Comparison of Methods for Estimating Stochastic Volatility

A Thesis Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

John Parnell Collins

August 2013

Acknowledgements

I would like to thank my advisor Dr. Vasileios Maroulas for all his help during this journey of completing my Master's in mathematics. In his class, I found my interest in mathematical statistics and probability. He showed a world of applications that led to many enjoyable projects and discussions. Through his guidance, I was able to work on a real world problem for my thesis that has been both challenging and rewarding. He put aside his time during the summer and at a very busy point in his life to always suggest new approaches and methods.

Furthermore, I would like to thank my committee members Dr. Jan Rosinski and Dr. Fernando Schwartz for their thoughtful comments.

Finally, I want to thank my friends and family. Thank you Steve Fassino and Daniel Bauer for allowing intellectual discussion to be an acceptable way of passing the time. Thank you for sharing in my joy when methods ran smoothly and my agony when my computer turned red in coding errors. Thank you to my parents Chuck and Cindy Collins for all the support you gave in good old-fashioned tough love, which proved to be the encouragement I needed to press on until the end. However, when a more gentle approach was needed, I could always turn to Kelly. Thank you Kelly Cox for all your patience and understanding this summer.

Abstract

Understanding the ever changing stock market has long been of interest to both academic and financial institutions. The early attempts to model the dynamics treated the volatility or sensitivity of the price change to random effects as constant. However, in matching the model to real data it was realized that the volatility was actually a random variable, and thus began efforts to determine methods for estimating the stochastic volatility from experimental data.

In this thesis, we develop and compare three different computational statistical filtering methods for estimating the volatility: The Kalman Filter, the Gibbs Sampler, and the Particle Filter. These methods are applied to a discrete time version of the log-volatility dynamic model and the results are compared based on their performance on synthetic data sets, where dynamics are nonlinear.

All the methods struggled to provide accurate estimates, but in comparison, the Gibbs Sampler provided the most accurate estimates, with Particle Filtering providing the least accurate results. Therefore, further investigation on the topic should take place.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Understanding the ever changing stock market has long been of interest to both academic and financial institutions. In 1972, the discover of the Black-Shcoles equations was a groundbreaking results in the field of financial mathematics [4]. The equation became the off-the-shelf methodology for the pricing of options in the market place. Through this approach, one was able to analyze the dynamic of the price, S, of a stock by the model

$$\frac{dS}{S} = rdt + \sigma dW, \tag{1.1}$$

where r is the constant intrinsic growth rate, W is the driving noise, and σ is the constant volatility or the sensitivity of the price change to the noise. Equivalently, if we consider the price to be measured at discrete times, we replace S by S_t for $t = 0, 1, \ldots$ and rewrite the Black-Scholes equation (1.1) as

$$S_{t+1} = S_t + rS_t + \sigma S_t \Delta W_t, \tag{1.2}$$

where r and σ are as in equation (1.1) (adjusted to the time scale), and $\Delta W_t = W_{t+1} - W_t$ is the noise increment.

Initially, stock prices were modeled under the assumption that the volatility was constant. In 1985, Rubinstein [30] addressed these concerns about this assumption by showing that the volatility should not be constant, but should be considered a randomly changing quantity. In attempting to draw inference from the stock price about how volatility should behave, equation (1.2) was adjusted to only examine the relative change in stock price, y_t, with the volatility, σ_t now being a random variable, depending on time:

$$y_t \doteq \frac{S_t - S_{t-1}}{S_{t-1}} = \sigma_{t-1}\Delta W_{t-1}. \tag{1.3}$$

Determining methods for estimating stochastic volatility from real market data became an active and important area of investigation.

There are basically two general types of methods for estimating the volatility: regression based methods and filtering methods. The early works were regression based with Engle publishing the first method in 1982 [10]. Engle assumed there was data at time $t-1$ that gave investors information about how a stock price might change. He then looked at the expected relative price of the stock and the variance of the stock conditional on this information. In particular, he defined $m_t \doteq \mathbb{E}[y_t \mid F_{t-1}]$, the expected price conditional on the information F_{t-1}, and $h_t \doteq \text{var}(y_t \mid F_{t-1})$, the variance conditional on F_{t-1}. In this case, h_t is a measure of σ_t^2. From the errors between actual and expected prices, $e_t \doteq y_t - m_t$, he applied an autoregressive conditional heteroskedasticity (ARCH) model of degree p to determine the coefficients a, b_i, $i = 1, \ldots, p$, for the volatility model

$$h_t = a + \sum_{i=1}^{p} b_i e_{t-i}^2. \tag{1.4}$$

The method was further extended in 1986 from the ARCH(p) model to a Generalized ARCH model (GARCH(p, q)) [5], where p represents how far back we use the information history and q represents how far back we use the variance history. In

this model, we have

$$h_t = a + \sum_{i=1}^{p} b_i e_{t-i}^2 + \sum_{i=1}^{q} c_i h_{t-i}. \tag{1.5}$$

Building on these models, many other approaches were made popular such as the Exponential GARCH (EGARCH), the nonlinear ARCH, the multiplicative ARCH, and several others outlined in [11]. Although these statistical methods are successful in predicting volatility, the concept of the new information needed for the predictions is mathematically difficult to represent in the general setting.

Soon after the regression models were developed, researchers started using a different set of methods based on stochastic filtering. In general, in filtering problems we estimate an unknown random variable by using observations perturbed by random noise, see e.g. [3, 25, 37]. These types of problems arise in many areas such as tracking an object's location, weather forecasting, monitoring ocean currents, biology, the military, predicting the volatility of stock prices, and many others [7, 8, 22, 24, 23, 28, 34]. Briefly, the generic filter problem consists of two stochastic differential equations, a transition model for the hidden state variable and an observation model to form intuition about the hidden variable. Typically, the transition density is a Markov process where the stochastic equation depends only on the previous state variable. The observation model depends only on the state variable, along with some random noise, typically white noise. For our problem, we observe the changing stock price in order to estimate the hidden state variable, volatility. We discuss several families of filtering methods below.

The Kalman Filter method is one such approach to produce this more precise estimate. Developed in 1960, this method calculates the optimal estimate by minimizing the variance of error between the estimate and the true value [17]. It is optimal when applied to systems with linear dynamics and data driven by Gaussian noise. The method involves producing a prior estimate, using the dynamic equation, before updating to a posterior estimate by making adjustments according to what we observe. A key component of the method is the Kalman Gain which determines the

degree to which the posterior estimate should be corrected to reflect the deviation in what is observed and what our prior says we should observe. The Kalman Gain is found by minimizing the variance of the posterior error. Although the Kalman Filter method is the optimal method under the right properties, for many problems, including ours, the state or observation dynamics are not linear and thus, to apply this method, we first would need to linearize the dynamics. We expect that this approximation would introduce additional error into the method, thus making it less accurate.

Another strategy for determining volatility estimates is via Monte Carlo Markov Chain (MCMC) methods. MCMC methods were formalized in 1949 [27], right after World War II. At that time, in the field of physics, there existed a strong interest in the movement of particles [29]. The problem of interest was to compute the configuration of particles after some time had passed. Due to all the possible collisions that can take place, the analytic approach was not feasible. With recent developments in computers, the solution was found by having the computer randomly place particles and then simulate their movement through given rules. After many iterations, the average configuration of particles approached the analytic solution with probability 1. This method of repeatedly simulating the unknown to estimate an expectation became known as Monte Carlo method.

For the general filtering problem, we want to use observations y_t in order to estimate the state variables x_t. Mathematically speaking, we want to estimate a function of the state of the system, say $f(x_t)$, using the sequence of noisy data. In other words, the filtering problem attempts to approximate the conditional expectation $\mathbb{E}[f(X_t) \mid Y_t]$ or equivalently to compute the conditional probability $p(x_t \mid y_t)$. If this relation was easy to sample from, then by the standard Monte Carlo sampling methods, we could simulate $\hat{x}_t^i$, $i = 1, \ldots, N$ from the conditional distribution to produce an estimate of the true value. However, in practice, the filter density is hard to sample from. This is especially true for higher dimensional problems. In 1953, the Metropolis algorithm, an MCMC method, overcame this issue

by proposing a new symmetric distribution $Q(x_t \mid y_t, x_{t-1})$ that was easy to sample from [26]. Using this distribution, we update our estimate by either accepting the sample from the proposed distribution or using an estimate from a previous time step. The acceptance rate depends on the likelihood of the new sample in comparison to the old estimate using the original density p.

Due to the restricting assumptions for choosing the proposal distribution, the algorithm was further generalized in 1970 into the Metropolis-Hastings algorithm [15]. The more generalized approach was more relaxed in selecting the proposal distribution, which only required the proposal distribution to be symmetric, a full conditional or something else entirely.

A special case of Metropolis-Hastings algorithm is Gibbs Sampling. The proposal in the Gibbs Sampler is chosen to be $Q(x_t \mid x_{-t})$, where $x_{-t} = (x_1, x_2, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T)$, i.e. all the time series values except for x_t [13]. Formalized in 1984, the method uses the full conditional distribution to use all past and previous estimates in time instead of only the value from the previous step. Unlike the Metropolis-Hasting algorithm, the advantage of this proposal is that all the samples are accepted, i.e. the acceptance rate is always one. Basically, using the Gibbs Sampler, we sample from the full conditional distribution of each parameter. It is an iterative algorithm which constructs a dependent sequence of parameter values whose distribution converges to the target joint posterior distribution.

The third method of interest is the Particle Filtering method. It is a special case of the importance sampling algorithm [21]. Importance sampling originated around the time of the Metropolis-Hastings algorithm as another type of MCMC method [12]. Similarly, it deals with the difficulty of sampling from the posterior conditional distribution $p(x_t \mid y_t)$ by proposing a new distribution $Q(x_t \mid y_t, x_{t-1})$ that is easy to sample from. Where the Metropolis algorithm only kept some samples depending on the likelihood of the estimate, importance sampling keeps all samples but weights them according to their likelihood, and uses these weights to approximate the expected value.

A popular method of computational implementation of filtering is the Particle Filter [18, 24]. Particle Filters are capable of handling nonlinear and non-Gaussian scenarios. This method is an importance sampling method, which approximates the posterior distribution $p(X_t \mid Y_t)$ by a discrete set of weights. The weighs of the samples are computed using the observations. Briefly, instead of using $p(X_t \mid Y_t)$, one may employ an alternative density, say $q(X_t \mid Y_t)$, which can be easily sampled.

For this thesis, we focus on using versions of the Kalman Filter, Gibbs Sampling and Particle Filtering methods to develop and compare computational statistical algorithms for determining stochastic volatility. We apply these methods to a specific model related to stock price volatility. Through the work of [16, 31, 36], volatility can be expressed as a solution of the following stochastic differential equation:

$$\frac{d\sigma^2}{\sigma^2} = \phi dt + \eta dB, \tag{1.6}$$

where the parameter ϕ is the intrinsic growth and η is the sensitivity to the driving noise which is a Brownian motion, B, in our case. In particular, we will use the discrete version of equation (1.6):

$$x_t = \nu + \phi x_{t-1} + \eta w_t, \tag{1.7}$$

where $x_t = \log \sigma_t^2$, the log variance, ν is a scaling factor to ensure a non-zero fixed point, and $w_t \sim N(0,1)$ by a property of Brownian motion. With this log-transform, we also rewrite (1.3) with σ now depending on time and $\sigma_t = e^{x_t/2}$, as

$$y_t \doteq \sigma_t v_t = e^{x_t/2} v_t, \tag{1.8}$$

where we have $v_t \sim N(0,1)$ for the difference in the noise. For our work, we will use the dynamic and observation models defined by equations (1.7) and (1.8) respectively.

