The estimation and filtering for the Stochastic Volatility model

Bingsheng Chen
Supervisor: Prof Michael Pitt

October 5, 2023

Abstract

Time series have wide applications in many fields, including economic and physics. Inference in time series, however, is not easy since it involves the techniques of filtering and parameter estimation. In this project, we focus on a specific class of model- stochastic volatility model, whose volatility depends on a latent AR(1) process. We performed filtering and parameter estimation with three different methods: Kalman filter with Quasi-MLE, Bayesian approach via Markov Chain Monte-Carlo and particle filter with likelihood estimation. As a conclusion, Kalman filter can certainly provide reasonable estimations for both parameters and states, but the state estimations become noisy at a low signal to noise ratio. Meanwhile, Quasi Maximum likelihood is also sub-optimal, due to the poor log-normal approximation to chi-square distribution. The Bayesian approach via Monte-Carlo provides optimal estimators for both state and parameters. However, this method has the issue of low-convergence rate and computational inefficiency. We finally illustrated the technique of particle filter(including SIR and Auxiliary SIR algorithm), with its application to stochastic volatility model and proved the likelihood estimator is unbiased.

Contents

1 Introduction

Financial times series display many complex statistical behaviours as seen in non-equilibrium system, such as scaling fractal, self-organization[1]. However, analyzing statistical properties of time series model is not straight forward, since the data might involves several noise sources. State space model is one of the most common models to solve this problem, it assumes that the observations from a times series evolves dynamically according to latent state variables, where the relations between state variables and the observations were characterised via a set of difference equations[3]. Solving state space models for a set of data require techniques of parameter estimation and filtering, which is not simple. *Filtering* in this project is a technique that can be used find an optimal estimator for the true latent process and *parameter estimation* is a technique that can compute the most "likely" parameters was used in the model via maximising posterior. In the practice, both parameters and states are missing and there are several methods to solve the state space models, including Kalman filter, Monte-Carlo Sampling and more recent technique, paricle filter. Kalman filter provide a recursive computational efficient method to solve linear state space model that minimizes the mean of the squared error, however, if the state space model is highly non-linear and non-Gaussian, the Kalman was unable to produce a good estimate[2]. If the posterior or conditional posterior for states or parameter are tractable, Markov Chain Monte-Carlo method can provide a very good estimate for both parameters and states, via sampling parameters or state from posterior via Gibbs sampling. However, the rate of convergence may be very slow and computationally inefficient. The most recent developing technique after 1990, particle filters are built from the method of Markov Chain Monte-Carlo that can solve the filtering and estimation problem numerically from simulations, which deliver optimal estimator for likelihood[4].

Stochastic volatility is a model that displayed heavier tail than normal, which implies rare event are more likely to occur compare to normal distribution. The volatility of the observations depend on the latent state variable. Furthermore, the latent state variable follows an AR(1) process. In this project, we use the data generated from stochastic volatility model to test the performance of the parameter estimation and state filter from three methods: Kalman filter, MCMC and particle filter. This project is going to be separated into four part, in the first part, we introduced the general state space model and investigated the property of the stochastic volatility model. In the second section, we reviewed A.C. Harvey's book[13] and derived the Kalman filter and demonstrate the how it can be applied to the state space model. We further use Quasi-MLE approach to perform parameter estimations for stochastic volatility model and discuss the limitations of the method, which is due to poor approximation to chi-square distribution. In the third section, we introduce Markov-Chain Monte-Carlo from a very basic level, including the method of Gibbs Sampling, Metropolis-Hasting sampling and rejection sampling. The proof of convergence can be found in the appendix and also a lot of examples from a pedagogical level. We reviewed paper written by Chib, Kim, and, Shepard [17] and derived the conditional posterior for the states and all three different parameters that can be used to perform to Single step Gibbs sampling. We performed simulation and discussed the issue of slow convergence in large ϕ case. In the fourth section, we introduced particle filters, including SIR and Auxiliary particle filtering and with its algorithm that can be applied to stochastic volatility model. At the end, we reviewed Pitt's paper[27] and show a proof why auxiliary particle filter is optimal and provide an unbiased estimator for true likelihood in the large particle number limit. Due to limited amount of time, the parameter estimation is not performed but the theory and method is completed, which help our readers to implement.

2 Time series, State Space Model and Stochastic Volatility Model

2.1 Time series

Time series is a common analysis method that develops mathematical models to explain observations. To capture the statistical character of random fluctuations in the data, *time series* is commonly defined as a collection of stochastic variables, which indexed in order according to the time they have been obtained. In general, a time series $\{y_t, t = 1, \cdots, T\}$ can also be referred to *stochastic process*. The index t is typically a time-like discrete variable, which can vary over integers $t = 0, \pm 1, \pm 2, \cdots \pm T$. All of the observed values $\{x_t, t = 1, \cdots, T\}$ corresponding to a stochastic process is defined a *realization*. From the characters of the randomness in stochastic processes, the realizations generated from a stochastic process may not be *identical* and a *time series* discussed in this project is interpreted as a particular realization from a stochastic process.

2.2 State Space Model

The statistical behaviour from a time series are often explained as aggregations of many information, therefore, it is difficult to develop mathematical models to describe the system evolution from the observations directly. The state space model is one of the standard approaches to provide solutions to these problems. In the state space model, the observations $\mathbf{y_t}$ are assumed to be explained by some vectors of state variables(sometimes also known as latent variable), $\mathbf{x_t}$. To illustrate how it can be applied in a real world problem, we gives a very simple model to explain GDP growth [5],

$$\begin{aligned} y_t &= \mu_t + \epsilon_t \\ \mu_t &= \mu_{t-1} + \eta_t, \end{aligned} \tag{1}$$

where y_t is the observed GDP growth from the data, μ_t is the latent slow-moving trend(state variable), which can be used to forecast the future GDP growth. Furthermore, the measurement noise, ϵ_t, and the process noise η_t are scaled i.i.d random variables normally distributed around zero mean and unit variance.

There are two fundamental assumptions in the state-space models.In state space models, if a state process x_t exists, then it is assumed to be Markovian, which means the past and the future only independent conditional on the current x_t. The second assumption is that the observations, y_t are independent to each other observations, given the states x_t. This governs the observations y_t are generated only by the states variables.

There are many state-space models available for different types of system or different type of noise in the literature, including non-linear non-Gaussian state space model[6]. In this project, we primary focused on the Gaussian linear state-space model[1], since it is one of foundations to understand the estimation and filtering technique in this project. In this section, we will introduce the notation and general form of the state space model, with its representation for AR(1), which is the pre-requisite to understand how Kalman filter can be applied to Stochastic Volatility Model. To start with, we denotes $N \times 1$ vector of observed variable at time t as $\mathbf{y_t}$, and the corresponding state vectors $\boldsymbol{\alpha_t}$, via a *measurement equation*, as following:

$$\boldsymbol{y_t} = \mathbf{Z_t}\alpha_\mathbf{t} + \boldsymbol{d_t} + \boldsymbol{\epsilon_t}, \tag{2}$$

where $\mathbf{Z_t}$ is an $N \times m$ matrix which describes the relations between state variables and observations, $\mathbf{d_t}$ is an $N \times 1$ vector and $\epsilon_\mathbf{t}$ is an $N \times 1$ vector of serially uncorrelated(in time) random noise with zero mean and covariance of $\mathbf{H_t}$, i.e.:

$$E(\boldsymbol{\epsilon_t}) = \underline{\mathbf{0}}, \qquad \text{and} \qquad \text{Var}(\boldsymbol{\epsilon_t}) = \mathbf{H_t}. \tag{3}$$

In general, the elements of state-variable is not observable, but in the state space model, we assume that the state process is generated by a first-order Markov process with the general form of:

$$\boldsymbol{\alpha_t} = \boldsymbol{T_t\alpha_{t-1}} + \boldsymbol{c_t} + \boldsymbol{R_t\eta_t}, \tag{4}$$

where $\mathbf{T_t}$ is a $m \times m$ matrix, $\mathbf{c_t}$ is an $m \times 1$ vector, $\mathbf{R_t}$ is a $m \times g$ matrix and $\boldsymbol{\eta_t}$ is a $g \times 1$ uncorrelated vector (in time) random noise with zero mean and covariance of $\mathbf{Q_t}$:

$$E(\boldsymbol{\eta_t}) = \underline{\mathbf{0}}, \qquad \text{and} \qquad \text{Var}(\boldsymbol{\eta_t}) = \mathbf{Q_t}. \tag{5}$$

Equation (4) is called *transition equation.* To specify the state-space system, we need to understand the initial conditions and ensure that $\boldsymbol{\eta}_t$ and $\boldsymbol{\epsilon_t}$ are uncorrelated. If we denote the initial state vector, $\boldsymbol{\alpha_0}$ and mean and covariance is given as, i.e.

$$E(\boldsymbol{\alpha_0}) = \mathbf{a_0}, \qquad \text{and} \qquad \text{Var}(\boldsymbol{\alpha_0}) = \mathbf{P_0} \tag{6}$$

Furthermore,since the random noises is uncorrelated from the initial state to all the moment, we have,

$$E(\boldsymbol{\epsilon_t\eta_s'}) = \underline{\mathbf{0}}, \quad \forall s, t = 1, \cdots, T. \tag{7}$$

and

$$E(\boldsymbol{\epsilon_t\alpha_0^\mathsf{T}}) = \underline{\mathbf{0}}, \qquad E(\boldsymbol{\eta_t\alpha_0^\mathsf{T}}) = \underline{\mathbf{0}}, \quad \forall t = 1, \cdots, T \tag{8}$$

In this project, we assume the the system matrices $\boldsymbol{Z, d, H, T, c, R, Q}$ are non-stochastic(deterministic). As a result, the system is linear and for any time t, observations $\mathbf{y_t}$ can be expressed as a linear combination of the present and past $\boldsymbol{\epsilon}$ and $\boldsymbol{\eta}$ and initial states $\boldsymbol{\alpha_0}$. If system matrices are time independent, the model is said time invariant, for example, the state-form of AR(1) models is a time invariant state-space model. In this case, we have $\mathbf{Z} = 1, \mathbf{d} = 0, \mathbf{T} = \phi, \mathbf{c} = (1 - \phi)\mu$:

$$\begin{aligned} y_t &= \alpha_t + \epsilon_t, \qquad \text{Var}(\epsilon_t) = \sigma_\epsilon^2 \\ \mu_t &= \phi\alpha_{t-1} + (1 - \phi)\mu + \eta_t, \quad \text{Var}(\eta_t) = \sigma_\eta^2 \end{aligned} \tag{9}$$

[1] State space model apply to linear system and Gaussian noise and the state was assumed to evolve as a linear system

The procedure is highly flexible and can be also applied for higher order AR model or ARMA models, which is not listed in this project but examples can be found in Harvey's book [13]. The aim of the state-space formulation is to design a system of a $\boldsymbol{\alpha}_t$, which will contain all the relevant information on the system with the minimum number of elements. Therefore, the detailed model selection for state space model may be another research topics and many common techniques including to tackle the model selection problems, including MLE and BIC(More details can be found at [15]). In this paper, we assume the state-space is simply AR(1) for the propose of presenting the techniques of filtering and estimation.

2.3 Stochastic Volatility Model

The volatility modelling is a crucial topics in time series analysis. In the classic Black-Scholes option pricing theory, the variance of an asset was assumed to be constant[8], however, many researches suggest the assumption is not valid in many financial data. One of the evidence is the volatility clustering phenomenon observed in time series, which is the high serial correlation in variance of return [14]. There are two well-researched models to capture the volatility clustering behaviours in the past two decades, which are stochastic volatility model and the Autoregressive Conditional Heteroscedasticity model. There are many criticism about ARCH type model from the literature, Matei [7] suggested that the ARCH model responds slowly to outliers and therefore it overpredicts the volatility and Nelson[9] noted that the oscillatory behaviour of the conditional variance can no-longer been observed. Stochastic volatility models is often considered as an alternative model that overcomes drawbacks of ARCH model due to Taylor[10]. The stochastic volatility model assumes that variance of return is an latent AR(1) process, as the following(also see Figure 1):

$$\begin{aligned} y_t &= \exp\left(\frac{x_t}{2}\right)\epsilon_t, \\ x_t &= \mu(1-\phi) + \phi x_{t-1} + \eta_t. \end{aligned} \tag{10}$$

where $\epsilon_t \sim \mathcal{N}(0,1)$ and $\eta_t \sim \mathcal{N}(0,\sigma_\eta^2)$, y_t is the observations from the financial data and x_t is the latent variable. The parameter ϕ is known as persistence of the volatility, in the real financial data, the persistence in daily return data, it is high and close to 1. The stochastic volatility model exhibit greater kurtosis than normal distribution(which means have a fatter tail than normal). This implies that many standard deviation in return is more likely to be observed than normal, which provide a better explanation instead of rare-event in normal distribution. The kurtosis can be analytically derived via the method of moment and moment generating function. To begin with, we need to restrict that the persistence parameter $|\phi| < 1$, which will ensure the AR(1) process in Equation(10) is stationary. In the state space model frame work, the observations between y_t is assumed to be independent to each other, the assumption is also valid in stochastic volatility model, as:

$$E(y_t y_{t-\tau}) = E(\epsilon_t \epsilon_{t-\tau})E\left[\exp\left(\frac{x_t + x_{t-\tau}}{2}\right)\right] = 0, \ \forall \tau \neq 0 \tag{11}$$

Despite for independence between observations y_t, however, if we take $\log y_t^2$ transform, the correlation appears as $\rho(h; \ln y_t^2) = \frac{\phi^h \sigma_x^2}{4.93+\sigma_x^2}, \forall h > 0$, where h is the time lag between the observations(see full derivation at appendix). The intuitive reason behind it is after transformations that we have $\ln y_t^2 = x_t + \ln \epsilon_t^2$, where x_t is a AR(1) process and it will depend on previous x_{t-1}, therefore, it will depend on $\ln y_{t-1}^2$and the correlation appears(see more details at appendix).

