad d_t = d_0 + \frac{1}{2}\sum_{i=1}^{t} \frac{(\Delta Z_i - \kappa(\theta - Z_{i-1})\Delta t)^2}{Z_{i-1}\Delta t}$$

Giving the recursive relation

$$c_t = c_{t-1} + \frac{1}{2}$$

$$d_t = d_{t-1} + \frac{1}{2}\frac{(\Delta Z_t - \kappa(\theta - Z_{t-1})\Delta t)^2}{Z_{t-1}\Delta t}$$

B.4 Distribution of κ

Similarly,

$$p(\kappa|Z_{0:t}) = \mathcal{N}(e_t, f_t)$$

$$e_t = \frac{e_0 + f_0 \sum_{i=1}^{t} \frac{\Delta Z_i (\theta - Z_{i-1})}{\sigma_z^2 Z_{t-1}}}{1 + f_0 \sum_{i=1}^{t} \frac{(\theta - Z_{i-1})^2 \Delta t}{\sigma_z^2 Z_{i-1}}} \qquad f_t = \frac{f_0}{1 + f_0 \sum_{i=1}^{t} \frac{(\theta - Z_{i-1})^2 \Delta t}{\sigma_z^2 Z_{i-1}}}$$

Giving the recursive relation

$$f_t = \frac{1}{\frac{1}{f_{t-1}} + \frac{(\theta - Z_{t-1})^2 \Delta t}{\sigma_z^2 Z_{t-1}}}$$

$$e_t = \left(\frac{e_{t-1}}{f_{t-1}} + \frac{\Delta Z_t (\theta - Z_{t-1})}{\sigma_z^2 Z_{t-1}}\right) f_t$$

B.5 Distribution of θ

Similarly,

$$p(\theta|Z_{0:t}, \Theta) = \mathcal{N}(g_t, h_t)$$

$$g_t = \frac{g_0 + h_0 \sum_{i=1}^{t} \frac{\Delta Z_i \kappa + \kappa^2 Z_{i-1} \Delta t}{\sigma_z^2 Z_{i-1}}}{1 + h_0 \frac{\kappa^2 \Delta t}{\sigma_z^2} \sum_{i=1}^{t} Z_{i-1}^{-1}} \qquad h_t = \frac{h_0}{1 + h_0 \frac{\kappa^2 \Delta t}{\sigma_z^2} \sum_{i=1}^{t} Z_{i-1}^{-1}}$$

Giving the recursive relation

$$h_t = \frac{1}{\frac{1}{h_{t-1}} + \frac{\kappa^2 \Delta t}{\sigma_z^2 Z_{t-1}}}$$

$$g_t = \left(\frac{g_{t-1}}{h_{t-1}} + \frac{\Delta Z_t \kappa + \kappa^2 Z_{t-1} \Delta t}{\sigma_z^2 Z_{t-1}} \right) h_t$$

B.6 Sufficient Statistic and Parameter Propagation

The array of sufficient statistics s_t in section B is defined as

$$s_t = \{a_t, b_t, c_t, d_t, e_t, f_t, g_t, h_t\}$$

with the function $\mathcal{S}$ defined by the recursive relationships stated above.

The parameter particles are therefore propagated according to their posterior distributions, $p(\Theta|s_t)$.