We will apply the three filtering algorithms we develop and compare estimates of the corresponding stochastic volatility for this model. The basis for the comparisons

will be the performance of the algorithms on synthetic data. The difficulty of this problem is the non-linearity, e.g. in equation (1.8). Drawing inference on the variance of a random variable only given one observation at each time step is a difficult task. Our methods, which all build on this relationship, struggled to overcome this obstacle when looking at the numerical results of each method. In comparison, after applying each method to the same randomly generated data set, the Gibbs Sampler produced the most accurate estimates over many simulations.

The rest of this thesis is organized as follows: Chapter 2 discusses preliminary concepts for non-experts. Chapter 3 gives a brief overview of the Kalman Filter, Gibbs Sampler and Particle Filtering methods. Lastly, Chapter 4 focuses on the different computations of the stochastic volatility for the stock model with a comparison and discussion of the errors.

Chapter 2

General Preliminaries

We expose preliminary results for the uninitiated reader. Experts could skip to Chapter 3.

2.1 Sampling Random Variables

Let X be a random variable distributed according to F, and we will denote this from now on as $X \sim F$. Define

$$F^{-1}(y) = \inf \{x : F(x) \geq y\}, 0 \leq y \leq 1, \tag{2.1}$$

where F^{-1} is well-defined since F is a non-decreasing function.

We want to produce samples from this generic distribution F.

Lemma 2.1. *Let U follow a uniform distribution over the interval $[0, 1]$, i.e. $U \sim U(0, 1)$. Consider $X = F^{-1}(U)$, where F^{-1} is defined in equation (2.1). Then X is F-distributed.*

Proof. Following the basic definition of a cumulative distribution function (cdf) we have that

$$\mathbb{P}(X \leq x) = \mathbb{P}(F^{-1}(U) \leq x) = \mathbb{P}(U \leq F(x)) = F(x), \tag{2.2}$$

since $U \sim U(0,1)$. $\square$

Therefore, we use the following algorithm to generate samples from a specific distribution F.

Algorithm 2.2 (Inverse Transform Algorithm)**.**

1. *Sample* $X \sim U(0,1)$
2. *Compute* $Y = F^{-1}(x)$

Definition 2.3. *Consider* $X^1, \ldots, X^N$ *samples of a distribution* F*. Then the empirical distribution of* X^i*,* $i = 1, \ldots, N$ *is*

$$\hat{F}_N(t) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}_{\{X^i \leq t\}}, \tag{2.3}$$

where $\mathbb{1}$ *is the indicator function.*

Example 2.4. *Suppose we want to sample from a distribution* F *with density* $f(y) = 3y^2$*, if* $0 \leq y \leq 1$ *(*$f(y) = 0$ *otherwise).* $F(y) = \mathbb{P}(Y \leq y) = \int_0^y 3y^2 dy = y^3$*. Following the inverse transform algorithm we generate* $N = 1000$ *samples from* $U(0,1)$*, i.e.* $X^1, \ldots, X^{1000} \sim U(0,1)$*, and then we evaluate the inverse based on these samples, i.e.* $y^i = \sqrt[3]{x^i}$*,* $i = 1, \ldots, N$*. The empirical distribution of the samples,* y^i*, are compared with the true cdf* $F(y) = y^3$ *and are shown in Figure (2.1).*

Although Lemma 2.1 suggests to us an explicit way of generating a random variable when the cdf is available, there are several cases in which the corresponding distributions are mathematically inconvenient. The acceptance-rejection method is a very general algorithm for sampling [33].

Consider a probability density function (pdf) f bounded on some interval $[a,b]$ and zero outside. Let $c = \sup\{f(x) : x \in [a,b]\}$. Then f takes values within the rectangle $[a,b] \times [0,c]$. We generate $Z \sim f$ with the following algorithm:

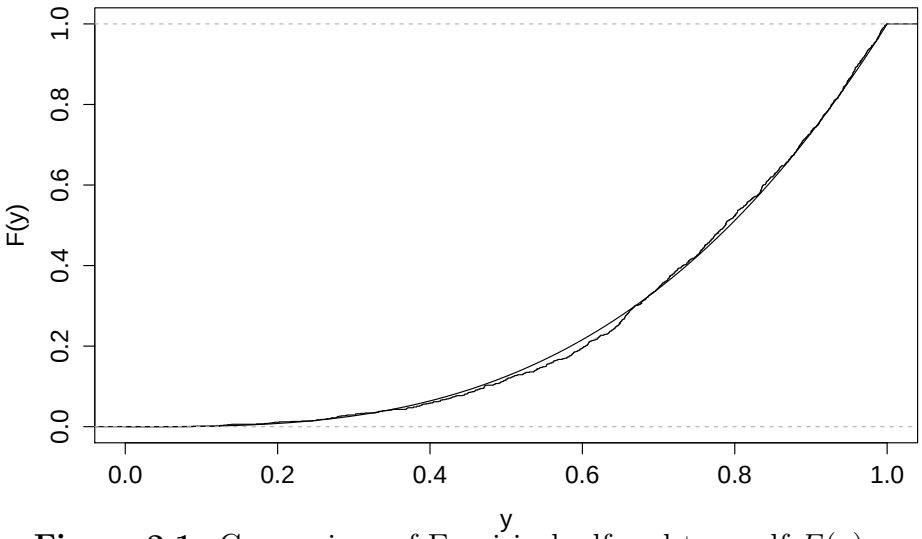

Figure 2.1: Comparison of Empirical cdf and true cdf $F(y) = y^3$.

Algorithm 2.5.

1. *Sample* $U \sim U(a, b)$
2. *Sample* $V \sim U(0, c)$
3. *If* $Z = U$ *if* $V \leq f(U)$*, otherwise go to Step 1.*

The generated vector $(U, V) \sim U([a, b] \times [0, c])$, and therefore, the accepted pair (U, V) is uniformly distributed under the pdf f. This implies that the distribution of the accepted values has the desired pdf f.

We further generalize the algorithm by letting g be any density where $Cg(x) \geq f(x)$ for some constant C. We call $g(x)$ the proposal pdf and assume that is it easy to sample from.

Algorithm 2.6 (Acceptance-Rejection Algorithm)**.**

1. *Generate* $X \sim g(x)$
2. *Generate* $Y \sim U(0, Cg(X))$

3. $Z = X$, if $Y \leq f(X)$, otherwise go to Step 1.

Theorem 2.7. *The random variable generated according to the acceptance-rejection algorithm has the desired pdf $f(x)$.*

Proof. Define

$$\begin{aligned} A &= \{(x, y) : 0 \leq y \leq Cg(x)\} \\ B &= \{(x, y) : 0 \leq y \leq f(x)\}. \end{aligned}$$

Following the acceptance-rejection algorithm, we sample $X \sim g(x)$ and $Y \sim U(0, Cg(X))$. The resulting vector (X, Y) is uniformly distributed on A. Let (X^*, Y^*) be the first accepted sample, i.e. $(X^*, Y^*) \in B$. Thus, we take $Z = X^*$, which will have the pdf:

$$f_Z(z) = \int_0^{f(x)} f_{Z,Y^*}(z, y^*)dy^* = \int_0^{f(x)} 1dy^* = f(x). \tag{2.4}$$

□

The efficiency of the acceptance-rejection algorithm is defined as

$$\mathbb{P}((X, Y) \texttt{ is accepted}) = \frac{\texttt{area of } B}{\texttt{area of } A} = \frac{1}{C}. \tag{2.5}$$

Thus, in choosing our proposal density $g(x)$, we want high efficiency $\frac{1}{c}$, i.e. $c \approx 1$. We achieve this by $g(x) \approx f(x)$.

2.2 Bayes Theorem

This section will briefly introduce the Bayes rule. Bayes rule and its philosophy have a great impact in engineering, science, statistics, and mathematics. Bayes rule, briefly, does not inform what our belief should be, but rather "the essence of the Bayesian approach is to provide a mathematical rule explaining how you should change your

existing beliefs in the light of new evidence, it allows scientists to combine new data with their existing knowledge" as stated in [1].

Consider a parameter of interest θ which is unknown and one wants to identify values of $\theta \in \Theta$ based on data X. According to Bayesian methods, one needs a distribution of θ, called the prior distribution, $\pi(\theta)$, and a sampling model, $f(x \mid \theta)$. The prior distribution, $\pi(\theta)$, quantifies the uncertainty about θ prior to seeing data or describes the belief that x would be the outcome of our study if we knew θ to be true. Once we obtain the data X, we update our beliefs about θ with the posterior distribution, $\pi(\theta \mid x)$, using the Bayes formula described below,

$$\pi(\theta \mid x) = \frac{\pi(\theta) f(x \mid \theta)}{\int_\Theta \pi(\theta) f(x \mid \theta) d\theta} \tag{2.6}$$

The posterior distribution, $\pi(\theta \mid x)$, describes our belief that θ is the true value, having observed the data set x. It is crucial to note that the Bayes rule does not inform what our belief should be, but it informs how our belief should change after seeing new evidence.

Definition 2.8. *For* $\mathbf{R}$*-valued* θ *the posterior mean and variance are given by:*

$$\mathbb{E}(\theta \mid x) = \int_{-\infty}^{\infty} \theta \pi(\theta \mid x) d\theta \tag{2.7}$$

$$Var(\theta \mid x) = \int_{-\infty}^{\infty} (\theta - \mathbb{E}(\theta \mid x))^2 \pi(\theta \mid x) d\theta \tag{2.8}$$

The following clarifies how one may use a prior and a sampling model in order to propagate a posterior distribution using the equation (2.6).

Example 2.9. *Let us consider the independent identically distributed (iid) random variables* $X = (X^1, \ldots, X^N)$ *normally distributed,* $N(\mu, \sigma^2)$*, where* σ^2 *is known. Assuming a normal prior distribution for the mean,* μ*, i.e.* $N(\eta, \tau^2)$*, we will show that the posterior distribution* $\mu \mid X$ *is also normal.*

$$\textit{Prior distribution: } \pi(\mu) = \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{(\mu-\eta)^2}{2\tau^2}}$$

$$\textit{Sampling model: } f(X \mid \mu) = (2\pi\sigma^2)^{-n/2} e^{-\frac{\sum(X^i-\mu)^2}{2\sigma^2}}$$

$$\begin{aligned}
\pi(\mu \mid X) &\propto \frac{1}{\sqrt{2\pi\tau^2}} e^{-\frac{(\mu-\eta)^2}{2\tau^2}} (2\pi\sigma^2)^{-n/2} e^{-\frac{\sum(X^i-\mu)^2}{2\sigma^2}} \\
&\propto e^{-\frac{(\mu-\eta)^2}{2\tau^2} - \frac{\sum(X^i-\mu)^2}{2\sigma^2}} \\
&\propto e^{-\frac{1}{2}\left[\left(\frac{1}{\tau^2}+\frac{N}{\sigma^2}\right)\mu^2 - 2\left(\frac{\eta}{\tau^2}+\frac{\sum X^i}{\sigma^2}\right)\mu + \left(\frac{\eta^2}{\tau^2}+\frac{\sum(X^i)^2}{\sigma^2}\right)\right]} \\
&\propto e^{-\frac{\left[\mu - \left(\frac{\eta}{\tau^2}+\frac{\sum X^i}{\sigma^2}\right)/\left(\frac{1}{\tau^2}+\frac{N}{\sigma^2}\right)\right]^2}{2/\left(\frac{1}{\tau^2}+\frac{N}{\sigma^2}\right)}} \\
&= \frac{1}{\sqrt{2\pi\left(1/\sqrt{\frac{1}{\tau^2}+\frac{N}{\sigma^2}}\right)^2}} e^{-\frac{(\mu - \left(\frac{\eta}{\tau^2}+\frac{\sum X^i}{\sigma^2}\right)/\left(\frac{1}{\tau^2}+\frac{N}{\sigma^2}\right))^2}{2(1/\sqrt{\frac{1}{\tau^2}+\frac{N}{\sigma^2}})^2}}
\end{aligned}$$

Hence, $\mu \mid X \sim N\left(\left(\frac{\eta}{\tau^2}+\frac{\sum X^i}{\sigma^2}\right)/\left(\frac{1}{\tau^2}+\frac{N}{\sigma^2}\right), 1/\sqrt{\frac{1}{\tau^2}+\frac{N}{\sigma^2}}\right)$.