Furthermore, since it involves covariance of Gaussian white noise, it can guaranteed that all odd moments, i.e. $E(y_t^{2n+1}) = 0, \forall n \geq 0$. The variance can be computed via $\text{Var}(y_t) = E(\epsilon^2)E_\eta[\exp(x_t)] = \exp(\mu_x + 1/2\sigma_x^2)$. Since η is a Gaussian white noise, therefore, it is equivalent to compute moment generation function for normal distribution. We can use $E_\eta[\exp(jx_t)] = \exp(j\mu_x + 1/2j^2\sigma_x^2)$. Therefore, the fourth moment can be computed as(where we use the fourth moment of standard normal is 3):

$$E(y_t^4) = E(\epsilon^4)E_\eta[\exp(2x_t)] = 3\exp(2\mu_x + 2\sigma_x^2) \tag{12}$$

Then the kurtosis can be equivalent computed as:

$$k = \frac{E[(y_t - E(y_t)^4]}{\sigma^4} = E(y_t^4)/\sigma^4 = 3\exp(\sigma_x^2). \tag{13}$$

. The variance is also greater than zero, therefore, kurtosis is always greater than 3, which implies have fatter tail than Gaussian.

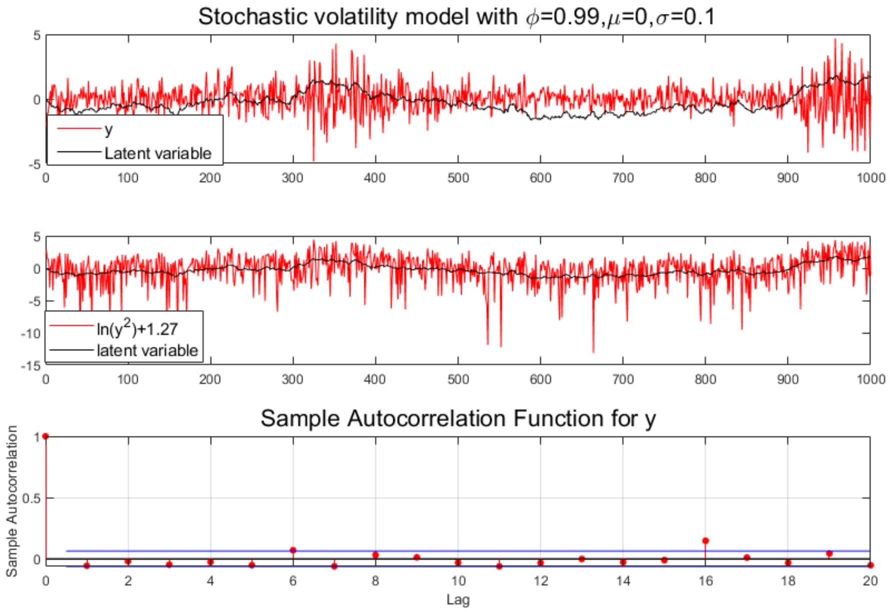

Figure 1: The upper graph is the stochastic volatility generated from the solutions of Ornstein–Uhlenbeck process with parameters of $\phi = 0.99, \mu = 0$ and $\sigma = 0.1$ with steps of $N = 1000$. The observations of y_t are dependent on the latent process x_t. The latent process can be interpreted as a reflection of the volatility at the observations, while x is more negative, it can be observed the volatility significantly decreased(see time between $600 - 700$), for more positive x, y fluctuated hugely(see time $300 - 400$), which match the intuition from the Equation (10)(the volatility exhibit in y is an exponential function of latent variable x). The middle graph illustrate how the transformation mentioned in Equation (39) performs, where transformed latent variable should lie at the mean displacement of y. The lowest graph is the sample autocorrelation function which help to illustrate observations $y_t, \forall t = 1, \cdots, 1000$ from the stochastic volatility model are independent to each other.

3 Kalman Filter

Kalman Filter provides a recursion solution to discrete-data linear filtering problems, which are originally invented by Rudolf E.Kálmán at 1960[11]. Kalman filter is still one of the most common and popular filters that provides a computational efficient means for estimating the state of a process, via minimizing the mean of the squared error. Kalman filter has been successfully applied in many different fields, especially in autonomous or assisted navigation[12]. In this section, the recursion equations for Kalman filter will be derived with the knowledge from basic vector algebra. We will also illustrate a method to approximate the stochastic volatility model to Gaussian-Linear model and also how the Kalman filter can be used to construct the likelihood function. Furthermore, we will perform Quasi-maximum likelihood estimation method to estimate the system parameters and discuss the weakness of the approximation with detailed examples, also in which limit, the filter perform worse. Furthermore, Kalman Smoother,a method to predict missing observations and its derivation was explained in a great detail in appendix.

3.1 Kalman Filter and its derivation

The notations and derivations in this section is adapted from A.C. Harvey's, 1983[13]. We use $\mathbf{a_t}$ to denote the optimal estimator of the state vector, $\boldsymbol{\alpha_t}$ based on all observations up to y_t. The prediction error at time t is denoted by a $m \times m$ co-variance matrix $\mathbf{P_t}$, which can be computed:

$$\mathbf{P_t} = E\left[(\boldsymbol{\alpha_t} - \mathbf{a_t})(\boldsymbol{\alpha_t} - \mathbf{a_t})'\right]. \tag{14}$$

Suppose at time $t-1$, the state and prediction error $\mathbf{a_{t-1}}$ and $\mathbf{P_{t-1}}$ can be computed, from *prediction equations* as follow, the optimal estimator of $\boldsymbol{\alpha_t}$ can be computed via:

$$\mathbf{a}_{t|t-1} = \mathbf{T_t}\mathbf{a_{t-1}} + \mathbf{c_t}, \tag{15}$$

and the optimal estimator of $\boldsymbol{P}_t$ as:

$$\mathbf{P}_{t|t-1} = \mathbf{T_t P_{t-1} T'} + \mathbf{R_t Q_t R'_t}, \quad \forall\ 1 \leq t \leq T, \tag{16}$$

while the corresponding estimator for $\mathbf{y_t}$ is:

$$\tilde{\mathbf{y}} = \mathbf{Z_t a_{t|t-1}} + \mathbf{d_t}. \tag{17}$$

The innovations vectors of prediction $\mathbf{v}$ at time t is denoted by,

$$\mathbf{v_t} = \mathbf{y_t} - \tilde{\mathbf{y}}_\mathbf{t} = \mathbf{Z_t}(\boldsymbol{\alpha_t} - \mathbf{a_{t|t-1}}) + \boldsymbol{\epsilon_t}, \tag{18}$$

and mean square errors $\mathbf{F_t}$ as:

$$\mathbf{F_t} = \mathbf{Z_t P_{t|t-1} Z'_t} + \mathbf{H_t}. \tag{19}$$

The estimator of the state will update once new information $\mathbf{y_t}$ is available. The updating equation are therefore can be written as the following:

$$\mathbf{a_t} = \mathbf{a_{t|t-1}} + \mathbf{P_{t|t-1} Z'_t F_t^{-1}}(\mathbf{y_t} - \mathbf{Z_t a_{t|t-1}} - \mathbf{d_t}), \tag{20}$$

and

$$\mathbf{P_t} = \mathbf{P_{t|t-1}} - \mathbf{P_{t|t-1} Z'_t F_t^{-1} Z_t P_{t|t-1}}. \tag{21}$$

From Equation 20, it implies that the more of the predicted for observation deviates from its realistic value, the bigger change it will be made to the estimator of the states. From Equation 15-16, 20-21, we can set up Kalman filter, given the initial conditions $\mathbf{a_0}, \mathbf{P_0}$, and then iterate through Kalman filter and produce an optimal estimator of the state of new observation. Once all T observations are processed, the estimator $\mathbf{a_T}$ contains all information.

3.1.1 Derivation of Kalman filtering

The prediction equation is derived from taking expectation of the transition equations (Equation 4) and use the fact that $E(\eta_t) = 0$, the optimal estimator for the state $\boldsymbol{\alpha_t}$ can be derived as:

$$\mathbf{a_{t|t-1}} = E(\mathbf{a_t}|\mathbf{y_t}) = \mathbf{T_t a_{t-1}} + \mathbf{c_t}. \tag{22}$$

From the error matrix definition(14), the error matrix $\mathbf{P_{t|t-1}}$ for the optimal estimator of the state variable $\mathbf{a_t}$ can be computed as:

$$\begin{aligned}
\mathbf{P_{t|t-1}} &= E\left[(\boldsymbol{\alpha_t} - \mathbf{a_{t|t-1}})(\boldsymbol{\alpha_t} - \mathbf{a_{t|t-1}})'\right] \\
&= E\left[(\mathbf{T_t}(\boldsymbol{\alpha_{t-1}} + \mathbf{c_t} + \mathbf{R_t}\boldsymbol{\eta_t} - \mathbf{T_t a_{t-1}} - \mathbf{c_t}))\,(\mathbf{T_t}(\boldsymbol{\alpha_{t-1}} + \mathbf{c_t} + \mathbf{R_t}\boldsymbol{\eta_t} - \mathbf{T_t a_{t-1}} - \mathbf{c_t}))'\right] \\
&= E\left[(\mathbf{T_t}\boldsymbol{\alpha_{t-1}} - \mathbf{T_t a_{t-1}}) + \mathbf{R_t}\boldsymbol{\eta_t})\,(\mathbf{T_t}\boldsymbol{\alpha_{t-1}} - \mathbf{T_t a_{t-1}}) + \mathbf{R_t}\boldsymbol{\eta_t})'\right] \\
&= \mathbf{T_t P_{t-1} T'_t} + \mathbf{R_t Q_t R'_t}.
\end{aligned} \tag{23}$$

With similar reasoning, mean square errors $\mathbf{F_t}$ for prediction $\tilde{y}_t$ can be written as:

$$F_t = E[(\mathbf{y_t} - \tilde{\mathbf{y}}_\mathbf{t})(\mathbf{y_t} - \tilde{\mathbf{y}}_\mathbf{t})] = \mathbf{Z_t P_{t|t-1} Z'_t} + \mathbf{H_t}. \tag{24}$$

Furthermore, in order to obtain the updating Equation (20), the co-variance between $\boldsymbol{\alpha_t}$ and $\boldsymbol{\epsilon_t}$ was computed, (the first line is about the linear shift of variable will not change the co-variance), i.e

$$\begin{aligned}
\mathrm{Cov}(\boldsymbol{\alpha_t}, \mathbf{v_t}|\mathbf{y_{1:t-1}}) &= \mathrm{Cov}(\boldsymbol{\alpha_t} - \mathbf{a_{t|t-1}}, \mathbf{y_t} - \mathbf{Z_t a_{t|t-1}} - \mathbf{d_t}|\mathbf{y_{1:t-1}}) \\
&= \mathrm{Cov}(\boldsymbol{\alpha_t} - \mathbf{a_{t|t-1}}, \mathbf{Z_t}\boldsymbol{\alpha}_\mathbf{t} - \mathbf{Z_t a_{t|t-1}} + \boldsymbol{\epsilon}_\mathbf{t}|\mathbf{y_{1:t-1}}) \\
&= \mathbf{P_{t|t-1} Z'}.
\end{aligned} \tag{25}$$

Therefore, $\mathbf{v_t}, \boldsymbol{\alpha_t}$ will be normally distributed accordingly,

$$\begin{pmatrix} \boldsymbol{\alpha_t} \\ \mathbf{v_t} \end{pmatrix} |\mathbf{y_{1:t-1}} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{a_{t|t-1}} \\ \mathbf{0} \end{pmatrix}, \begin{pmatrix} \mathbf{P_{t|t-1}} & \mathbf{P_{t|t-1}\ Z'} \\ \mathbf{Z\ P_{t|t-1}} & \mathbf{F_t} \end{pmatrix}\right). \tag{26}$$

Then the optimal state estimator $\mathbf{a_t}$ of the state variable $\boldsymbol{\alpha_t}$ which cooperates with includes the observation $\mathbf{y_t}$ can be calculated from computing the conditional expectation and use the result from inverse block matrix, which is:

$$\mathbf{a(t)} = E(\mathbf{a_t}|\mathbf{y_{1:t-1}}, \mathbf{v_t}) = \mathbf{a_{t|t-1}} + \mathbf{P_{t|t-1} Z'_t F_t^{-1} v_t} = \mathbf{a_{t|t-1}} + \mathbf{P_{t|t-1} Z'_t F_t^{-1}}(\mathbf{y_t} - \mathbf{Z_t a_{t|t-1}} - \mathbf{d_t}). \tag{27}$$

The updated optimal estimator prediction error $\mathbf{P_t}$ can be obtained using the inverse block matrix of the first term, i.e.

$$\mathbf{P_t} = \mathrm{Var}(\mathbf{a_{t|t}}, |\mathbf{y_{1:t-1}}, \mathbf{v_t}) = \mathbf{P_{t|t-1}} - \mathbf{P_{t|t-1} Z'_t F_t^{-1} Z_t P_{t|t-1}}. \tag{28}$$

as required, which can be updated recursively.

3.2 Parameter Estimation- Maximum Likelihood and Quasi MLE

State space model will have a system matrices and the parameters in the matrices in many case are unknown. We can denote these parameter by $n \times 1$ vector by $\boldsymbol{\psi}$, which is also know as *hyper-parameters*. Maximum likelihood estimation of the hyper-parameters is a method that determine the most probable hyper-parameters via maximising the likelihood given by the Kalman filter numerically.

The joint density of a set of T observations can be denoted in terms of conditional distribution, for a multivariate model, as

$$\mathcal{L}(\mathbf{y}; \boldsymbol{\psi}) = \prod_{t=1}^{T} p(\mathbf{y_t} | \mathbf{Y_{t-1}}), \tag{29}$$

where $p(\mathbf{y_t}|\mathbf{Y_{t-1}})$ denotes the distribution of y_t, which is conditioned on the information given before t, where $\mathbf{Y_{t-1}} = \{\mathbf{y_1}, \mathbf{y_2} \cdots, \mathbf{y_{t-1}}\}$. From the Kalman filter derivation, the state which is conditioned on Y_{t-1}, $\boldsymbol{\alpha_t} \sim \mathcal{N}(\mathbf{a_{t|t-1}}, \mathbf{P_{t|t-1}})$. Therefore, the state variable can be combined with the updating equation and the likelihood function can therefore be derived as:

$$\ln \mathcal{L}(\boldsymbol{\psi}) = -\frac{NT}{2} \ln 2\pi - \frac{1}{2} \sum_{t=1}^{T} \ln |\mathbf{F_t}| - \frac{1}{2} \sum_{t=1}^{T} \mathbf{v_t' F_t^{-1} v_t}. \tag{30}$$

Just reminding, v_t is a vector of prediction error $\mathbf{v_t} = \mathbf{y_t} - \mathbf{\tilde{y}_t}$.

Prior to perform MLE, we applies some transformations to y so we have $\ln y_t^2$, which is :

$$\ln y_t^2 = x_t + \ln \epsilon_t^2 \tag{31}$$

In this case, the error term $\ln \epsilon_t^2$ is clearly non-Gaussian, though ϵ_t follows normal distribution. Instead of normally distributed, ϵ^2 follows a chi-squared distribution with 1 degree-of-freedom, $\epsilon^2 \sim \chi^2_{(1)}$. Therefore, the true likelihood is no-longer Gaussian, the true likelihood function can no longer being constructed via the above framework. Instead of using the true likelihood, *Quasi-MLE* is a method commonly used that approximates non-Gaussian residual as Gaussian. In this case, Quasi-MLE method approximates $\ln \epsilon^2$ with a normal variable, whose mean and variance are -1.2703 and 4.93 respectively as,

$$\ln y_t^2 = -1.27 + x_t + \xi_t \tag{32}$$

where $\xi_t = \ln \epsilon_t^2 + 1.27$ and is a i.i.d with zero mean and variance of 4.93. Now the above likelihood equation 37 for Kalman filter can be used, simply insert the equation into the state space form, i.e.

$$\begin{aligned} \ln y_t^2 &= \underbrace{1}_{\text{Z}} x_t + \underbrace{-1.27}_{d} + \xi_t \\ x_t &= \underbrace{\phi}_{\text{T}} x_{t-1} + \underbrace{\mu(1-\phi)}_{\text{c}} + \eta_t \end{aligned} \tag{33}$$

The parameters of the Kalman filter can be substituted into the state space form:

$$\begin{aligned} a_{t|t-1} &= \phi a_{t-1} + \mu(1-\phi), & P_{t|t-1} &= \phi^2 P_{t-1} + \sigma_\eta^2 \\ \tilde{y}_t &= a_{t|t-1} - 1.2703, & v_t &= \ln y_t^2 - \tilde{y}_t = y_t - a_{t|t-1} + 1.2703 \\ F_t &= \phi^2 P_{t-1} + \sigma_\eta^2 + \sigma_\epsilon^2, & & \end{aligned} \tag{34}$$

The updated optimal estimator prediction error $\mathbf{P_t}$ can be obtained using the inverse block matrix of the first term, i.e.

$$\mathbf{P_t} = \text{Var}(\mathbf{a_{t|t}}, |\mathbf{y_{1:t-1}}, \mathbf{v_t}) = \mathbf{P_{t|t-1}} - \mathbf{P_{t|t-1} Z_t' F_t^{-1} Z_t P_{t|t-1}}. \tag{35}$$

as required, which can be updated recursively.

3.3 Parameter Estimation- Maximum Likelihood and Quasi MLE

State space model will have a system matrices and the parameters in the matrices in many case are unknown. We can denote these parameter by $n \times 1$ vector by $\boldsymbol{\psi}$, which is also know as *hyper-parameters*. Maximum likelihood estimation of the hyper-parameters is a method that determine the most probable hyper-parameters via maximising the likelihood given by the Kalman filter numerically.

The joint density of a set of T observations can be denoted in terms of conditional distribution, for a multivariate model, as

$$\mathcal{L}(\mathbf{y};\boldsymbol{\psi}) = \prod_{t=1}^{T} p(\mathbf{y_t}|\mathbf{Y_{t-1}}), \tag{36}$$

where $p(\mathbf{y_t}|\mathbf{Y_{t-1}})$ denotes the distribution of y_t, which is conditioned on the information given before t, where $\mathbf{Y_{t-1}} = \{\mathbf{y_1}, \mathbf{y_2} \cdots, \mathbf{y_{t-1}}\}$. From the Kalman filter derivation, the state which is conditioned on Y_{t-1}, $\boldsymbol{\alpha_t} \sim \mathcal{N}(\mathbf{a_{t|t-1}}, \mathbf{P_{t|t-1}})$. Therefore, the state variable can be combined with the updating equation and the likelihood function can therefore be derived as:

$$\ln \mathcal{L}(\boldsymbol{\psi}) = -\frac{NT}{2}\ln 2\pi - \frac{1}{2}\sum_{t=1}^{T} \ln|\mathbf{F_t}| - \frac{1}{2}\sum_{t=1}^{T} \mathbf{v_t'}\mathbf{F_t^{-1}}\mathbf{v_t}. \tag{37}$$

Just reminding, v_t is a vector of prediction error $\mathbf{v_t} = \mathbf{y_t} - \tilde{\mathbf{y_t}}$.

Prior to perform MLE, we applies some transformations to y so we have $\ln y_t^2$, which is :

$$\ln y_t^2 = x_t + \ln \epsilon_t^2 \tag{38}$$

In this case, the error term $\ln \epsilon_t^2$ is clearly non-Gaussian, though ϵ_t follows normal distribution. Instead of normally distributed, ϵ^2 follows a chi-squared distribution with 1 degree-of-freedom, $\epsilon^2 \sim \chi^2_{(1)}$. Therefore, the true likelihood is no-longer Gaussian, the true likelihood function can no longer being constructed via the above framework. Instead of using the true likelihood, *Quasi-MLE* is a method commonly used that approximates non-Gaussian residual as Gaussian. In this case, Quasi-MLE method approximates $\ln \epsilon^2$ with a normal variable, whose mean and variance are -1.2703 and 4.93 respectively as,

$$\ln y_t^2 = -1.27 + x_t + \xi_t \tag{39}$$

where $\xi_t = \ln \epsilon_t^2 + 1.27$ and is a i.i.d with zero mean and variance of 4.93. Now the above likelihood equation 37 for Kalman filter can be used, simply insert the equation into the state space form, i.e.

$$\begin{aligned} \ln y_t^2 &= \underbrace{1}_{\text{Z}} x_t + \underbrace{-1.27}_{d} + \xi_t \\ x_t &= \underbrace{\phi}_{\text{T}} x_{t-1} + \underbrace{\mu(1-\phi)}_{\text{c}} + \eta_t \end{aligned} \tag{40}$$

The parameters of the Kalman filter can be substituted into the state space form:

$$\begin{aligned} a_{t|t-1} &= \phi a_{t-1} + \mu(1-\phi), & P_{t|t-1} &= \phi^2 P_{t-1} + \sigma_\eta^2 \\ \tilde{y}_t &= a_{t|t-1} - 1.2703, & v_t &= \ln y_t^2 - \tilde{y}_t = y_t - a_{t|t-1} + 1.2703 \\ F_t &= \phi^2 P_{t-1} + \sigma_\eta^2 + \sigma_\epsilon^2, & & \end{aligned} \tag{41}$$

and the updating equation from Equation (27) and Equation (35) can be written as :

$$\begin{aligned} a_t &= \phi a_{t-1} + \mu(1-\phi) + \frac{(\phi^2 P_{t-1} + \sigma_\eta^2)(\ln y_t^2 - (\phi a_{t-1} + \mu(1-\phi)) + 1.27)}{\phi^2 P_{t-1} + \sigma_\eta^2 + \sigma_\epsilon^2} \\ P_t &= \phi^2 P_{t-1} + \sigma_\eta^2 - \frac{(\phi^2 P_{t-1} + \sigma_\eta^2)^2}{\phi^2 P_{t-1} + \sigma_\eta^2 + \sigma_\epsilon^2}. \end{aligned} \tag{42}$$

With treating ξ_t as a Gaussian variable, the optimal estimator for state $\boldsymbol{\alpha}_t$ is required to be used to compute a prediction, $\tilde{y}_t$. With the Gaussian noise assumption, the following log-likelihood from the Kalman Filter can be computed:

$$\begin{aligned} \ln p(y_t|Y_{1:t-1}) &= \text{cons} - \frac{1}{2}\ln F_t - \frac{1}{2}\frac{(\ln y_t^2 - \tilde{y}_t)^2}{F_t}, \\ \ln p(\alpha_t|Y_{1:t}) &= \text{cons} - \frac{1}{2}\ln P_t - \frac{1}{2}\frac{(\alpha - a_t)^2}{P_t}. \end{aligned} \tag{43}$$

Therefore, the likelihood function can be written as:

$$
\begin{aligned}
\ln \mathcal{L}(y_{1:T};\theta) &= \sum_{t=1}^{T} \ln p(y_t|Y_{1:t-1}) \\
&= \text{cons} - \frac{1}{2}\sum_{t=1}^{T} \ln F_t - \frac{1}{2}\sum_{t=1}^{T} \frac{(\ln y_t^2 - \tilde{y}_t)^2}{2F_t} \\
&= \text{cons} - \frac{1}{2}\sum_{t=1}^{T} \ln(\phi^2 P_{t-1} + \sigma_\eta^2 + \sigma_\epsilon^2) - \frac{1}{2}\sum_{t=1}^{T} \frac{\left[\ln y_t^2 - (\phi a_{t-1} + \mu(1-\phi)) + 1.2703\right]^2}{\phi^2 P_{t-1} + \sigma_\eta^2 + \sigma_\epsilon^2},
\end{aligned} \tag{44}
$$

which can be used to perform Quasi-MLE.

3.4 Performance of the estimator&filter to perform inference on artificial data

In this project, Kalman filter was applied to the artificial datasets for the purpose of checking the performance, since the true parameters and latent variable are already known. Firstly, we apply Kalman filter to stochastic volatility model 10000 realizations with 1000 time steps and compute the mean of the likelihood across all realizations, which will average out the randomness in each paths to see whether the mean maximum likelihood estimation, on average can provide a 'roughly' good estimate to different parameters. We firstly chose the persistence as $\phi = 0.99$, which is most closely financial data related in the real-world and set $\mu = 0.5$ with $\sigma = 0.1$. From Figure 2, it can clearly observed that the estimated result is peaked around the true parameters, but not identically distributed at the true parameter. Furthermore, one Kalman filtering for filtering the state example from the result has also been plotted(See Figure 3), as an illustration of how Kalman filter can provide a estimate of the latent variable x. Overall, from the above result, at a fast glance that Kalman filter provide a roughly good estimate.