2.3 Markov Chains

This section discusses a few preliminary definitions with respect to Markov Chain. This is important for introducing the Markov Chain Monte Carlo techniques. We say that X_t is a discrete Markov Chain with transition matrix $p(i,j)$, if for any $j, i, i_{t-1}, \ldots, i_0$, we have the following property.

$$\mathbb{P}(X_{t+1} = j \mid X_t = i, X_{t-1} = i_{t-1}, \ldots, X_0 = i_0) = \mathbb{P}(X_{t+1} = j \mid X_t = i) \tag{2.9}$$

Equation (2.9) is called the Markov property. Basically, it admits that the future behavior of the system depends only on the present and not on its past history. Furthermore, the transition probability gives the rules of the model.

Definition 2.10. *The one-step transition probability, denoted as $p_{ij}(t)$, is defined as the following conditional probability:*

$$\mathbb{P}(X_{t+1} = j \mid X_t = i) \tag{2.10}$$

in other words the probability that the process is in state j at time $t+1$ given that the process was in state i at the previous time t.

Definition 2.11. *If the transition probabilities $p_{ij}(t)$ in a Markov chain do not depend on time t, they are said to be stationary or time-homogeneous.*

For more information on Markov chains and their properties one may refer to [2, 9].

Chapter 3

Sampling and Filtering Methods

We often want to summarize several aspects of a posterior distribution. For instance, we may be interested in the moments of some function of a parameter θ. However, it might not be easy to approximate the full posterior distribution, especially when we deal with multi-parameter models. Hence, one may address this problem by sampling from the full conditional distribution of each parameter by employing a posterior approximation based on the Gibbs sampler. We also investigate the special linear and Gaussian cases using the optimal technique, the Kalman Filter.

3.1 Dynamic Models

Let $\{x_t \in \mathcal{X} : t \in \mathbf{N}\}$ be the hidden state vectors of the system, let $\{y_t \in \mathcal{Y} : t \in \mathbf{N}\}$ be the observable variables, and let $\theta \in \Theta$ be the parameter vector for the model. Let us further assume that the state space satisfies $\mathcal{X} \subset \mathbf{R}^{n_x}$, the observation space satisfies $\mathcal{Y} \subset \mathbf{R}^{n_y}$, and the parameter space satisfies $\Theta \subset \mathbf{R}^{n_\theta}$. For convenience, we denote the collection of state vectors up to time t as $x_{0:t}$, i.e. $x_{0:t} = (x_0, x_1, \ldots, x_t)$ and the collection of observable vectors as $y_{1:t}$, i.e. $y_{1:t} = (y_1, y_2, \ldots, y_t)$. The general

dynamic model [2, 9] is then given as

$$\text{Initial distribution: } x_0 \sim p(x_0 \mid \theta) \tag{3.1}$$

$$\text{Transition density: } x_t \sim p(x_t \mid x_{0:t-1}, y_{1:t-1}, \theta) \tag{3.2}$$

$$\text{Measurement density: } y_t \sim p(y_t \mid x_t, y_{1:t-1}, \theta), \tag{3.3}$$

where $p(x_0 \mid \theta)$ can be interpreted as the prior distribution on the initial state of the system. From an analytic point of view, to obtain simpler relations, we need to introduce simplifying hypotheses on the dynamics of the model and on the measurement density. Assume that the dynamic model is Markovian and that it does not depend on the past observations $y_{1:t-1}$, then equations (3.1), (3.2) and (3.3) become

$$\text{Initial distribution: } x_0 \sim p(x_0 \mid \theta) \tag{3.4}$$

$$\text{Transition density: } x_t \sim p(x_t \mid x_{t-1}, \theta) \tag{3.5}$$

$$\text{Measurement density: } y_t \sim p(y_t \mid x_t, \theta). \tag{3.6}$$

We are interested in estimating the hidden state vector when the parameter vector is known, in other words, we want to estimate the density $p(x_t \mid y_{1:s}, \theta)$. If $t = s$, the density is called the filtering density, if $t < s$, it is called the smoothing density, and if $t > s$, it is called the prediction density. Assume that at the recurrent time t the density $p(x_{t-1} \mid y_{t-1}, \theta)$ is known. For $t = 1$ we have $p(x_0 \mid y_0, \theta) = p(x_0 \mid \theta)$, the initial distribution. Applying the Chapman-Kolmogorov transition density, we obtain the one step ahead prediction density:

$$p(x_t \mid y_{t-1}, \theta) = \int_{\mathcal{X}} p(x_t \mid x_{t-1}, \theta) p(x_{t-1} \mid y_{t-1}, \theta)\, dx_{t-1}. \tag{3.7}$$

Next, using the new observation y_t and Bayes formula (2.6), we can update the prediction density and filter the current state of the system to get the filtering density

$$p(x_t \mid y_t, \theta) = \frac{p(y_t \mid x_t, \theta)p(x_t \mid y_{t-1}, \theta)}{\int_{\mathcal{X}} p(y_t \mid x_t, \theta)p(x_t \mid y_{t-1}, \theta)\, dx_t}. \tag{3.8}$$

At each time step t, it is possible to determine the K-steps-ahead prediction density, conditional on the available information $y_{1:t}$. Given the dynamic model described by equations (3.4)-(3.6), the prediction density at the first step is

$$p(x_{t+1} \mid y_t, \theta) = \int_{\mathcal{X}} p(x_{t+1} \mid x_t, \theta)p(x_t \mid y_t, \theta)\, dx_t. \tag{3.9}$$

We can then establish the prediction density at the k-th step, $k = 1, \ldots, K$ by

$$p(x_{t+k} \mid y_t, \theta) = \int_{\mathcal{X}} p(x_{t+k} \mid x_{t+k-1}, \theta)p(x_{t+k-1} \mid y_t, \theta)\, dx_{t+k-1}, \tag{3.10}$$

where

$$\begin{aligned} p(x_{t+k} \mid x_{t+k-1}, \theta) &= \int_{\mathcal{Y}^{k-1}} p(x_{t+k}, y_{t+1:t+k-1} \mid x_{t+k-1}, \theta)\, dy_{t+1:t+k-1} \\ &= \int_{\mathcal{Y}^{k-1}} p(x_{t+k} \mid x_{t+k-1}, y_{t+1:t+k-1}, \theta)p(y_{t+1:t+k-1} \mid y_t)\, dy_{t+1:t+k-1}. \end{aligned}$$

Similarly, the K-steps-ahead prediction density of the observable variable y_{t+k} conditional on the information available at time t is given by

$$p(y_{t+k} \mid y_t, \theta) = \int_{\mathcal{X}} p(y_{t+k} \mid x_{t+k}, \theta)p(x_{t+k} \mid y_t, \theta)\, dx_{t+k}. \tag{3.11}$$

3.2 Monte Carlo Methods

There are several ways to compute the integral in equation (3.7) for the posterior prediction density. The feasibility of these integration methods depends heavily on the particular details of the dynamic model, prior distribution, etc. An alternate

technique is to use the Monte Carlo approximation which does not require a deep knowledge of calculus nor numerical analysis [8, 14].

For the general Monte Carlo approximation, let θ be the parameter of interest and let $y_1, \ldots, y_n$ be numerical samples from a distribution $p(y_1, \ldots, y_n \mid \theta)$. Suppose we sample S independent, random θ-values from the posterior distribution $p(\theta \mid y_1, \ldots, y_n)$, i.e.

$$\theta^1, \ldots, \theta^S \overset{\text{i.i.d.}}{\sim} p(\theta \mid y_1, \ldots, y_n).$$

Then the empirical distribution of $\theta^1, \ldots, \theta^S$ is known as the Monte Carlo approximation of the posterior distribution $p(\theta \mid y_1, \ldots, y_n)$.

In our situation defined by equations (3.4)-(3.6), usually we are able to sample from the prior $p(x_{t-1} \mid y_{t-1})$ and the transition density $p(x_t \mid x_{t-1})$, but it is too complicated to sample from the posterior predictive distribution $p(x_t \mid y_{t-1})$. We can then create these samples indirectly using a Monte Carlo procedure. From equation (3.7) we see that $p(x_t \mid y_{t-1})$ is the expectation of $p(x_t \mid x_{t-1})$, therefore we can approximate the predictive distribution by this two-stage Monte Carlo scheme:

$$\begin{aligned} &\text{Sample } x_{t-1}^i \sim p(x_{t-1} \mid y_{t-1}), i = 1, \ldots, S \\ &\text{Sample } x_t^i \sim p(x_t \mid x_{t-1}^i), i = 1, \ldots S, \end{aligned}$$

where the sequence $\{x_t^i\}_{i=1}^S$ constitutes S independent samples from the posterior predictive distribution.

3.3 The Gibbs Sampler

For many models, the posterior distribution is non-standard, and therefore, it is difficult to sample from it directly. However, it may be easy to consider the full conditional distribution of each parameter and then to construct a posterior approximation using the Gibbs Sampler [13, 14]. The Gibbs Sampler is an iterative algorithm which constructs a Markov chain which converges to the distribution of

interest. Broadly speaking, this method generates the states, one at a time, using the Markov property of the dynamic model to condition on the current value of the neighboring states.

Again, start with a Markovian dynamic model as in Section 3.1:

$$\text{Prior distribution for } \theta\text{: } \theta \sim p(\theta) \tag{3.12}$$

$$\text{Initial distribution: } x_0 \sim p(x_0 \mid \theta) \tag{3.13}$$

$$\text{Transition density: } x_t \sim p(x_t \mid x_{t-1}, \theta) \tag{3.14}$$

$$\text{Measurement density: } y_t \sim p(y_t \mid x_t, \theta). \tag{3.15}$$

On the time interval $\{1, \ldots, T\}$, the conditional posterior distributions of the parameter vector and the state vectors are given by

$$p(\theta \mid x_{0:T}, y_{1:T}) \quad \propto \quad p(\theta)p(x_0 \mid \theta) \prod_{t=1}^{T} p(y_t \mid x_t, \theta)p(x_t \mid x_{t-1}, \theta) \tag{3.16}$$

$$p(x_{0:T} \mid y_{1:T}, \theta) \quad \propto \quad p(x_0 \mid \theta) \prod_{t=1}^{T} p(y_t \mid x_t, \theta)p(x_t \mid x_{t-1}, \theta). \tag{3.17}$$

From the conditional distributions, we construct the Gibbs Sampler. In each step we simulate the parameter θ from the distribution in (3.16) and then simulate from the distribution in (3.17) using the value of θ from the previous step. When the conditional distributions cannot be directly simulated, the corresponding steps in the Gibbs Sampler can be replaced by the Metropolis-Hastings algorithm, which is a generalization of the Gibbs and the Metropolis algorithms [15, 29].