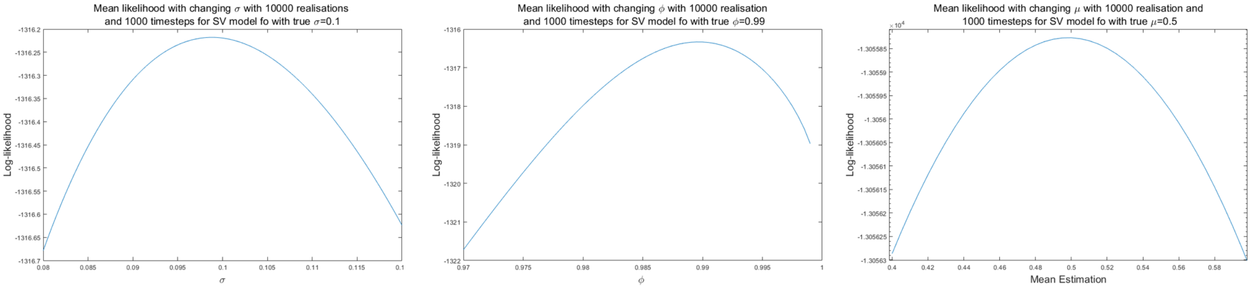

Figure 2: Mean likelihood corresponding to parameters respect to ϕ, σ, μ, with the data generated from the true parameter of $\phi = 0.99, \mu = 0.5, \sigma = 0.1$ with 10000 realizations and 1000 timesteps. From the graph we can see that the quasi-maximum likelihood estimation produce a fairly accurate result, but the still not exactly peaked at the true parameter(slightly biased)

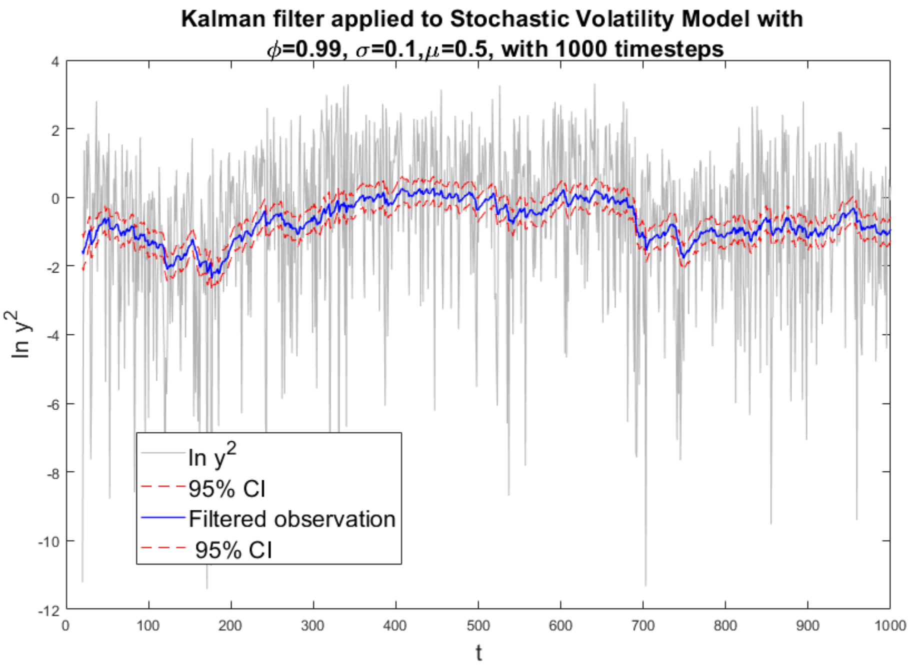

Figure 3: Kalman filter applied to a realization of the stochastic volatility model, with using the input of true parameter of $\phi = 0.99, \mu = 0.5, \sigma = 0.1$ with 1000 timesteps. The confidence interval is computed via using the prediction error P_t at time step and the blue line illustrated the filtered observations. The filter line smooths $\ln y_t^2$ and produce a similar trend.

Then Kalman filter was applied to 4000 realizations and 1000 times steps data from a stochastic volatility model with true parameters of $\psi = (\phi, \mu, \sigma) = (0.99, 0.5, 0.1)$. In each iteration, the parameters that optimize the likelihood estimation in Equation (44) was computed with constrained univariate search [2]. From the estimation result in Figure(4), it is fairly clear that the estimator is consistent for all parameters, which means the estimator for parameters $\hat{\psi}$ is converging to some value $\boldsymbol{\psi^0}$, and meanwhile, all parameters also asymptotically normal distributed, apart from for ϕ(see QQ plot in Figure 4). Furthermore, the performance of the approximation also depends on heavily the true σ_η(see Equation 40). For large values of σ_η^2, the latent process x_t will dominate the non-Gaussian error term ξ, which implies the QMLE will be more efficient. In order to demonstrates the size of σ_η will affect the quality of inference, three simulations were carried out for stochastic volatility models with different level of σ_η, compare to the level of measurement error ξ. To indicate the level of bias, we compare Root mean squared error with standard deviation of the predicted parameters across different realisations. In this case, we can compute bias square via subtracting standard deviation from root mean square[16] i.e.

$$E\left[(\hat{\theta} - \theta)^2\right] = \mathrm{Var}(\hat{\theta}) + \mathrm{Bias}_{\hat{\theta}}(\hat{\theta}, \theta)^2 \tag{45}$$

From Table3.4, it is obvious that the inference result for ϕ is slightly sub-optimal, regardless the level of signal noise changes. However, the bias level in σ_η and μ scaled with the ratio between σ_η and ξ. In the case $\sigma_\eta = 0.1$, the bias level in σ_η and μ are significantly higher than the rest of table. The noise ratio ratio between σ_η and ξ also affect the quality of the inference(See Figure 5). Kalman filters were applied to three data set generated from the stochastic volatility model, with the same sequence random number and system parameters but different σ_η. From the figure, it can be clearly observed the performance of the filtering is poor for low σ_η, it constantly fluctuates around the true latent variable, however, for large σ_η, the filtered result will more precisely predict the latent variable. The reason of being sub-optimal is due to the poor log-normal approximation to chi-square distribution(see Figure 6). In the financial industries, the process noise σ_η^2 is very possible to be small like 0.1, which leads to bad estimation for latent variable. In this sense, we have introduced our second second method, Monte-Carlo sampling.

[2] We use less realization due to time costing in optimizations, many literature's suggests Newton-Ralphson Optimization problem or alternatively use annealing algorithm but this report focus on the filtering and estimation problem due to time limitations.

Table 1: Summary of estimated parameters from Kalman Filter via MLE

8000 realisations and 1000 time steps Stochastic Volatility model with $\phi = 0.99, \sigma = 0.1, \mu = 0.5$				
Parameters	Mean from MLE	Standrad deviation	Stdandrad Error($\times 10^{-5}$)	Root Mean Squared Error
$\hat{\phi}$	0.9869	0.0074	8.2932	0.0080
$\hat{\mu}$	0.4302	0.2841	317.63	0.2926
$\hat{\sigma_\eta}$	0.0959	0.0230	25.714	0.0234

8000 realisations and 1000 time steps Stochastic Volatility model with $\phi = 0.95, \sigma = 0.5, \mu = 0.6$				
Parameters	Mean from MLE	Standrad deviation	Stdandrad Error($\times 10^{-5}$)	Root Mean Squared Error
$\hat{\phi}$	0.9472	0.0118	13.163	0.0121
$\hat{\mu}$	0.5817	0.3145	350.12	0.3150
$\hat{\sigma}_\eta$	0.4966	0.0514	57.514	0.0515

8000 realisations and 1000 time steps Stochastic Volatility model with $\phi = 0.99, \sigma = 1, \mu = 0.5$				
Parameters	Mean from MLE	Standard deviation	Standard Error($\times 10^{-5}$)	Root Mean Squared Error
$\hat{\phi}$	0.9871	0.0055	6.1490	0.0063
$\hat{\mu}$	0.4790	2.6707	2999.1	2.6707
$\hat{\sigma}_\eta$	0.9972	0.0726	81.690	0.0726

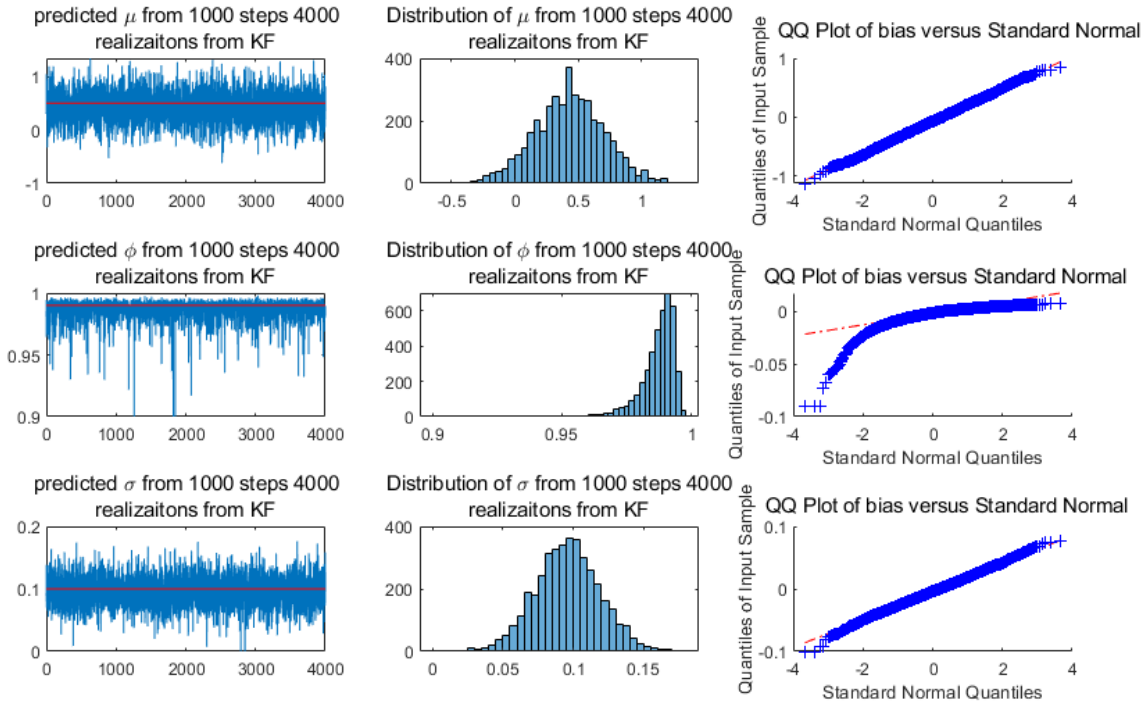

Figure 4: We generated 4000 realizations with 1000 timesteps data from stochastic volatility model with $\phi = 0.99, \sigma = 0.01, \mu = 0.5$. The first column of the graph describes the parameter estimated from the Kalman filter and the red line corresponding to true parameter, we can see that Kalman Filter provide a good estimation for mean, where $\hat{\mu} = 0.4308$, a fairly good estimate for the volatility with a slightly negative bias $\hat{\sigma} = 0.0959$ but some negative bias for persistence, $\hat{\phi} = 0.9768$. From the second column, we can see the parameter seeming asymptotically normally distributed apart from for the estimation towards to ϕ. The third column is qq plot, which measure whether the residual is normally distributed around its mean. If the most of the dots lie on the line, see the first and the third, then it suggest the residual(predicted parameters- mean of predictions) is normally distributed.

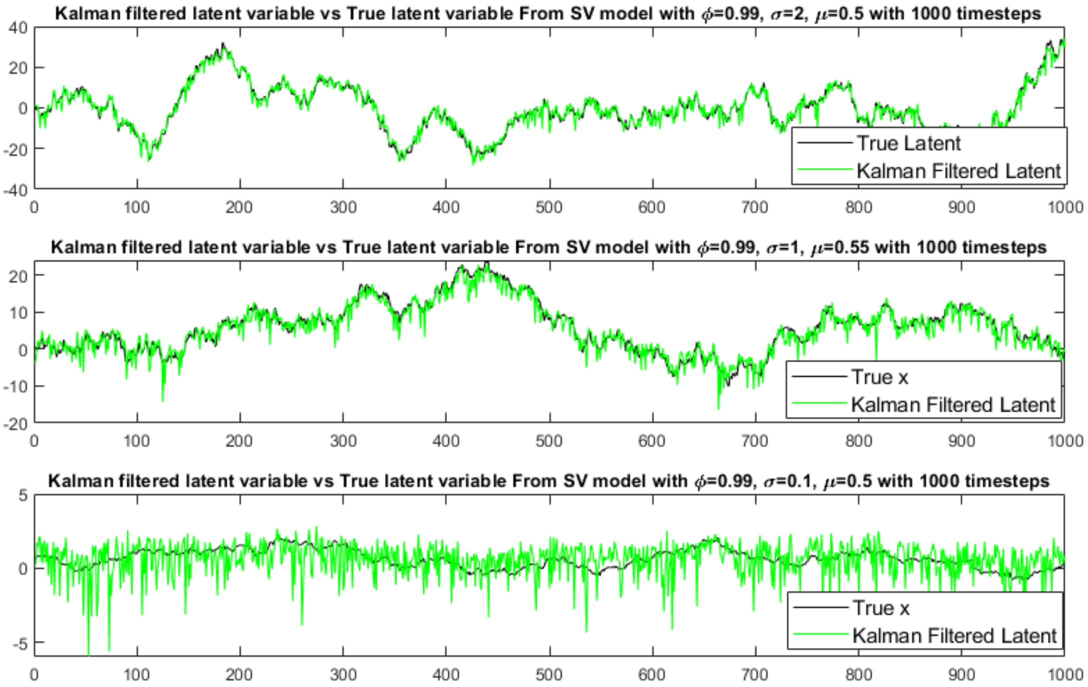

Figure 5: Despite Kalman Filter provide some good estimations of the parameters, however, the quality of the filtering depend on the signal to noise ratio. We have demonstrated this for three sets of data generated from the stochastic volatility model with the same parameters, but different level of the process noise ($\sigma_\eta = 0.1, 1, 2$). Clearly, the example illustrates the performance of Kalman filter will not be good in the case that the measurement signal is much greater than the process noise.

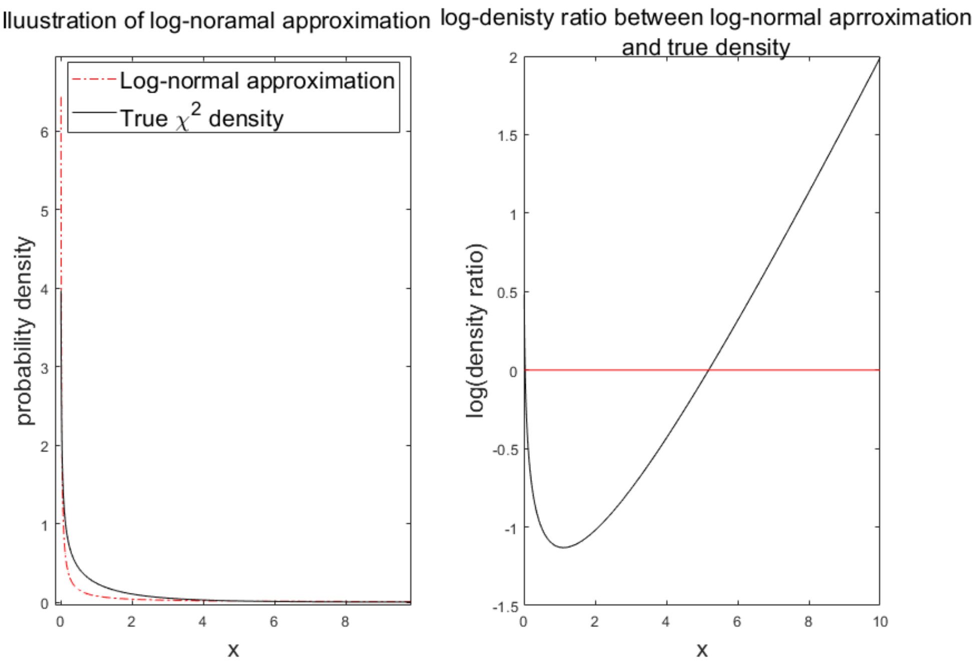

Figure 6: Illustration of the approximations of χ_1^2 density to log-normal density. The second graph corresponding to the density ratio, which indicates the performance of the approximation is worse, while we sampling random number from approximated distribution, larger random number is more likely to be obtained compared to the original distribution(higher density)

4 Markov Chain Monte-Carlo

An alternative method to inference in stochastic volatility model is based on Markov Chain Monte Carlo method. Monte-Carlo methods had a wide spread influence on the theory and practice of Bayesian inference[17]. The idea of Markov Chain Monte Carlo method is to produce a variate from a given multivariate density(the posterior density in Bayesian applications) by repeatedly sampling a Markov Chain whose invariant distribution is the target density of interest. The ways of constructing a Markov chain for stochastic volatility models with this property is not unique. In this project, the primary problem is the likelihood function $f(y|\theta) = \int f(y|h,\theta)f(h|\theta)\ dh$ is intractable. Therefore, the direct analysis of the posterior density $\pi(\theta|y)$ by MCMC method is no longer possible. The problem can be overcame by sampling from the latent density $\pi(\theta, h|y)$. The most common MCMC method is Gibbs sampler, which is an modified version of metropolis algorithm developed by Hastings in the statistical setting. The basic strategy is to use a conditional distribution to set up a Markov Chain to draw a sample at each time. The sample drawn from the previous was used to obtain the next sample, once the time is long enough, the the sample drawn will follow the true posterior, which achieved the goal of inference and filtering.

In this section, we will first introduced our readers about the foundations required to perform the single-step Gibbs sampling for stochastic volatility model, which includes the general framework for Gibbs sampling and the proof why it converges the stationary distribution(in the appendix), meanwhile, an example of how Gibbs Sampling to produce bi-variate distribution from Gibbs sampling was illustrated in the appendix. Later on, Metropolis-Hasting Algorithm and the derivation of detailed balance was also introduced (Example of how to metropolis-hasting sampling algorithm to generate mixture of Gaussian distribution were appended in the appendix) and also the rejection sampling with an empirical proof about why accept sample will have the same distribution of the target density. At the end, the conditional posterior for the state and parameters for stochastic volatility model required by performing the inference were derived, which need to be sampled via using the Gibbs sampling and Metropolis-Rejection algorithm.

4.1 Gibbs sampling

The goal of the Gibbs sampling is to generate the posterior samples by sweeping through variable to sample from its conditional distribution, while keeping all other variable fixed. A very detailed example about how to generates samples from bi-variate normal for building a good understanding Gibbs sampling can be found in the appendix. Here are the general algorithm: Instead of generating random variables from the posterior directly, Gibbs simulating

Algorithm 1 Gibbs sampling

```
1: Sample $\mathbf{x}^{(0)}$ from $\mathbf{x}^{(0)}$ from $q(\mathbf{x})$
2: for $i = 1, \cdots, N$ do
3:     Sample $x_1^{(i)} \sim p(x_1^{(i)}|x_2^{(i-1)}, x_3^{(i-1)}, x_4^{(i-1)}, \cdots, x_D^{(i-1)})$
4:     Sample $x_2^{(i)} \sim p(x_2^{(i)}|x_1^{(i)}, x_3^{(i-1)}, x_4^{(i-1)}, \cdots, x_D^{(i-1)})$
5:         $\cdots$
6:     Sample $x_D^{(i)} \sim p(x_D^{(i)}|x_1^{(i)}, x_2^{(i)}, x_3^{(i)}, x_4^{(i)}, \cdots, x_{D-1}^{(i)})$
7: end
8: return x
```

samples with sweeping through all posterior conditioned on given information, at each update, only one random variable was generated. Furthermore, we define a complete updates for all variables as a *sweep*(from line 3-6 in Algorithm 1). For the early stage, the sweeps will generate some transient samples, see Figure(12) in the appendix, as the most Monte-Carlo method do. However, after many sweeps, the Gibbs sampler will generate samples satisfies the stationary distribution, which has the same density as the target posterior(see the proof in the appendix).

4.2 Metropolis-Hasting Algorithm

One of the most popular MCMC algorithm is the Metropolis-Hasting Algorithm, which is firstly introduced at the second of world war used to work for Nuclear weapon. It is first published at 1953 and then generalised by Hastings in 1970[19]. The only requirement of Metropolis-Hasting algorithm is to evaluate a function which is proportional to the target density. The standard algorithm can be summarised below:

Algorithm 2 The Metropolis-Hastings Algorithm

```
Choose an initial state, $\boldsymbol{\theta}^{(0)}$
while $t < T$ do
    Draw a sample from the proposal distribution $q$, $\boldsymbol{\theta}^{(t-1)} \sim q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})$
    Compute the acceptance probability, $\alpha(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)}) = \min\left(1, \frac{q(\theta^{(t-1)}|\theta)\pi(\theta)}{q(\theta|\theta^{(t-1)})\pi(\theta^{(t-1)})}\right)$
    Accept the $\boldsymbol{\theta}$ with probability of $\alpha(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})$, $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}$, if not accepted, $\boldsymbol{\theta}^{(t)} = \boldsymbol{\theta}^{(t-1)}$
return $\boldsymbol{\theta}^{(1,\cdots,T)}$
```

In order to obtain samples from the stationary distribution of $\pi(\boldsymbol{\theta})$, the iterations need to be continued until it is stationary. There are some remarks about the algorithm: firstly, the proposed value of $\boldsymbol{\theta}$ can be rejected, therefore, θ may stay at the current value, which depends on the acceptance probability. Secondly, if the proposed distribution is symmetrical(i.e.$q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)}) = q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})$), then acceptance probability will simply be $\min\{\pi(\boldsymbol{\theta})/\pi(\boldsymbol{\theta}^{(t-1)}), 1\}$. For $\pi(\boldsymbol{\theta}) > \pi(\boldsymbol{\theta}^{(t-1)}$, then we accept $\boldsymbol{\theta}$ for sure, but there is still non-zero probability to move from higher π region to lower π, therefore, the sampler will not trapped at local maximum and can explore the whole space. This special case is the original algorithm that developed by Metropolis at 1953[20], which provides a basics for several algorithms to solve the optimisation problem, notably the method of annealing problem.

As it is a MCMC method, the draws are regarded as samples from the target density $\pi(\theta)$ only after the transient stage (the same as the Gibbs sampling), which means the effect from the initial condition is no-longer important. Furthermore, the condition of converging to unique stationary distribution is that the Markov-chain is regular, which means, aperiodic and irreducible[18]. These conditions are generally satisfied by that the positive transition rate q associate with positive $\pi(\cdot)$. Therefore, a question might be raise, how large is the sample we should take and how long we should run. One of the possible solution is that start multiple chain with different initial condition(for more detailed discussion see[21]). In this project, the convergence rate issue, especially at high ϕ case, which is also discussed via the using the analytically correlations between parameters.

4.3 Rejection sampling

Metropolis-Hastings algorithm may have problems with choosing an appropriate proposal distribution. Again sampling directly from the distribution using the inverse of cumulative distribution function is not always feasible,

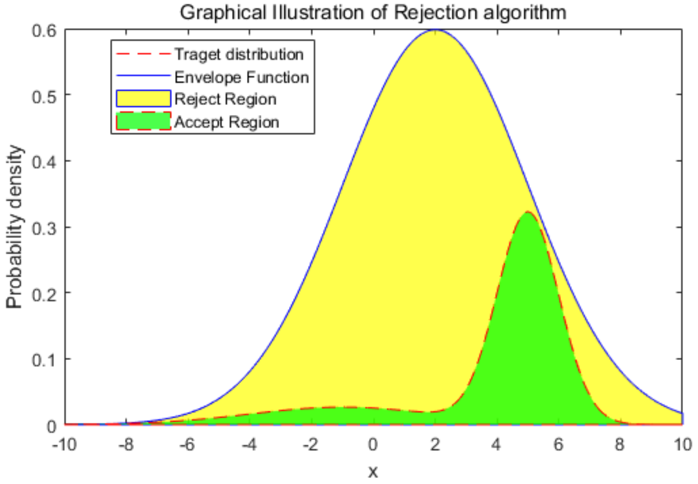

Figure 7: Rejection Sampling graphical illustration. We choose a mixture of Gaussian as the target density and normal density as proposed density. The yellow area represent the rejection region. If a random number generated from proposed distribution and then we can compute the acceptance probability and use uniform random number to reject or accept the sample. If the random number is smaller than the acceptance probability, then it is in green area, as accepted sample.

therefore, an alternative algorithm can be used to perform the sampling task, which is *rejection sampling*. Rejection sampling enable us to sample a complex distribution from a relative easier distribution that can may be directly sampled. The reason of why it is called rejection-sampling is that once samples are generated from the envelope functions, the sample points were either accept or reject, subject to the desired distribution. To understand how it works, we introduce some definitions in this paragraph. Firstly, the target density are denoted as $\pi(\theta)$ and the proposed density as density $g(\theta)$ and $cg(\theta)$ as envelope function, where c is a constant. Furthermore, it is required that the envelope function need to cover the whole target density so we can accept or reject(See an example at Figure 7). Most common criterion for accepting samples from the target density $\theta \sim \pi(\theta)$ is the efficiency of the sampling scheme, which is the ratio of the target distribution to that of the envelope distribution. The algorithm can be summarised as:

Algorithm 3 Rejection Sampling

```
Choose an appropriate envelope function, so that $Kg(\theta) \geq \pi(\theta)$ for all $\theta$
**while** $t < T$ **do**
    Draw a sample from the proposal distribution $g$, $\theta \sim g(\theta)$
    Accept the sample with probability $\frac{\pi(\theta)}{Kg(\theta)}$
**return** $\theta^{(1,\cdots,T)}$
```