Algorithm 3.1 (Gibbs Sampler for the parameter)**.**

1. $\theta_1^i \sim p(\theta_1 \mid \theta_{2:n_\theta}^{i-1}, x_{0:T}^{i-1}, y_{1:T})$

2. $\theta_2^i \sim p(\theta_2 \mid \theta_1^i, \theta_{3:n_\theta}^{i-1}, x_{0:T}^{i-1}, y_{1:T})$

... ...

$k.$ $\theta_k^i \sim p(\theta_k \mid \theta_{1:k-1}^i, \theta_{k+1:n_\theta}^{i-1}, x_{0:T}^{i-1}, y_{1:T})$

... ...

$n_\theta - 1.$ $\theta_{n_\theta-1}^i \sim p(\theta_k \mid \theta_{1:n_\theta-2}^i, \theta_{n_\theta}^{i-1}, x_{0:T}^{i-1}, y_{1:T})$

$n_\theta.$ $\theta_{n_\theta}^i \sim p(\theta_{n_\theta} \mid \theta_{1:n_\theta-1}^i, x_{0:T}^{i-1}, y_{1:T})$

Algorithm 3.2 (Gibbs Sampler for the hidden state)**.**

0. $x_0^i \sim p(x_1 \mid x_{1:T}^{i-1}, y_{1:T}, \theta^i)$

1. $x_1^i \sim p(x_t \mid x_0^i, x_{2:T}^{i-1}, y_{1:T}, \theta^i)$

... ...

t. $x_t^i \sim p(x_t \mid x_{0:t-1}^i, x_{t-1:T}^{i-1}, y_{1:T}, \theta^i)$

... ...

T-1. $x_{T-1}^i \sim p(x_{T-1} \mid x_{0:T-2}^i, x_T^{i-1}, y_{1:T}, \theta^i)$

T. $x_T^i \sim p(x_T \mid x_{0:T}^i, y_{1:T}, \theta^i)$

If we let x_{-t} denote all the state vectors, except for the one at time t, i.e. $x_{-t} = \{x_0, \ldots, x_{t-1}, x_{t+1}, \ldots, x_T\}$, then we can write the distribution we sample from in the Algorithm 3.2 as $p(x_t \mid x_{-t}, y_{1:T}, \theta)$. Using an interplay of Bayes rule and the Markov property, we can rewrite it using the proportionality

$$p(x_t \mid x_{-t}, y_{1:T}, \theta) \propto p(x_{t+1} \mid x_t, \theta) p(y_t \mid x_t, \theta) p(x_t \mid x_{t-1}, \theta).$$

3.4 Importance Sampling Methods

Once we have a distribution f on a sample space $\mathcal{X}$, a typical application is to sample from it to approximate the expectation

$$\mathbb{E}[h(X)] = \int_{\mathcal{X}} h(x) f(x)\, dx, \tag{3.18}$$

for a given function h. However, it may not be optimal to directly sample from f because, for example, the distribution f is difficult to sample from, or interesting values of f are unlikely to show up in the samples. The principal alternative is to use importance sampling, where we replace sampling from the distribution f with sampling from a more productive distribution g [8, 14]. To approximate the expectation (3.18) we generate samples $x_1, \ldots, x_m$ from a given distribution g and calculate

$$\mathbb{E}[h(X)] \approx \frac{1}{m} \sum_{j=1}^{m} \frac{f(x_j)}{g(x_j)} h(x_j). \tag{3.19}$$

We can see this approximation from the alternate representation:

$$\mathbb{E}_f[h(X)] = \int_{\mathcal{X}} h(x) f(x)\, dx = \int_{\mathcal{X}} h(x) \frac{f(x)}{g(x)} g(x)\, dx = \mathbb{E}_g \left[h(X) \frac{f(X)}{g(X)} \right],$$

where $\mathbb{E}_f$ denotes the expectation with respect to the distribution f.

The challenge in importance sampling is the choice of the distribution g. Any distribution can be used if it is appropriate for the expectation (3.18), but some choices are better than others. Typically, we want to use a distribution where the tails of g are heavier than the ones of f so that the ratio $\frac{f}{g}$ is bounded. We can address this issue (bounded ratio) to yield a more stable estimator if we replace $\frac{1}{m}$ by $\sum_j \frac{f(x_j)}{g(x_j)}$. Then, we can estimate the expectation μ from

$$\mu = \int_{\mathcal{X}} h(x) f(x)\, dx$$

by the following algorithm:

Algorithm 3.3 (Importance Sampling)**.**

1. *Choose a distribution* g
2. *Sample* $x_1, \ldots, x_m \sim g$
3. *Calculate the importance weights* $w_j = \frac{f(x_j)}{g(x_j)}$, $j = 1, \ldots, m$

4. Approximate μ by

$$\hat{\mu} = \frac{w_1 h(x_1) + \cdots + w_m h(x_m)}{w_1 + \cdots + w_m}.$$

3.4.1 Particle Filters

Particle filtering lies in the framework of Sequential Monte Carlo (SMC) approximation and their popularity is due to their flexibility in handling nonlinear, non-Gaussian scenarios [8, 14, 18, 24].

Consider the state variables X_t evolving according to the model

$$X_t = g_t(X_{t-1}) + u_t, \tag{3.20}$$

where u_t is the driving noise and g_t is a nonlinear function. Suppose we have noisy observations Z_t derived from the state variables via

$$Z_t = h_t(X_t) + \xi_t, \tag{3.21}$$

where h_t is a nonlinear function, and ξ_t, $t = 1, \ldots, K$, are mutually independent random variables.

Our goal is to obtain an estimate of a function of the state of the system, say $f(X_t)$, from the sequence of observations. The filtering problem is to approximate the conditional expectation $\mathbb{E}[f(X_t) \mid \{Z_j\}_{j=1}^t]$ or, equivalently, to compute the conditional probability distribution $p(X_t \mid Z_1, \ldots, Z_t)$.

Because of the potential difficulty of sampling from the conditional distribution, we use the sequential importance sampling (SIS) technique. Using an alternate density $q(X_t \mid Z_1, \ldots, Z_t)$ which can be more easily sampled, and sampling N values from it, we approximate the expectation by

$$\mathbb{E}[f(X_t) \mid Z_1, \ldots Z_t] \approx \frac{1}{N} \sum_{n=1}^{N} f(X_t^n) \frac{p(X_t^n \mid Z_1, \ldots, Z_t)}{q(X_t^n \mid Z_1, \ldots, Z_t)} \tag{3.22}$$

or, by approximating N by

$$\sum_{n=1}^{N} \frac{p(X_t^n \mid Z_1, \ldots, Z_t)}{q(X_t^n \mid Z_1, \ldots, Z_t)}$$

we have

$$\mathbb{E}[f(X_t) \mid Z_1, \ldots Z_t] \approx \left(\sum_{n=1}^{N} f(X_t^n) \frac{p(X_t^n \mid Z_1, \ldots, Z_t)}{q(X_t^n \mid Z_1, \ldots, Z_t)} \right) / \left(\sum_{n=1}^{N} \frac{p(X_t^n \mid Z_1, \ldots, Z_t)}{q(X_t^n \mid Z_1, \ldots, Z_t)} \right).$$

The conditional probabilities are updated from the recursive relation

$$p(X_t \mid Z_1, \ldots, Z_t) \propto p(Z_t \mid X_t) p(X_t \mid Z_1, \ldots, Z_{t-1}),$$

where $p(Z_t \mid X_t)$ is the likelihood of the observation given the system's state, and the predictive prior $p(X_t \mid Z_1, \ldots, Z_{t-1})$ is given by

$$\int_{\mathcal{X}} p(X_t \mid X_{t-1}) p(X_{t-1} \mid Z_1, \ldots, Z_{t-1}) \, dX_{t-1},$$

where $p(X_t \mid X_{t-1})$ is the Markov transition probability based on the stochastic dynamics of the state system. For the Particle Filter, we use the predictive prior as the alternative density, i.e.

$$q(X_t \mid Z_1, \ldots, Z_t) = p(X_t \mid Z_1, \ldots, Z_{t-1}).$$

Sampling from this density, we form the weights

$$w_t^n = \frac{p(Z_t \mid X_t)}{\sum_{k=1}^{N} p(Z_t \mid X_t^k)}$$

and then we have the approximation

$$\mathbb{E}[f(X_t) \mid Z_1, \ldots, Z_t] \approx \sum_{n=1}^{N} w_t^n f(X_t^n).$$

In many applications, many of the samples generate negligible weights and therefore do not contribute to the approximation [32]. One solution is to resample the weights to create more copies of samples with significant weights.

Algorithm 3.4 (Particle Filter with resampling)**.**

Given X_{t-1}^i *for* $i = 1, \ldots, N$

1. *Sample* $X_t^i \sim p(X_t^i \mid X_{t-1}^i)$
2. *Compute the weight through* $w_t^i = \frac{p(Z_t|X_t)}{\sum_{k=1}^N p(Z_t|X_t^k)}$
3. *Repeat from Step 1.*
4. *Generate* $\tilde{N}$ *samples of* $U \sim U(0,1)$
5. *Set* $\tilde{X}_t^j = X_t^i$ *if* $\sum^j w_t^i \leq U^i \leq \sum^{j+1} w_t^i$

3.5 Kalman Filter

In this section, we focus on a method which is optimal when the dynamic and the observation processes are linear and Gaussian [17, 35]. In this case, the model dynamics are given by

$$X_{t+1} = F_t X_t + V_t, \tag{3.23}$$

where V_t is the normally distributed noise, i.e. $V_t \sim N(0, Q_t)$. We also have an observation vector Z_t defined by

$$Z_t = H X_t + W_t, \tag{3.24}$$

where H is a $M \times N$ matrix, with $M < N$, capturing the fact that not all components of the state vector X_t may be part of the observation, and W_t is a zero-mean Gaussian random vector with covariance matrix R.

From equation (3.23), the Kalman Filter predictor gives the prior estimate $X_{t+1|t} = F_t X_{t|t}$ and that the potential error is measured by the covariance matrix $P_{t+1|t} = F_t P_{t|t} F_t^T + Q_t$. After extrapolating the state vector to time $t+1$, the prediction will be updated by taking into account the actual observation Z_{t+1}, giving the Kalman Filter update

$$X_{t+1|t+1} = X_{t+1|t} + K(Z_{t+1} - HX_{t+1|t}), \tag{3.25}$$

where the Kalman gain-matrix is given by

$$K = P_{t+1|t} H^T (H P_{t+1|t} H^T + R)^{-1}.$$

The error is measured by the covariance matrix $P_{t+1|t+1} = (I - KH) P_{t+1|t}$.