4.3.1 Empirical Proof of Accepting-rejection sampling

The algorithm suggests that the distribution of the accepted sample $p(\theta|\mathbf{A})$, is equivalent to the target density$\pi(\theta)$, we can prove it simply with the aid of the Bayes Rule:

$$p(\theta|\mathrm{A}) = \frac{p(\mathrm{A}|\theta)g(\theta)}{p(\mathrm{A})} = \frac{\frac{\pi(\theta)}{Kg(\theta)}g(\theta)}{\int \frac{\pi(\theta^*)}{Kg(\theta^*)}g(\theta^*)d\theta^*} = \pi(\theta). \tag{46}$$

4.4 Bayesian Inference apply to Stochastic Volatility Model

In this context, performing estimation requires to maximise $p(\theta, x_{1:T}|y_{1:T})$, where y_t is the data and x_t is the state variable in the stochastic volatility model, intuitively, it can be interpreted as the given all the information and what is the most likely parameter will be. According to Bayes rule, the posterior density decomposed to:

$$p(\theta, x_{1:T}|y_{1:T}) = \frac{p(y_{1:T}|\theta, x_{1:T})p(x_{1:T}|\theta)p(\theta)}{p(y_{1:T})} \propto p(y_{1:T}|\theta, x_{1:T})p(x_{1:T}|\theta)p(\theta), \tag{47}$$

where the denominator is simply marginal density over all parameter space θ and all state space $x_{1:T}$, i.e. $p(y_{1:T}) = \int p(y_{1:T}|\theta, x_{1:T})p(x_{1:T}|\theta)p(\theta)d\theta dx_{1:T}$, therefore, it is a constant and will only linearly scale the posterior density. For this reason, maximising the posterior is equivalent to maximise the numerator. Furthermore, $p(\theta)$ is the joint distribution of the parameters, which incorporates with prior knowledge about these parameters(which is generally known as *prior*). The second term $p(x_{1:T}|\theta)$ is the conditional probability of the state variable if the parameter θ is given and the first term $p(y_{1:T}|\theta, x_{1:T})$ is the likelihood function of the data given by the model. In the practical case, the data dimension is high so that sampling directly from the posterior density is not possible. To overcome the problem, Gibbs sampling method from Kim and Shepard (1998)[17] was used to use. As discussing above, Gibbs sampling update one parameter at each steps and the Gibbs sampling can be adapted to stochastic volatility model as below:

Algorithm 4 Gibbs Sampling Algorithm for Stochastic Volatility model

```
Initialise the variable, $\mu^{(0)}, \sigma^{2(0)}, x^{(0)}, \phi^{(0)}, \beta^{(0)}$
for $i = 1, \cdots, n$, do
    for $t = 1, \cdots, T$ do
        Draw $x_t^{(i)}$ from $p(x_t^{(i)}|x_{<t}^{(i)}, x_{>t}^{(i-1)}, \phi^{(i-1)}, \mu^{(i-1)}, \sigma_\eta^{2(i-1)}, y, \beta)$
    end
    Draw $\sigma_\eta^{2(i)}$ from $p(\sigma_\eta^{2(i)}|x^{(i)}, \phi^{(i-1)}, \mu^{(i-1)}, y, \beta, \sigma_\eta^{2(i-1)})$
    Draw $\phi^{(i)}$ from $p(\phi^{(i)}|x^{(i)}, \phi^{(i-1)}, \mu^{(i-1)}, y, \beta, \sigma_\eta^{2(i)})$
    Draw $\mu^{(i)}$ from $p(\mu^{(i)}|x^{(i)}, \phi^{(i)}, \mu^{(i-1)}, y, \beta, \sigma_\eta^{2(i)})$.
end
Compute the distribution for $\mu, \sigma^2, x_{1:T}, \phi, \beta$.
```

where $x_{<t} = x_1, \cdots x_{t-1}$ are all moments less than t and similar for $x_{>t}$. In the following section we will explain with more details about how these sampling methods work.

4.4.1 Rejection Sampling for the state using the conditional distribution

Since x is a AR(1) model in the stochastic volatility model and according to the assumptions in the state space model, the process x_t is a Markovian, therefore, x_t depends on x_{t-1} and x_{t+1}. With the help of the Bayes rule, we can denote:

$$\begin{aligned} p(x_t|x_{<t}, x_{>t}, \phi, \mu, \sigma_\eta^2, y, \beta) = p(x_t|x_{t-1}, x_{t+1}, \phi, \mu, \sigma_\eta^2, y, \beta) &\propto p(x_t|x_{t-1}, x_{t+1}, \phi, \mu, \sigma_\eta^2)p(y_t|x_t, \beta) \\ &\propto p(x_{t+1}|x_t, \phi, \mu, \sigma_\eta^2)p(x_t|x_{t-1}, \phi, \mu, \sigma_\eta^2)p(y_t|x_t, \beta). \end{aligned} \tag{48}$$

Since x is an AR(1) process, therefore, $p(x_{t+1}|x_t, \phi, \mu, \sigma_\eta^2) = \mathcal{N}(x_{t+1}|\phi x_t + (1-\phi)\mu, \sigma_\eta^2)$ and similarly for $p(x_{t-1}|x_t, \phi, \mu, \sigma_\eta^2)$. The product of two Gaussian densities is another Gaussian and it can be shown with completing the square as,

$$\begin{aligned} p(x_t|x_{t-1}, x_{t+1}, \phi, \mu, \sigma_\eta^2) &= p(x_{t+1}|x_t, \phi, \mu, \sigma_\eta^2)p(x_t|x_{t-1}, \phi, \mu, \sigma_\eta^2) \\ &= \mathcal{N}(x_t|\mu + \frac{\phi[(x_{t-1} - \mu) + (x_{t+1} - \mu)]}{1 + \phi^2}, \frac{\sigma_\eta^2}{1 + \phi^2}). \end{aligned} \tag{49}$$

Meanwhile, to derive density for $p(y_t|x_t, \beta)$, we need to transform the variable first :

$$\epsilon = y_t \beta^{-1} e^{-\frac{1}{2}x_t} \qquad \mathrm{Var}(\epsilon) = \beta^{-2} e^{-x_t} \mathrm{Var}(y_t), \qquad \mathrm{Var}(y_t) = \beta^2 e^{x_t}. \tag{50}$$

Therefore, we can derive that $p(y_t|x_t, \beta) = \mathcal{N}(y_t|0, \beta^2 e^{x_t})$. The log-likelihood function can be written as below:

$$\ln p(y_t|x_t, \beta) = \mathrm{const} - \frac{1}{2}x_t - \frac{1}{2}y_t^2 e^{-x_t}\beta^{-2} = \ln p^*(y_t|x_t, \beta) + \mathrm{const}. \tag{51}$$

Since $\exp -x_t$ is a convex function, therefore, it can be bounded by some linear functions, h_t. One of the simplest way is to achieve it is to expand around the maximum of the conditional Gaussian density of the state variable, $p(x_t|x_{t-1}, x_{t+1}, \phi, \mu, \sigma_\eta^2)$. Before performing calculation, we further denote the mean x^* as $x^* = \mu + \phi\left[(x_{t-1} - \mu) + (x_{t+1} - \mu)\right]/(1 + \phi^2)$ and the variance as $v = \sigma_\eta^2/(1 + \phi^2)$. Then we can apply Taylor expansion around x^* as:

$$\ln p^*(y_t|x_t, \beta) \leq -\frac{1}{2}x_t - \frac{1}{2}y_t^2\left[\exp(-x_t^*) + (x_t^* - x_t)\exp(-x_t^*)\right] = \ln g^*(y_t|x_t, x^*, \beta), \tag{52}$$

therefore, we notice,

$$\ln p(x_t|x_{t-1}, x_{t+1}, \phi, \mu, \sigma_\eta^2)p(y_t|x_t, \beta) \leq \ln p(x_t|x_t^*, v^2)g^*(y_t|x_t, x_t^*, \beta). \tag{53}$$

Furthermore, after expanding the right hand side term with the help of Equation (46), following equation can be obtained:

$$\begin{aligned}\ln p(x_t|x_t^*, v^2)g^*(y_t|x_t, x_t^*, \beta) &= \text{const} - \frac{(x_t - x_t^*)^2}{2v^2} - \frac{1}{2}x_t - \frac{1}{2}y_t^2\left[\exp(-x_t^*) + (x_t^* - x_t)\exp(-x_t^*)\right] \\ &= \text{const} - \frac{x_t^2 - 2x_t^*x + x_t^{*2} + v^2\{x_t - y_t^2\left[\exp(-x_t^*) + (x_t^* - x_t)\exp(-x_t^*)\right]\}}{2v^2} \\ &= \text{const} - \frac{\left\{x_t - x^* - \frac{v^2}{2}\left[y_t^2\exp(-x_t^*) - 1\right]\right\}^2}{2v^2}.\end{aligned} \tag{54}$$

Since x_t is the random variable we interest and all other variable are already determined, they can be concluded. The above calculation demonstrated that the right part of Equation (31) is also proportional to a Gaussian density with a mean of $\mu^* = x^* + \frac{v^2}{2}\left[y_t^2\exp(-x_t^*) - 1\right]$ and variance v^2, which means:

$$p(x_t|x_t^*, v^2)g^*(y_t|x_t, x_t^*, \beta) = kp(x_t|\mu^*, v^2) = k\mathcal{N}(x_t|\mu^*, v^2) \tag{55}$$

Here the envelope function was already demonstrated to be greater or equal to the target density. The rejection-sampling procedure can be summarised below:

1. Samples x_t from $p(x_t|\mu^*, v^2)$.
2. Accept the x_t with probability $\frac{p(x_t|x_{t-1}, x_{t+1}, \phi, \mu, \sigma_\eta^2)p(y_t|x_t, \beta)}{kp(x_t|\mu^*, v^2)} = \frac{p^*(y_t|x_t, \beta)}{g^*(y_t|x_t, x_t^*, \beta)}$.
3. Go back to step 1.

4.4.2 Sampling the parameters

Sampling the σ_η^2. We use Gibbs sampling to draw σ_η^2 with using conjugate prior$\sigma_\eta^2|\phi, \mu \sim \mathcal{IG}\{a_\sigma/2, b_\sigma/2\}$, and then sample σ_η from, $\sigma_\eta^2|\phi, \mu, y, x \sim \mathcal{IG}\left\{\hat{\mathbf{a}}_\sigma/2, \hat{\mathbf{b}}_\sigma/2\right\}$, where $\hat{\mathbf{a}}_\sigma = a_\sigma + T$ and $\hat{\mathbf{b}}_\sigma = b_\sigma + (x_1 - \mu)^2(1 - \phi^2) + \sum_{i=2}^T[(x_i - \mu) - \phi(x_{i-1} - \mu)]^2$.

Derivation

We can derive it from the posterior. Firstly, use the Bayes Rule and we have:

$$\begin{aligned}p(\sigma_\eta^2|x, \phi, \mu, y) \propto p(\sigma_\eta^2, x, \phi, \mu, y) &= p(x|\mu, \phi, \sigma_\eta^2)p(\sigma_\eta^2) \\ &= p(\sigma_\eta^2)p(x_1|\mu, \phi, \sigma_\eta^2)\prod_{i=2}^n p(x_i|x_{i-1}, \mu, \phi, \sigma_\eta^2)\end{aligned} \tag{56}$$

Since x is AR(1) process and we can derive the posterior with multiplying the conditional likelihood for x and the prior as:

$$\begin{aligned}p(\sigma_\eta^2|x, \phi, \mu, y) &\propto p(\sigma_\eta^2)p(x_1|\mu, \phi, \sigma_\eta^2)\prod_{i=2}^n p(x_i|x_{i-1}, \mu, \phi, \sigma_\eta^2) \\ &\propto \mathcal{IG}\left(\sigma_\eta^2|\frac{a_\sigma}{2}, \frac{b_\sigma}{2}\right)\mathcal{N}\left(x_1|0, \frac{\sigma_\eta^2}{1 - \phi^2}\right)\prod_{i=2}^n\mathcal{N}\left(x_i|\mu + \phi(x_{i-1} - \mu), \sigma_\eta^2\right) \\ &\propto \sigma_\eta^{-T}\frac{b_\sigma^{1/2a_\sigma}\sigma_\eta^{-a_\sigma - 2}}{\Gamma(\frac{1}{2}a_\sigma)\exp(\frac{b_\sigma}{2\sigma_\eta^2})}\exp\left(\frac{-(x_1 - \mu)^2(1 - \phi^2) - \sum_{i=2}^T[(x_i - \mu) - \phi(x_{i-1} - \mu)]^2}{2\sigma_\eta^2}\right)\end{aligned} \tag{57}$$

where a_σ, b_σ is known as parameter and can generally be determined by using Maximum A Posterior. With rearranging equation and rewrite $v = \sigma_\eta^2$ and c=$b_\sigma^{1/2a_\sigma}/\Gamma(\frac{1}{2}a_\sigma)$,

$$p(\sigma_\eta^2|x,\phi,\mu,y) \propto cv^{-(T-1-a_\sigma)/2}\exp\left(\frac{-b_\sigma-(x_1-\mu)^2(1-\phi^2)-\sum_{i=2}^{T}[(x_i-\mu)-\phi(x_{i-1}-\mu)]^2}{2v}\right) \tag{58}$$

Substitute $\hat{\mathbf{a}}_\sigma$ and $\hat{\mathbf{b}}_\sigma$ back and it can be shown that the posterior probability is still an inverse-gamma distribution as following:

$$\begin{aligned}
p(\sigma_\eta^2|x,\phi,\mu,y) &\propto cv^{-\hat{\mathbf{a}}_\sigma+1}\exp\left(\frac{-\hat{\mathbf{b}}_\sigma}{2v}\right)\frac{\Gamma(\hat{\mathbf{a}}_\sigma)\hat{\mathbf{b}}_\sigma^{\hat{\mathbf{a}}_\sigma}}{\Gamma(\hat{\mathbf{a}}_\sigma)\hat{\mathbf{b}}_\sigma^{\hat{\mathbf{a}}_\sigma}} \\
&= c\frac{\Gamma(\hat{\mathbf{a}}_\sigma)}{\hat{\mathbf{b}}_\sigma^{\hat{\mathbf{a}}_\sigma}}v^{-\hat{\mathbf{a}}_\sigma+1}\exp\left(\frac{-\hat{\mathbf{b}}_\sigma}{2v}\right)\frac{\hat{\mathbf{b}}_\sigma^{\hat{\mathbf{a}}_\sigma}}{\Gamma(\hat{\mathbf{a}}_\sigma)} \\
&= dv^{-\hat{\mathbf{a}}_\sigma+1}\frac{\hat{\mathbf{b}}_\sigma^{\hat{\mathbf{a}}_\sigma}}{\Gamma(\hat{\mathbf{a}}_\sigma)}\exp\left(\frac{-\hat{\mathbf{b}}_\sigma}{2v}\right)
\end{aligned} \tag{59}$$

where $d = c\Gamma(\hat{\mathbf{a}}_\sigma)/\hat{\mathbf{b}}_\sigma^{\hat{\mathbf{a}}_\sigma}$ and posterior is proportional to the inverse-gamma distribution, but scale with the constant d.

Sampling ϕ. For sampling ϕ, we assume $\phi = 2\phi^*-1$ follows an Beta distribution $\phi^* \sim \mathrm{B}(\phi^{(1)}, \phi^{(2)})$, and the probability density ϕ of:

$$\pi(\phi^*) \propto \phi^{*\phi^{(1)}-1}(1-\phi^*)^{\phi^{(2)}-1} \quad \text{and} \quad \pi(\phi) \propto \left(\frac{\phi+1}{2}\right)^{\phi^{(1)}-1}\left(\frac{1-\phi}{2}\right)^{\phi^{(2)}-1} \tag{60}$$

where $\phi^{(1)}, \phi^{(2)} > 1/2$. In this project, $\phi^{(1)}$ was chosen to be 20 and $\phi^{(2)}$ was chosen to be 1.5, which implies the mean is 0.86. And the posterior $p(\phi|\sigma_\eta^2, x, \mu, y)$ can be computed as before:

$$\begin{aligned}
p(\phi|\sigma_\eta^2,x,\mu,y) &\propto \pi(\phi)p(x_1|\mu,\phi,\sigma_\eta^2)\prod_{i=2}^{n}p(x_i|x_{i-1},\mu,\phi,\sigma_\eta^2) \\
&\propto \frac{1}{\sqrt{1-\phi^2}}B(\phi)\exp\left(\frac{-(x_1-\mu)^2(1-\phi^2)-\sum_{i=2}^{T}[(x_i-\mu)-\phi(x_{i-1}-\mu)]^2}{2\sigma_\eta^2}\right) \\
&\propto \mathcal{N}(x_1|\mu,\phi,\sigma_\eta^2)B(\phi)\exp\left(\frac{-\sum_{i=2}^{T}[-2\phi(x_i-\mu)(x_{i-1}-\mu)+\phi^2(x_{i-1}-\mu)^2]}{2\sigma_\eta^2}\right)
\end{aligned} \tag{61}$$

With the technique of the completing the square, it can be demonstrated that the posterior is a product of two Gaussian with one beta distribution $p(\phi|\sigma_\eta^2, x, \mu, y) \propto \mathcal{N}(x_1|\mu, \phi, \sigma_\eta^2)\mathcal{N}(\phi|\hat{\phi}, s^2)B(\phi)$, where,

$$\hat{\phi} = \frac{\sum_{i=2}^{T}(x_i-\mu)(x_{i-1}-\mu)}{\sum_{i=2}^{T}(x_i-\mu)^2} \quad \text{and} \quad s^2 = \frac{\sigma_\eta^2}{\sum_{i=2}^{T}(x_{i-1}-\mu)^2} \tag{62}$$

Therefore, we are able to use Metropolis-Hasting Algorithm to perform the sampling, since the posterior is a joint distribution of a beta and two Gaussian distributions, the steps were summarised as below:

1. Sample from ϕ^* from the proposed distribution $\phi^* \sim \mathcal{N}(\hat{\phi}, s^2)$
2. Compute the acceptance probability $\frac{\mathcal{N}(\phi^{(j-1)}|\hat{\phi},s^2)\mathcal{N}(x_1|\mu,\phi^*,\sigma_\eta^2)\mathcal{N}(\phi^*|\hat{\phi},s^2)B(\phi^*)}{\mathcal{N}(\phi^*|\hat{\phi},s^2)\mathcal{N}(x_1|\mu,\phi^{(j-1)},\sigma_\eta^2)\mathcal{N}(\phi^{(j-1)}|\hat{\phi},s^2)B(\phi^{(j-1)})} = \frac{\mathcal{N}(x_1|\mu,\phi^*,\sigma_\eta^2)B(\phi^*)}{\mathcal{N}(x_1|\mu,\phi^{(j-1)},\sigma_\eta^2)B(\phi^{(j-1)})}$
3. Accept the ϕ^*, so $\phi^{(j)} = \phi^*$ otherwise, $\phi^{(j)} = \phi^{(j-1)}$

Sampling μ. For sampling μ, we need to apply a diffuse prior(non-informative prior), for instance, i.e.

$$\pi(\mu) = \text{const} \tag{63}$$

Therefore the posterior can be written as:

$$
\begin{aligned}
p(\mu|y,\phi,x,\sigma_\eta) &\propto p(\mu)p(x_1|\mu,\phi,\sigma_\eta^2)\prod_{i=2}^{n} p(x_i|x_{i-1},\mu,\phi,\sigma_\eta^2) \\
&\propto \exp\left(-\frac{(1-\phi^2)(h_1-\mu)^2}{2\sigma_\eta^2} - \frac{\sum_{i=2}^{T}\left[(h_t-\phi h_{t-1})-\mu(1-\phi)\right]^2}{2\sigma_\eta^2}\right) \\
&\propto \exp\left(\frac{2h_1(1-\phi^2)\mu-(1-\phi^2)\mu^2-(T-1)(1-\phi)^2\mu^2+2\sum_{i=2}^{T}(1-\phi)(h_t-\phi h_{t-1})\mu}{2\sigma_\eta^2}\right) \\
&\propto \exp\left(\frac{2\left[h_1(1-\phi^2)+\sum_{i=2}^{T}(1-\phi)(h_t-\phi h_{t-1})\right]\mu-\left[(1-\phi^2)+(T-1)(1-\phi)^2\right]\mu^2}{2\sigma_\eta^2}\right)
\end{aligned}
\tag{64}
$$

Applying the completing square method and $\mu|\phi,x,\sigma_\eta \sim \mathcal{N}(\hat{\mu},\sigma_\mu^2)$ can be obtained:

$$
\hat{\mu} = \sigma_\mu^2\left(\frac{h_1(1-\phi^2)+\sum_{i=2}^{T}(1-\phi)(h_t-\phi h_{t-1})}{\sigma_\eta^2}\right) \quad \text{and} \quad \sigma_\mu^2 = \sigma_\eta^2\left(\frac{1}{(1-\phi^2)+(T-1)(1-\phi)^2}\right) \tag{65}
$$

4.5 Result from the simulation

The posterior density from the Single step Gibbs sampling (From Algorithm 3) was plotted as below. The data was generated from the stochastic volatility model with the true parameter of $\phi = 0.99, \sigma = 0.1, \mu = 0.5$ and there are in total 1000 points are generated. In this case, we choose the burn-in period as 40000 steps and in total there are 460000 sweeps used to perform the single-step Gibbs sampling. The filtered latent state was computed as the mean of the state over all sweeps. From Table 4.5, it is fairly clear the Single-steps provide good estimation for the data, however, the simulation burn computer's memory and computing very slowly.

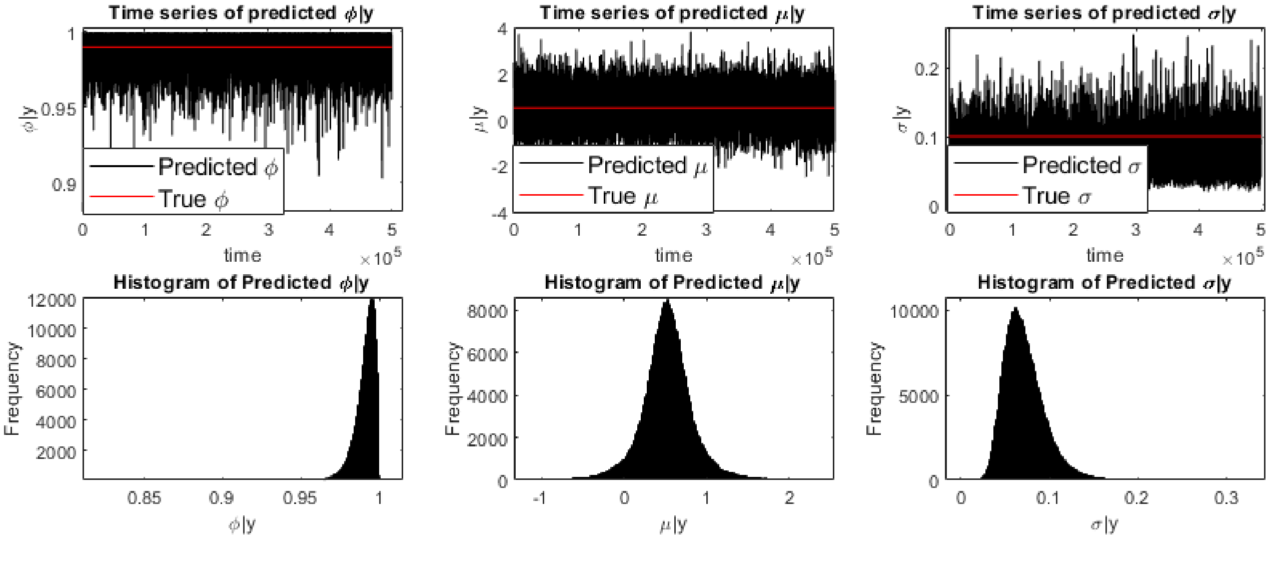

Figure 8: The simulation results generated from single step Gibbs sampling for parameters generated via the algorithm above to a 1000 time steps stochastic volatility model(for a burn-in period of 4000 and over 460000 iterations). As expected, the distribution of three parameter satisfies the original posterior as given above, which is beta-normal distribution for ϕ, normal for μ and inverse-gamma for σ^2.

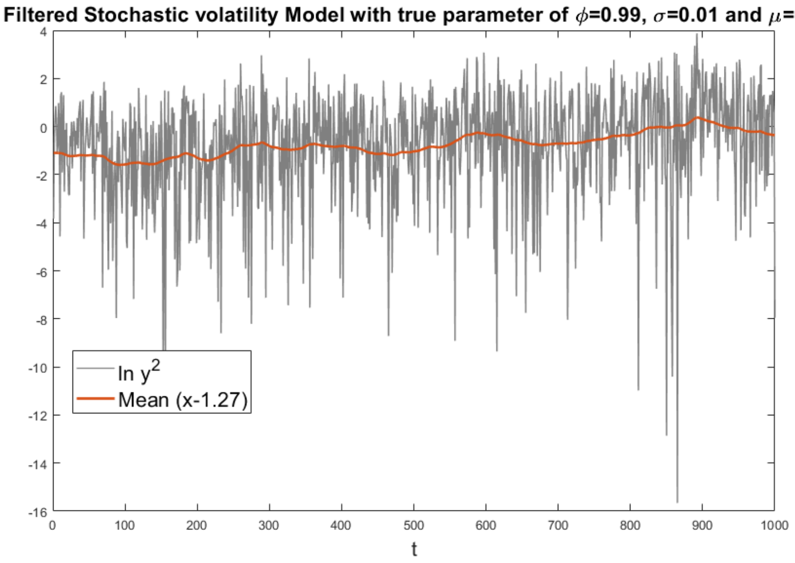

Figure 9: The filtered observation(which has been linearly shifted by 1.27) lies at around the mean of the $\ln y^2$ which indicates a good inference result. Meanwhile, the state estimation was taken the mean of the sampled result accrossing different sweeps.