The innovation $S_{t+1} = Z_{t+1} - HX_{t+1|t}$ indicates the degree to which the actual measurement Z_{t+1} differs from the predicted measurement $HX_{t+1|t}$, and K determines the degree to which the predicted state $X_{t+1|t}$ should be corrected to reflect their deviation. The resulting algorithm follows:

Algorithm 3.5 (Kalman Filter)**.**

Given $X_{t|t}$ and $P_{t|t}$

1. *Compute $X_{t+1|t} = F_t X_{t|t}$*
2. *Compute $P_{t+1|t} = F_t P_{t|t} F_t^T + Q_t$*
3. *Compute optimal $K = P_{t+1|t} H^T (H P_{t+1|t} H^T + R)^{-1}$*
4. *Update $X_{t+1|t+1} = X_{t+1|t} + K(Z_{t+1} - HX_{t+1|t})$*
5. *Update $P_{t+1|t+1} = (I - KH) P_{t+1|t}$*
6. *Repeat from Step 1. with $t = t + 1$*

Chapter 4

Our Problem

4.1 Modeling Stock Prices and Stochastic Volatilities

We focus now on a financial problem which has been widely studied and investigated. In this thesis, we explore a stock price model enhanced with a pertinent volatility model through filtering techniques. The stock price, say S, grows at a deterministic rate, called the drift, r. The complexity of analyzing stock prices follows from random movement, which may lead to unexpected spikes one sees in the market. Since different stocks respond differently to market spikes, let σ represent the stock's volatility but can more practically be thought of as the risk of the stock. We consider the stock prices model

$$dS = rSdt + \sigma SdW, \tag{4.1}$$

where W represents the driving noise, in our case a Brownian motion. According to equation (4.1), a low volatility implies the stock will behave nearly deterministically according to the growth parameter r. On the other hand, a large volatility means the stock price is likely to experience large spikes in pricing.

In order to simulate stock prices, we first discretize equation (4.1) using a basic Euler scheme [20]. Consider a given time discretization $0 = t_0 < t_1 < \cdots < t_n < t_N = T$ of the time interval $[0, T]$

$$S_{t_{n+1}} = S_{t_n} + rS_{t_n}\Delta t_n + \sigma S_{t_n}\Delta W_{t_n}, \tag{4.2}$$

where $\Delta W_{t_n} \sim N(0, \Delta t_n)$, $\Delta t_n = t_{n+1} - t_n$. For simplicity, we consider $\Delta t_n = 1$. We simulate the stock prices, and show the resulting simulation below in Figure 4.1 with two different volatilities.

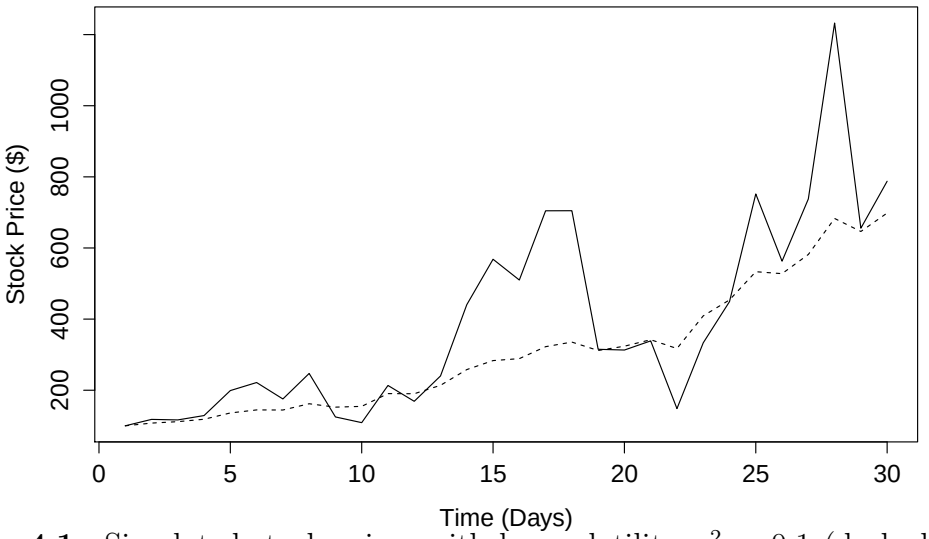

Figure 4.1: Simulated stock prices with low volatility $\sigma^2 = 0.1$ (dashed line) and high volatility (solid line) $\sigma^2 = 1$ with growth rate, $r = 0.05$

Equation (4.1) assumes that the volatility, σ is time invariant; however, in practice, volatility changes over time instead of remaining constant. Thus, we model the volatility's behavior via the following equation

$$log(\sigma_t^2) = \nu + \phi log(\sigma_{t-1}^2) + \eta w, \tag{4.3}$$

where the driving noise, $w \sim N(0,1)$. We denote the hidden state variable $x_t = log(\sigma_t^2)$, which we need to estimate. We also observe the relative change in stock price y_t, given by the observation model

$$y_t = e^{x_t/2}v, \tag{4.4}$$

where the driving noise v is considered standard normal as well.

4.2 Application of Methods

Based on the discussion of filtering methods in Chapter 3, we attempt to estimate the log-volatility given by equation (4.3) based on the observation model as described in equation (4.4). Consider the following dynamic system

$$\text{Initial distribution: } X_0 \sim N(0, \eta^2) \tag{4.5}$$

$$\text{Transition density: } X_t \mid X_{t-1} = x_{t-1} \sim N(\nu + \phi x_{t-1}, \eta^2) \tag{4.6}$$

$$\text{Observation density: } Y_t \mid X_t = x_t \sim N(0, e^{x_t}) \tag{4.7}$$

We examine three different approaches to the stochastic log-volatility problem. The volatility, which is updated through its own dynamic model, was estimated by drawing inference from observerable changes in the stock price. In comparison to many filtering problems, like those found in tracking, where the unknown appears in the mean of the observation density, the volatility is related to the observable stock price through the variance. Drawing inference on the variance of a random variable only given one observation at each time step is a difficult task. Our methods, which all build on this relationship, did not preform as well as one would hope as shown later in the results (Section 4.3).

4.2.1 Kalman Filter

In order to apply the Kalman Filter, we must first satisfy the assumption of linearity, which is violated by the observation equation. We transform the observation equation by squaring both sides before taking the natural logarithm

$$y_t^* = \log y_t^2 = x_t + \log v^2, \tag{4.8}$$

making the equation linear in form. Following the work of [19], we approximate the distribution of $\log v^2$ where $v \sim N(0,1)$ with a mixture of normal distributions.

$$\log v^2 \approx \sum q_i N(m_i, s_i^2) \tag{4.9}$$

where the parameters used in equation (4.9) are defined in Table 4.1.

Table 4.1: Parameters for the Mixture of Normal Distributions defined in equation (4.9)

i	q_i	m_i	s_i^2
1	0.00730	-10.12999	5.79596
2	0.10556	-3.97281	2.61369
3	0.00002	-8.56686	5.17950
4	0.04395	2.77786	0.16735
5	0.34001	0.61942	0.64009
6	0.24566	1.79518	0.34023
7	0.25750	-1.08819	1.26261

Below is given a pseudo-code for the Kalman Filter estimates.

Algorithm 4.1 (Kalman Filter)**.**

1. *Initialize* $x_{0|0} \sim N(0, \eta^2)$ *with certainty* $P_{0|0} = \eta^2$

2. *Compute* $x_{t|t-1} = \nu + \phi x_{t-1|t-1}$

3. *Compute* $P_{t|t-1} = \phi^2 P_{t-1|t-1} + \eta^2$

4. *Randomly choose a normal distribution from mixture:*

   (a) *Sample* $U \sim U(0,1)$

   (b) *Select* j *by* $\sum^{j} q^i \leq u \leq \sum^{j+1} q^i$

5. *Compute* $K = \frac{P_{t|t-1}}{P_{t|t-1}+s_j^2}$

6. *Update* $x_{t|t} = x_{t|t-1} + K(y_t - (m_j + x_{t|t-1}))$

7. *Update* $P_{t|t} = (1-K)^2 P_{t|t-1} + (Ks_j)^2$

8. *Repeat from Step 2. with* $t = t+1$ *until* $t = T$

After applying the Kalman Filter, we provide the following numerical results. Figure 4.2 shows the estimated log-volatility values in comparison to the true values. The Kalman Filter does not match the dynamics perfectly. In fact, at times we see extreme spikes in the Kalman Filter's estimates in contrast to relatively small changes in the true. Figure 4.3 displays a plot of the resulting absolute error. The error is consistently around 2 and has very large spikes of over 5 error. Note that for this problem, an error of 2 is also very large with respect to log-volatility with values near 2.

4.2.2 Gibbs Sampling

Next, we apply the Gibbs Sampling method to our problem. For our model the parameter space is $\theta = (\phi, \nu, \eta^2)$. In order to sample from the posterior distributions, we first construct prior distributions for θ and x_0, following the work of [6].

$$\phi \sim N(a, b^2) \tag{4.10}$$

$$\nu \sim N(c, d^2) \tag{4.11}$$

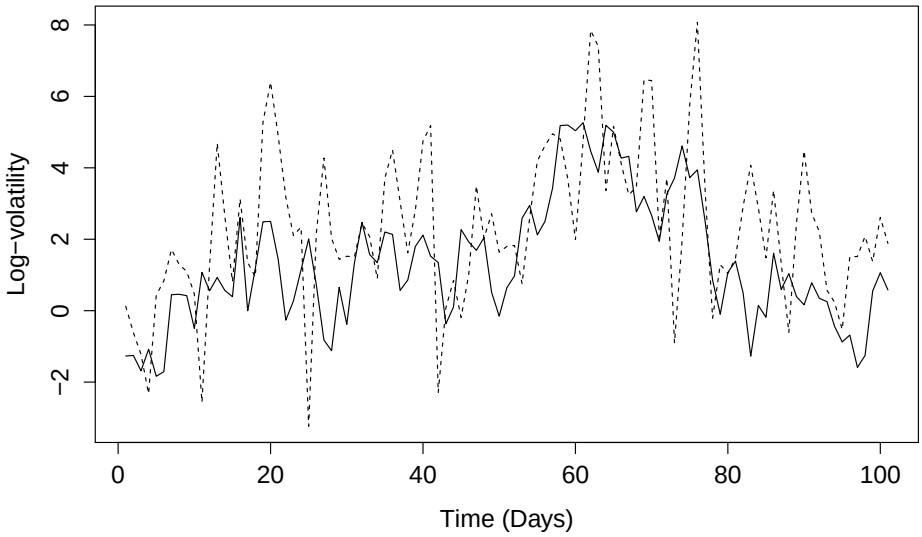

Figure 4.2: Plot of True Log-volatility (solid) and estimates from Kalman Filter (dashed) with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100$

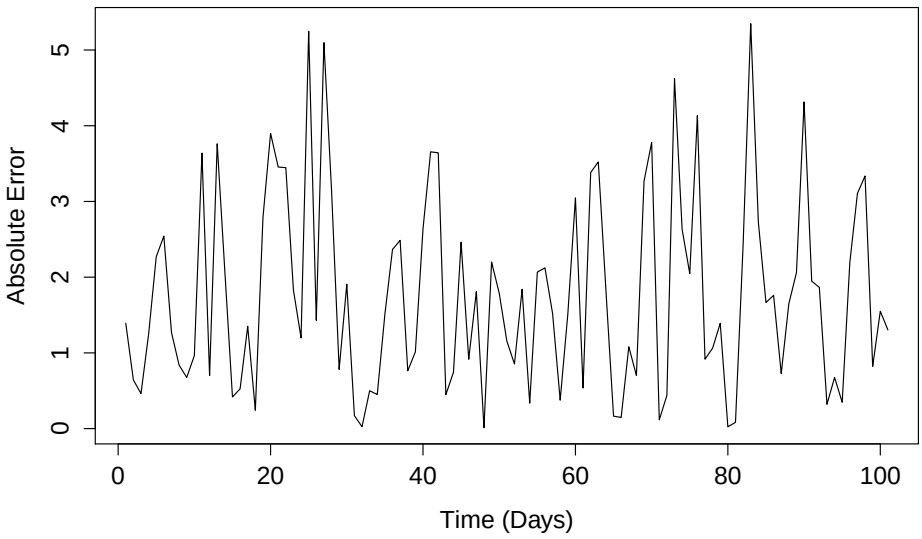

Figure 4.3: Plot of absolute difference in Kalman Filter estimates and true values with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100$

$$\eta^2 \sim IG(\alpha, \beta) \tag{4.12}$$

$$x_0 \sim N(0, \eta^2) \tag{4.13}$$

Using these priors one constructs the full conditional posterior distributions as defined in Section 3.3 from which we will sample. The detailed calculations can be found in [6]. The following is pseudo-code for generating Gibbs Sampling estimates:

Algorithm 4.2 (Gibbs Sampler)**.**

[1.] First simulate ϕ

$$\phi^i \mid \theta^{i-1}, x_{0:T}^{i-1}, y_{1:T} \sim N(\tilde{a}, \tilde{b}^2) \tag{4.14}$$

$$\tilde{b}^2 = \left(\frac{1}{b^2} + \frac{1}{\eta^{2^{i-1}}} \sum (x_{t-1}^{i-1})^2 \right)^{-1} \tag{4.15}$$

$$\tilde{a} = \tilde{b}^2 \left(\frac{a}{b^2} + \frac{1}{\eta^{2^{i-1}}} \sum x_{t-1}^{i-1} (x_t^{i-1} - \nu^{i-1}) \right) \tag{4.16}$$

[2.] Next, sample ν

$$\nu^i \mid \theta^{i-1}, x_{0:T}^{i-1}, y_{1:T} \sim N(\tilde{c}, \tilde{d}^2) \tag{4.17}$$

$$\tilde{d}^2 = \left(\frac{1}{d^2} + \frac{T}{\eta^{2^{i-1}}} \right)^{-1} \tag{4.18}$$

$$\tilde{c} = \tilde{d}^2 \left(\frac{c}{d^2} + \frac{1}{\eta^{2^{i-1}}} \sum x_t^{i-1} - \phi^i x_{t-1}^{i-1} \right) \tag{4.19}$$

[3.] The last parameter of interest is η^2

$$(\eta^i)^2 \mid \theta^i, x_{0:T}^{i-1}, y_{1:T} \sim IG(\tilde{\alpha}, \tilde{\beta}) \tag{4.20}$$

$$\tilde{\alpha} = \alpha + \frac{T+1}{2} \tag{4.21}$$

$$\tilde{\beta} = \beta + \frac{1}{2}\sum(x_t^{i-1} - \nu^i - \phi^i x_{t-1}^{i-1})^2 + (x_0^{i-1})^2 \tag{4.22}$$

[4.] After updating our parameters, we compute the posterior for the hidden variable x_t

$$x_t^i \mid \theta^i, x_{-t}^{i-1}, y_{1:T} \sim N(\mu, \sigma^2) \tag{4.23}$$

$$\sigma^2 = \left(\frac{1 + (\phi^i)^2}{(\eta^i)^2} + \frac{1}{2}\right)^{-1} \tag{4.24}$$

$$\mu = \sigma^2((\nu^i(1 - \phi^i) + \phi^i(x_{t-1}^i + x_{t+1}^{i-1}))(\eta^i)^{-2} + \log y_t) \tag{4.25}$$

We next analyze the numerical results of the Gibbs Sampling. We first present the results for the model parameter estimation in Table 4.2. The method produced very accurate estimates for each of the parameters. As before, we plot the estimated log-volatility values in comparison to the true values in Figure 4.4. We see a much better fit from the Gibbs Sampling estimates in comparison to the Kalman Filter estimates. They follow the random movement of the true values much more accurately; although, there is an obvious issue of under approximation in this simulation. Figure 4.5 shows the resulting absolute different between the true values and Gibbs Sampling estimates. The errors are much lower than those from the Kalman Filter. Again we see a large spike of error nearing 5, but unlike the Kalman Filter, there is only one spike of this magnitude.

Table 4.2: Parameter estimation using Gibbs Sampling with $N = 5000$ samples

Parameter	True Value	Estimate
ϕ	0.9	0.9035
ν	0.1	0.0972
η^2	1.0	1.0046

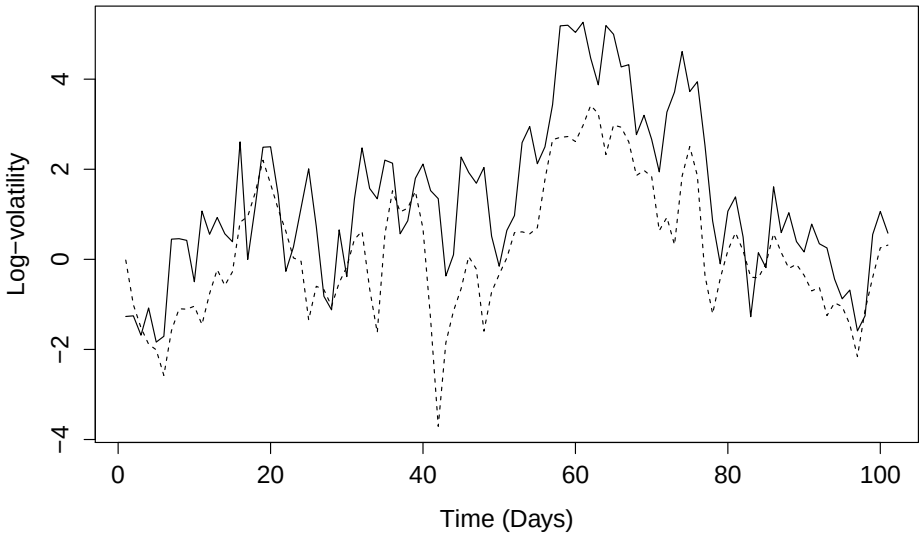

Figure 4.4: Plot of True Log-volatility (solid) and estimates from Gibbs Sampling (dashed) with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

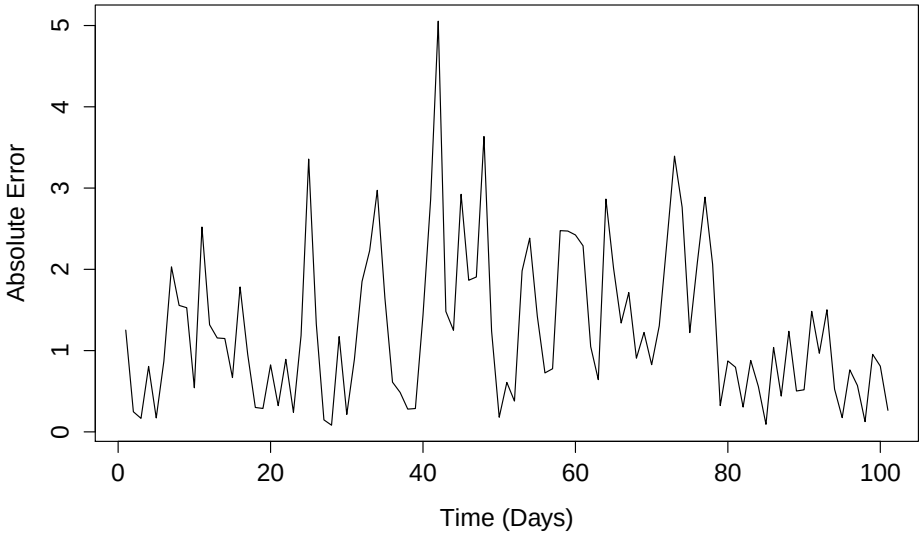

Figure 4.5: Plot of absolute difference in Gibbs Sampling estimates and true values with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

4.2.3 Particle Filtering

Lastly, we apply the Particle Filter as defined in Section 3.4.1. The dynamics model and observation model provide the following densities:

$$X_t \mid x_{t-1} \sim N(\nu + \phi x_{t-1}, \eta^2) \tag{4.26}$$

$$Y_t \mid x_t \sim N(0, e^{x_t}). \tag{4.27}$$

Taking our proposal density $Q(X_t \mid Y_t, X_{t-1})$ to be the transition density $P(X_t \mid X_{t-1})$, we have the following algorithm.

Algorithm 4.3 (Particle Filter)**.**

1. *Initialize N samples of $X_0^i \sim N(0, \eta^2)$*
2. *Sample $X_t^i \sim N(\nu + \phi x_{t-1}, \eta^2)$, for $i = 1, \ldots, N$*
3. *Compute the weights $w_t^i = w_{t-1}^i p(y_t \mid x_t^i)$ using $N(0, e^{x_t})$*
4. *Resample X_t^i based on the weights w_t^i*
   (a) *Sample $U^i \sim U(0, 1)$, for $i = 1, \ldots, \tilde{N}$*
   (b) *Set $\tilde{X}_t^j = X_t^i$ if $\sum^j w_t^i \leq U^i \leq \sum^{j+1} w_t^i$*
5. *Set $t = t + 1$ and repeat from Step 2.*
6. *Take the newly weighted average $E(X_t \mid Y_t) = \frac{1}{\tilde{N}} \sum \tilde{x}_t^i$*

Lastly, we analyze the estimates of the Particle Filter. Figure 4.6 shows the estimated log-volatility values in comparison to the true values. The estimates do a very poor job of approximating the true values. They have the opposite issue of the Gibbs Sampling estimates, which under approximated, by over approximating the true log-volatility from the beginning. In this simulation, the most accurate estimates occur when log-volatility spikes upwards around $t = 60$, closing in on the much larger

Particle Filter estimates. Figure 4.7 shows the resulting absolute error. As expected, the errors are consistently large, with most ranging between 3 and 4. The only place of decent estimates occurs around time $t = 60$ as mentioned above.

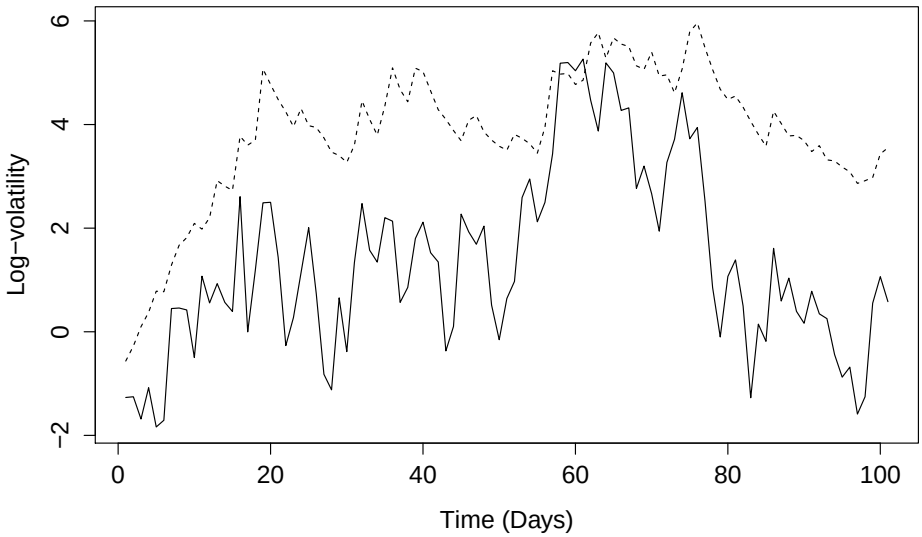

Figure 4.6: Plot of True Log-volatility (solid) and estimates from Particle Filter (dashed) with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

4.3 Comparison

With results from all three methods, we now can compare the estimates of each method for log-volatility. The true values are randomly simulated beforehand using equations (4.3) and (4.4). For a genuine comparison, we ensure that the algorithms are run on the same randomly generated data. Due to the randomness incorporated into the data, the results from each simulation can vary, making drawing definite conclusions a difficult task. For instance, Kalman Filter may produce the best results on one simulation but perform much worse then the others in another case. For comparison, we look at the results for three independent simulations.

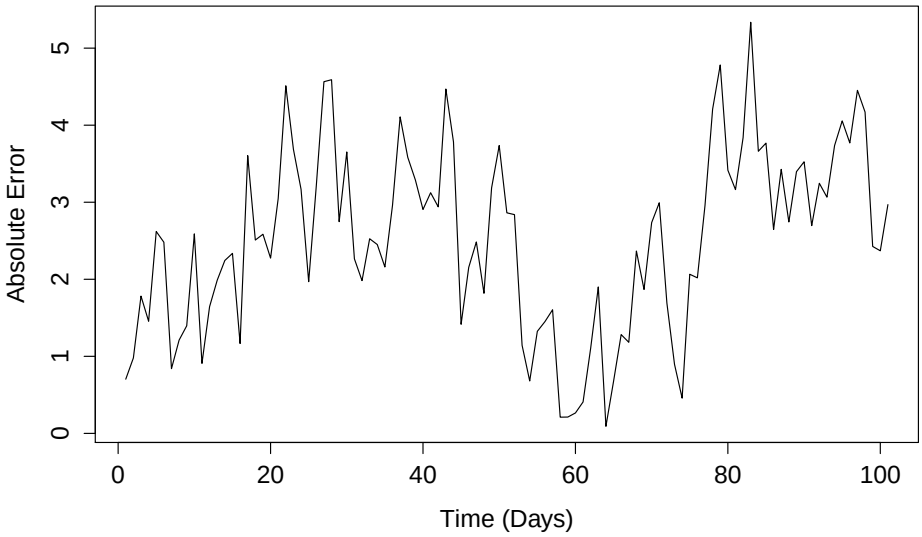

Figure 4.7: Plot of absolute difference in Particle Filter estimates and true values with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

In the first simulation, we compile the numerically results shown individually before into Figure 4.8. Kalman Filter produces estimates that drastically spike above and below the true values; while the Gibbs Sampling and Particle Filter consistently under and over approximate the values respectively. For further comparison, Figure 4.9 shows the absolute error of the three methods. All three methods experienced large error spikes of around magnitude 5 in estimating log-volatility, which typically took on values less than 2. Despite the overall poor approximation by all three methods, the Gibbs Sampling produced the most accurate estimates. We provide various statistics on the errors in Table 4.3, which shows that the Gibbs Sampling did in fact produce the most accurate estimates. Kalman Filter, despite the large spikes, on average did the second best. As expected from the plots, the Particle Filter performed the worst of the methods.

For the second simulation, Figure 4.10 shows the estimated log-volatility values in comparison to the true values for the three methods. The estimates exhibit the same

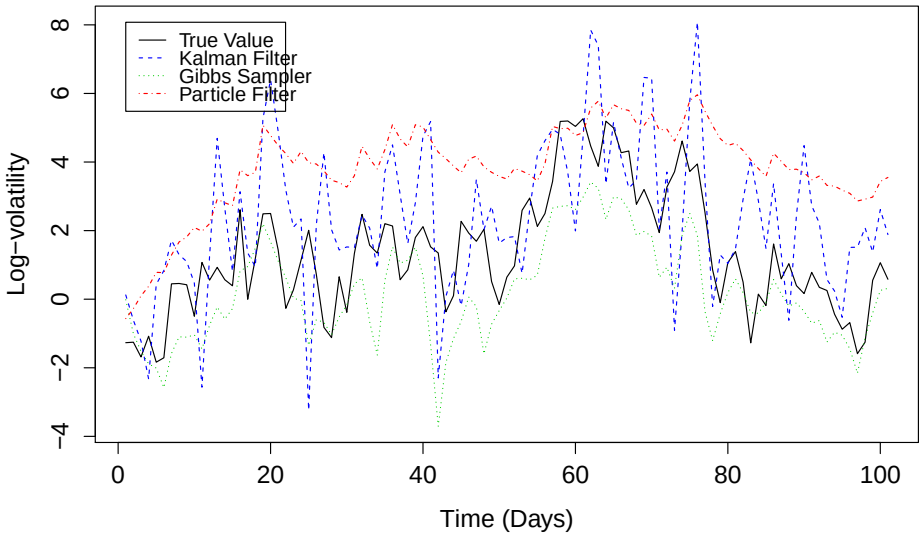

Figure 4.8: First simulation: Plot of true Log-volatility in comparison with all three approaches with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

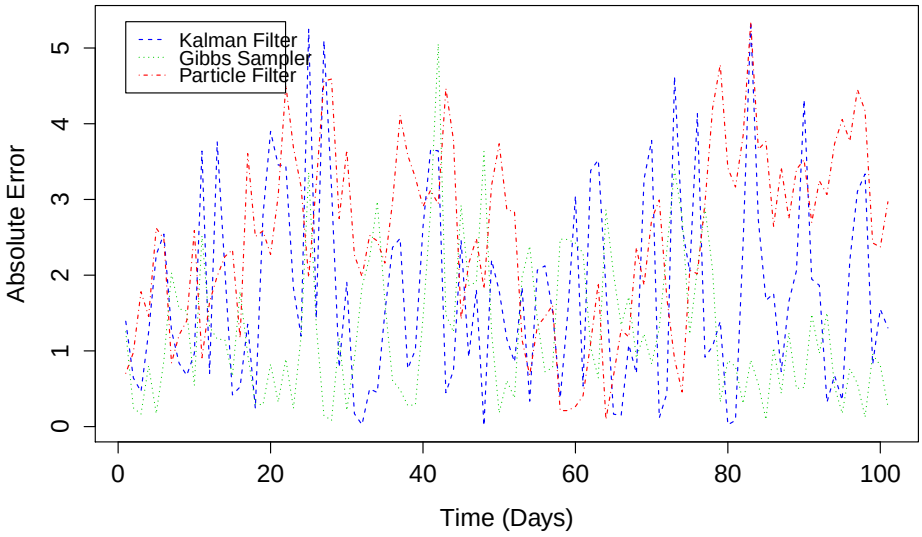

Figure 4.9: First simulation: Plot of absolute difference in estimates and true values with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

Table 4.3: Comparison of Methods in First Simulation

	Kalman Filter	Gibbs Sampling	Particle Filter
Average Error	1.7792	1.271655	2.514508
Max Error	5.347452	5.053752	5.335677
Min Error	0.01159551	0.0820308	0.09189744
Std Dev	1.31388	0.9515875	1.190669

issues as in the first simulation. The Kalman Filter goes to extreme in predicting the random movement, resulting in large spikes. The Gibbs Sampling is much more accurate; although it under approximates the true values. In contrast, the Particle Filter over approximates the true log-volatility with very poor estimates. The errors are shown in Figure 4.11. We see similar results to that of the first simulation, where Gibbs Sampling appears to do the best, but all methods fail to precisely estimate the log-volatility. In fact, for this simulation, the Particle Filter achieves an error of magnitude 6. Again for comparison, we provide various statistics on the errors in Table 4.4, which shows the same ranking of methods by average error. Gibbs Sampling did the best, followed by the Kalman Filter, and lastly, the Particle Filter did the worst. Furthermore, Gibbs Sampling did better on this simulation then the one previous, and Particle Filter did significantly worst.

Table 4.4: Comparison of Methods in Second Simulation

	Kalman Filter	Gibbs Sampling	Particle Filter
Average Error	1.861696	1.09255	3.2037
Max Error	5.1842	3.664468	6.253594
Min Error	0.03197209	0.03528706	0.0165376
Std Dev	1.355502	0.8520093	1.295717

In the final simulation we examined, Figure 4.12 shows the estimated log-volatility values in comparison to the true values for the three methods. The numerical results reflect what we have seen in previous simulations. A point of interest is the two extreme downward spikes in estimates produced by both the Kalman Filter and Gibbs

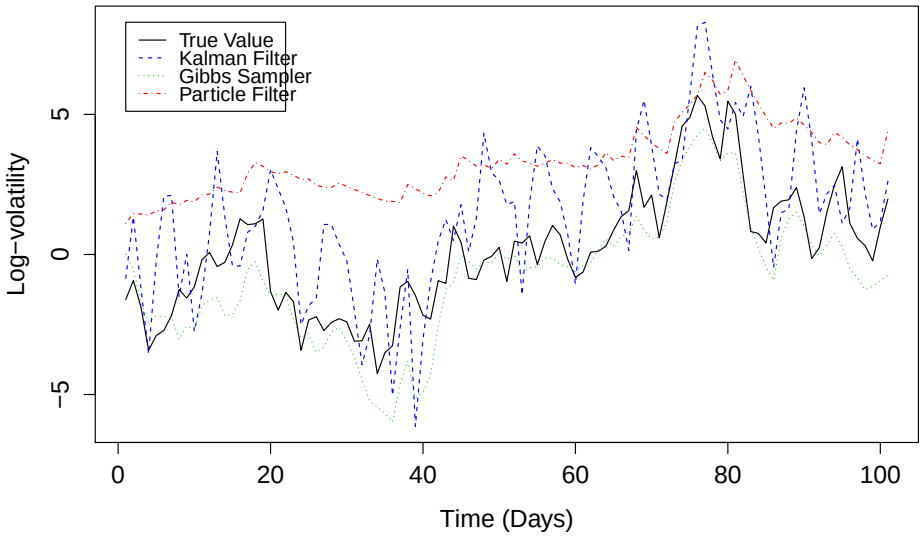

Figure 4.10: Second Simulation: Plot of True Log-volatility in comparison with estimates from each of the three approaches with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

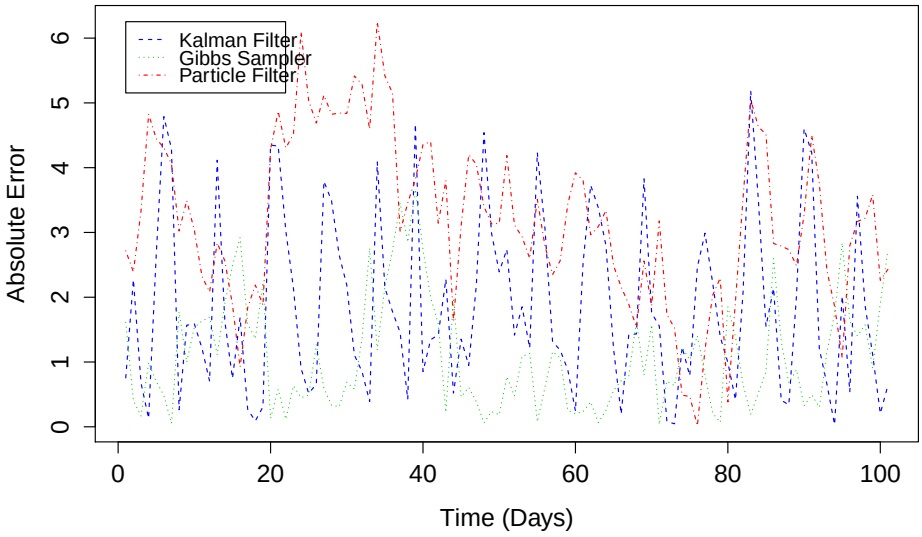

Figure 4.11: Second Simulation: Plot of absolute difference in estimates and true values with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

Sampler. Likely both methods responded similarly to a misleading observation. We see the affect this had on error in Figure 4.13. As expected, we see the largest error of the three simulations of over 7. Besides this spike, the errors are consistent with the previous simulations. Table 4.5 provides the usual statistics on error, which closely resemble the previous numerical results.

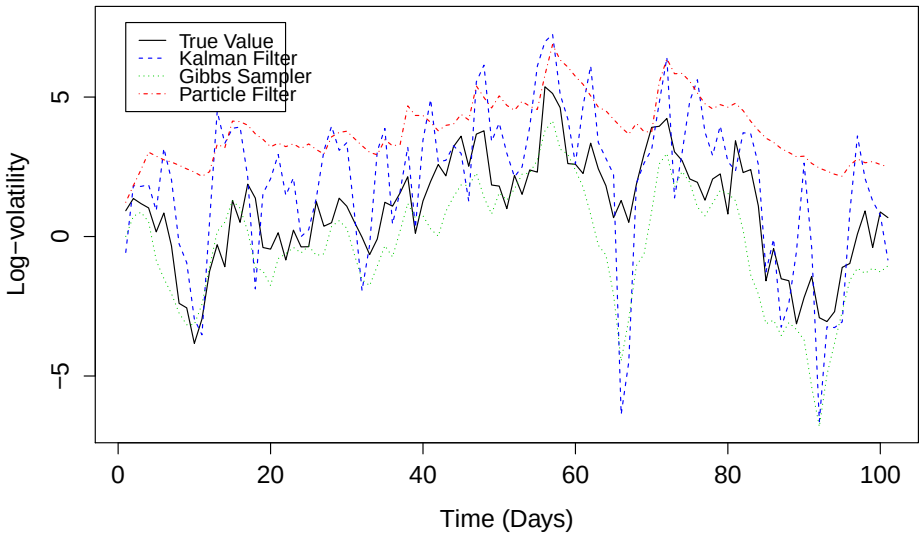

Figure 4.12: Third Simulation: Plot of True Log-volatility in comparison with all three approaches with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

Table 4.5: Comparison of Methods in Third Simulation

	Kalman Filter	Gibbs Sampling	Particle Filter
Average Error	1.698134	1.36461	2.862675
Max Error	7.670616	5.752662	6.141474
Min Error	0.02012137	0.001656477	0.1669187
Std Dev	1.32381	1.049757	1.259345

Lastly, we ran 50 independent simulations and compiled the error statistics in Table 4.6.

In conclusion, none of the methods was able to accurately estimate log-volatility; although Gibbs Sampling came the closest. The Kalman Filter is proven to be the best

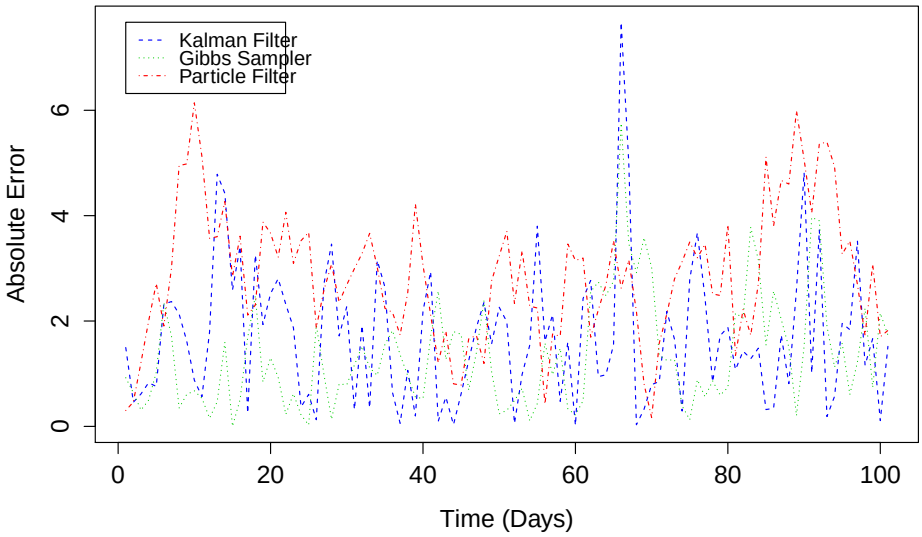

Figure 4.13: Third Simulation: Plot of absolute difference in estimates and true values with parameters: $\nu = 0.1$, $\phi = 0.9$, $\eta^2 = 1, T = 100, N = 5000$

Table 4.6: Comparison of Methods Over All Simulations

	Kalman Filter	Gibbs Sampling	Particle Filter
Average Error	1.79465	1.308611	2.79216
Max Error	11.1679	7.6599	8.8153
Min Error	0.0002	0.0001	0.0003
Std Dev	1.2944	0.9871	1.4226

method when dealing with linear models with Gaussian noise. Although we linearized our observation equation, it came at the cost of approximating the newly transformed noise. The optimality of the Kalman Filter kept the results competitive with that of the Gibbs Sampler, but did not outperform it due to the error added from this approximation. The Gibbs Sampler uses a full conditional distribution for sampling. The power of using a range of estimates over time generated more information to predict the stock's variance. The method also accurately estimated the unknown parameters of the model. Lastly, the Particle Filter relies on the observation density for weighing the more likely particles. One would hope to see a correlation between

the accuracy of an estimate and the size of the weight; however, this was not the case for our problem. In conclusion, the Particle Filter did the worse of the three methods in estimating the volatility, while Gibbs Sampling performed the best. This conclusion comes from looking across several simulations both individually and in total. Despite Gibbs Sampling producing the most accurate estimates of the three methods, an average error of around 1 is still significant when the true values often take small values in the range of -2 to 2. Therefore, further investigation on the topic should take place.

Bibliography

[1] (9/30/2000). In praise of Bayes. Economist. 12

[2] Allen, L. (2011). *An Introduction to Stochastic Processes with Applications to Biology*. Chapman & Hall, second edition. 14, 16

[3] Bain, A. and Crisan, D. (2009). *Fundamentals of Stochastic Filtering*. Springer. 3

[4] Black, F. and Scholes, M. (1972). The valuation of option contracts and a test of market efficiency. *Journal of Finance*, 27:399–417. 1

[5] Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31:307–327. 2

[6] Casarin, R. (2004). Bayesian Monte Carlo filtering for stochastic volatility models. *Cahier du CEREMADE*. 30, 32

[7] Delle Monache, L., Nipen, T., Liu, Y., Roux, G., and Stull, R. (2011). Kalman Filter and analog schemes to postprocess numerical weather predictions. *Monthly Weather Review*, 139. 3

[8] Doucet, A., de Freitas, N., and Gordon, N. (2001). *Sequential Monte Carlo Methods in Practice*. Springer. 3, 18, 21, 22

[9] Durrett, R. (1999). *Essentials of Stochastic Processes*. Springer. 14, 16

[10] Engle, R. (1982). Autoregressive conditional heteroskedasticity with estimates of the variance of U.K. inflation. *Econometrics*, 50:987–1008. 2

[11] Engle, R. and Ng, V. (1993). Measuring and testing the impact of news on volatility. *Journal of Finance*, 48:1749–1778. 3

[12] Fosdick, L. (1963). Monte Carlo calculations on the Ising lattice. *Methods Computational Physics*, 1:245–280. 5

[13] Geman, S. and Donald, G. (1984). Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6:721–741. 5, 18

[14] Ghosh, J., Delampady, M., and Samnta, T. (2006). *An Introduction to Bayesian Analysis Theory and Methods*. Springer. 18, 21, 22

[15] Hastings, W. (1970). Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika*, 57:97–109. 5, 19

[16] Hull, J. and White, A. (1987). The pricing of options on assets with stochastic volatilites. *Journal of Finance*, 42:281–300. 6

[17] Kalman, R. (1960). A new approach to linear filtering and prediction problems. *Journal of Basic Enginering*, 82:35–45. 3, 24

[18] Kang, K. and Maroulas, V. (To Appear). Drift homotopy for a nonGaussian filter. *The proceedings of information fusion*. 6, 22

[19] Kim, S., Shephard, N., and Chib, S. (1998). Stochastic volatility: Likelihood inference and comparison with ARCH models. *Review of Economic Studies*, 65:361–393. 29

[20] Kloeden, P. and Platen, E. (1992). *Numerical Solution of Stochastic Differential Equations*. Springer. 27

[21] Liu, J. and Chen, R. (1993). Sequential Monte Carlo methods of dynamic systems. *Journal of the American Stastical Association*, 93:1032–1044. 5

[22] Mahler, R. (2007). *Statistical Multisource-Multitarget Information Fusion*. Artech House Publishers. 3

[23] Maroulas, V. and Mahler, R. (2013). Tracking spawning objects. *IET Radar, Sonar, & Navigation*, 7:321–331. 3

[24] Maroulas, V. and Stinis, P. (2011). Improved particle filters for multi-target tracking. *Journal of Computational Physics*, 231:602–611. 3, 6, 22

[25] Maroulas, V. and Xiong, J. (2013). Large deviations for optimal filtering with fractional Brownian motion. *Stochastic Processes and their Applications*, 123:2340–2352. 3

[26] Metropolis, N., Rosenbluth, A., Rosenbluth, M., Teller, A., and Teller, E. (1953). Equation of state calculations by fast computing machines. *Journal of Chemical Physics*, 21:1087–1092. 5

[27] Metropolis, N. and Ulam, S. (1949). The Monte Carlo method. *Journal of the American Statistical Assocation*, 44:335–341. 4

[28] Nebenfuhr, A. e. a. (1999). Stop-and-go movements of plant Golgi statcks are mediated by the actomyosin system. *Plant Physiology*, 121:1127–1141. 3

[29] Richey, M. (2010). The evolution of Markov Chain Monte Carlo methods. *The American Mathematical Monthly*, 117:383–413. 4, 19

[30] Rubinstein, M. (1985). Nonparametric tests of alternative option pricing models using all reported trades and quotes on the 30 most active CBOE option classes from August 23, 1976 through August 31, 1978. *Journal of Finance*, 40:455–480. 2

[31] Scott, I. (1987). Option pricing when the variance changes randomly: Theory, estimation, and an application. *Journal of Financial and Quantitative Analysis*, 22:727–752. 6

[32] Synder, C., Bengtsson, T., Bickel, P., and Anderson, J. (2008). Obstacles to high-dimensional Particle Filtering. *Monthly Weather Review*, 136:4629–4640. 24

[33] von Neumann, J. (1951). Various techniques used in connection with random digits. *National Bureau of Standards Applied Mathematics Series*, 12:36–38. 9

[34] Weare, J. (2009). Particle filtering with path sampling and an applicaiton to a bimodal ocean current model. *Journal of Computational Physics*, 228:4312–4331. 3

[35] Welch, G. and Bishop, G. (2004). An introduction to the Kalman Filter. 24

[36] Wiggins, J. (1987). Option values under stochastic volatilites. *Journal of Financial Economics*, 19:351–372. 6

[37] Xiong, J. (2008). *An Introduction to Stochastic Filtering Theory*. Oxford. 3

Vita

John Collins was born December 14, 1990, in Ann Arbor, Michigan. At a young age, he and his family moved to Knoxville, Tennessee. John decided to remain in his hometown after finishing high school at Bearden by attending the University of Tennessee, Knoxville. He began his undergraduate career Fall 2009 in the field of mathematics, following in the footsteps of his father, Chuck Collins who is a mathematics professor at the university. Three years later in May 2012, John graduated from the University of Tennessee with his Bachelors of Science in mathematics as well as a minor in both statistics and economics. His education did not end there as he returned to the University of Tennessee in Fall 2012 pursuing his Master's in mathematics.