Single Step Gibbs-SV model with $\phi = 0.99, \sigma = 0.1, \mu = 0.5$ with 40000 burn-in period with 460000 iteration				
Parameters	Mean from posterior	Standrad deviation	Stdandrad Error($\times 10^{-5}$)	Mean Squared Error
$\hat{\phi}$	0.9908	0.0069	1.0173	0.0069
$\hat{\mu}$	0.5268	0.3259	48.051	0.3270
$\hat{\sigma_\eta}$	0.0716	0.0230	3.3911	0.0365

4.6 Convergence rate

Despite the single step Gibbs sampler delivered some impressive results, with fairly good predicted parameters. However, the common issue that noticed for Gibbs sampler commonly reported is the slow convergence rate. The intuition is that state variable h is highly correlated, especially for a large persistence ϕ, therefore, while sampling from the posterior, the draws will only progress slowly, therefore, slow decaying correlation. Under the Gaussian approximation to $\ln \epsilon^2$, Pitt and Shepard suggests the analytically convergence rate of sampling h is the posterior $p(h|y,\theta)$ is[22]:

$$\rho = \frac{4\phi^2}{(1 + \phi^2 + \sigma^2/\mathrm{var}(\ln \epsilon^2))} \tag{66}$$

where $\mathrm{var}(\ln \epsilon^2)$. For our case, the theoretical convergence rate $\rho \approx 0.997$. To obtain a correlation less than 1%, we need to have at least 1600 simulations and the more important issue is about computer memory so in this case we could only simulate up to 500000 simulations. The alternative improve is that the multi-move Gibbs sampler or ASIS method proposed by Yu and Men[23] that boost the efficiency. In this project, we are going to present another techniques that use sequential monte-carlo to provide filtering and estimating in a more quicker way.

5 Particle filter

Only a small number of non-linear non-Gaussian state space models are fully analytic, therefore, obtaining the optimal estimator for state and parameter from the posterior conditioned on data may not be always possible, which makes the above Gibbs Sampling Method no longer applicable. Traditional recursive method such as Kalman filter relies on the linearity and normality of the model, therefore, the application of these method will require the crude functional approximations and local linearisation technique[24](as shown in the first section), the performance of the estimators may be sub-optimal, due to poor approximations densities. Particle filter, which firstly introduced at 1993 by Gordon and Salmond, provides an numerical simulation method to solve the problem of non-linear non-Gaussian model filtering and parameter estimation[25]. Since the particle filter can be applied recursively while the observation is possible, particle filter can ,therefore, provide an online filter to solve the filtering and estimation problem, which can be widely applied in many fields, including econometric and computer vision[24]. The basic idea of the particle filter is to use re-sampling method method to provide an estimate a more likely state that based on the model and data recursively via simulation, therefore, it also known as Sequential Monte Carlo. In this section, we firstly introduced the notation and definition need to present the algorithm of Sampling-Importance-Resampling algorithm with examples of the algorithm applied to stochastic volatility model. Followed by introducing a re-sampling scheme(Auxiliary particle filter), that can deliver more accurate result with less particle in the simulation, which adapted from Pitt and Shepard's paper(1999)[26] and how it can be applied to the stochastic volatility model. Furthermore, we reviewed the estimator of likelihood function of particle filter for auxiliary particle filter and derived the unbiasness of the estimator. At the end, we propose how MLE can be used to performance inference on parameters, however, due to time limit, we did not have time to show an example.

The aim of this section is that given a time series, $\{y_t, t = 1 \cdots T\}$, to show how time series can be modelled using a state space model frame via particle filter. Briefly speaking, particle filters use simulation to estimate the posterior $p(x_t|\mathcal{F}_t)$ as an approximation of the true posterior and update each time, where $t = 1, \cdots, t$ and $\mathcal{F}_t$ is the filtration that which includes all contemporary information till to time t. There are some pre-conditions to apply particle filter. Firstly, as discussed in the State Space Model section, $y_t|x_t$ need to be independent to each other and x_t were assumed to be Markovian. Furthermore, it is a important condition to apply the particle filters that likelihood $p(y_t|x_t)$ can be measured and, the transition from the density $p(x_{t+1}|x_t)$ can be simulated. Meanwhile, the state space model is well indexed and the state equations have a fixed parameters θ.

The likelihood of all observations y is defined as:

$$\log L(\theta) = \log p(y_1, \cdots, y_T|\theta), \tag{67}$$

where θ are parameters in the state space model. From the independent assumptions of $y_t|x_t$ of the state space model, the likelihood can be split as:

$$\log p(y_1, \cdots, y_T|\theta) = \sum_{t=1}^{T} \log p(y_t|\theta; \mathcal{F}_{t-1}). \tag{68}$$

Furthermore, under the state-space condition, from the Chapman-Kolmogorov equation for Markov Process, the prediction density $p(y_t|\theta; \mathcal{F}_{\sqcup - \infty})$ can be written as:

$$p(y_t|\theta; \mathcal{F}_{t-1}) = \int p(y_t|x_t; \theta) p(x_t|\mathcal{F}_{t-1}; \theta) dx_t, \tag{69}$$

where we use the particle filter to sample $x_t|\mathcal{F}_{t-1}; \theta$ and we are able to simulate transition process $x_{t+1}|x_t$from $p(x_t|x_{t-1})$.

5.1 Sampling importance re-sampling algorithm(SIR)

Sampling importance re-sampling algorithm(SIR) of Gordon et al.(1993)[25] is one of the resampling scheme that used by particle. Consider the following system, indexed by parameter θ,

$$y_t \sim p(y_t|x_t), x_{t+1} \sim p(x_{t+1}|x_t), t = 1, \cdots, T. \tag{70}$$

As discussed before, the observations are conditional independent given the corresponding latent state x_t, which evolves according to a Markov chain. The particle filter firstly introduced by Gordon at 1993 is a general approach to sequentially obtain a sample from the filtering distribution through time using all contemporary information available, mathematically, to obtain samples $x_t^k \sim p(x_t|\mathcal{F}_\sqcup)$, where $k = 1, \cdots, N$. In sampling the importance re-sampling algorithm, the only requirement to be satisfied are that we can simulate the transition process and we can to evaluate the likelihood function $p(y_t|x_t)$. Particle filters are the class of simulation filters which recursively approximate the state variable $x_t|\mathcal{F}_t$ by particles $x_t^1, \cdots, x_t^M$, with associate discrete probability mass of $\pi_t^1, \cdots, \pi_t^M$, which is drawn from $p(x_t|\mathcal{F}_t)$. The posterior $(p(x_t|\mathcal{F}_t)$ is conditional on all the available information till to t and is constructed by the combing prior with a likelihood evaluated by implementing particle filtering recursively. At each time step, the particle filter propagates and updates particles that can be used to obtain a sample which follows the same distribution as the true posterior $p(x_{t+1}|\mathcal{F}_{t+1})$; The true filtering density was defined as:

$$p(x_{t+1}|\mathcal{F}_{t+1}) \propto p(y_{t+1}|x_{t+1}) \int p(x_{t+1}|x_t) p(x_t|\mathcal{F}_t) dx_t \tag{71}$$

The particle filter approximates the continuous variable by discrete variables and treat these variable as the true filtering density, i.e.:

$$p(x_{t+1}|\mathcal{F}_{t+1}) \propto p(y_{t+1}|x_{t+1}) \sum_{j=1}^{M} p(x_{t+1}|x_t^j) \tag{72}$$

The above equation is also known as the empirical filtering density, where x_t^j is the particles drawn from $p(x_t|\mathcal{F}_t)$. The particle filter was applied recursively, that is, using the old particles to produce new particles $x_{t+1}^1, \cdots, x_{t+1}^M$ from the filtering density Equation (72). To sample from the filtering density, we need to use the sampling the importance re-sampling algorithm. To begin with, we need to sample a set of initial particles, $x_0^k \sim p(x_0), k = 1, \cdots, M$, where the initial particle is produced from some stationary distribution. The algorithm can be summarized as below:

Algorithm 5 SIR Algorithm

```
1: Sample a set of initial particles, $x_0^k \sim p(x_0), k = 1, \cdots, M$, where the weight of each particles is $1/M$
2: for $t = 1, \cdots, T-1$;
3:     Sample the particle $x_t^k \sim p(x_t|\mathcal{F}_t)$ for $k = 1, \cdots, M$; do
4:     Make a transition from the previous particle, $\tilde{x}_{t+1}^k|x_t^k$, from the transition density $p(x_{t+1}|x_t^k), \forall k = 1, \cdots, M$.
5:     Compute the normalized weight, as
           $$\pi_{t+1}^k = \frac{\omega_{t+1}^k}{\sum_{i=1}^M \omega_{t+1}^i}, \text{ where } \omega_{t+1}^k = p(y_{t+1}|\tilde{x}_{t+1}^k), \forall k = 1, \cdots, N.$$
6:     Sample M particles from of the mixture of particles, $x_{t+1}^k \sim \sum_{i=1}^M \pi_{t+1}^i \delta(x_{t+1}^k - \tilde{x}_{t+1}^i)$.
7:     Update the old particles to particles generated in the step 6, the weight of each particle is $1/M$
8: end
```

The algorithm will produce an approximation of the desired posterior density $p(x_{t+1}|\mathcal{F}_{t+1})$ at each time, in the large M limit, the approximated density will converge into the true density(See Theorem 1 and its proof in the Appendix).

The samples obtained from the step 1 of Algorithm can be used to generate samples for $\tilde{x}_{t+1}^k$, which converges to the predictive distribution $p(x_{t+1}|\mathcal{F}_t)$ as long the large M limit. And the 3rd step will also produces samples x_{t+1} from the predictive density $p(x_{t+1}|\mathcal{F}_{t+1})$ in the large N limit. Therefore, the samples from the empirical filtering density can treated as true posterior density in the large N limit, in SIR algorithm, we need to evaluate

$$\hat{\mathbf{p}}(x_{t+1}|\mathcal{F}_{t+1}) \propto p(y_{t+1}|x_{t+1}) \sum_{k=1}^{N} \pi_t^k p(x_{t+1}|x_t^k) \tag{73}$$

for each t. Furthermore, since our SIR algorithm produced an equally weight sample for the Empirical Density Function to true posterior density, we are able to estimate all moments. For any function $h(\cdot)$, the conditional expectation for the first moment $E\left[h(x_{t+1}|\mathcal{F}_{t+1})\right]$ can be estimated use the step 3 from the SIR algorithm via:

$$E\left[h(x_{t+1}|\mathcal{F}_{t+1})\right] = \frac{1}{N} \sum_{k=1}^{M} h(x_{t+1}^k) \tag{74}$$

Similarly for the higher moment and variance. We use the first moment for the prediction of the state at each time t. With this, we can also compute the likelihood function with the samples generated by SIR algorithm, firstly we need to compute the prediction density(Equation) with Equation (3) and Equation (6) and the fact that the sample from SIR algorithm $\tilde{x}_{t+1}^k \sim p(\tilde{x}_{t+1}^k|\mathcal{F}_t)$ as:

$$\hat{\mathbf{p}}(y_{t+1}|\mathcal{F}_t) = \frac{1}{N} \sum_{k=1}^{M} p(y_{t+1}|\tilde{x}_{t+1}^k) = \frac{1}{M} \sum_{k=1}^{M} \omega_{t+1}^k \tag{75}$$

The estimator is unbiased, the proof will be introduced in the later section. Furthermore, the general algorithm can be adapted to the stochastic volatility model as following:

Algorithm 6 SIR Algorithm applied to Stochastic Volatility Models

```
1: Sample a set of initial particles, $x_0^k \sim p(x_0), k = 1, \cdots, M$, where the weight of each particles is $1/M$
2: for $t = 1, \cdots, T-1$;
3:     Sample the particle $x_t^k \sim p(x_t|\mathcal{F}_t)$ for $k = 1, \cdots, M$; do
4:     Make a transitions for the particles, $\tilde{x}_{t+1}^k|x_t^k$, from the transition density $\mathcal{N}(x_{t+1}|\phi x_t + (1-\phi)\mu, \sigma^2)$.
5:     Compute the normalized weight, as
           $\pi_{t+1}^k = \frac{\omega_{t+1}^k}{\sum_{i=1}^M \omega_{t+1}^i}$, where $\omega_{t+1}^k = \mathcal{N}(y_{t+1}|0, e^{\tilde{x}_{t+1}^k})$.
6:     Sample M particles from of the mixture of particles, $x_{t+1}^k \sim \sum_{i=1}^M \pi_{t+1}^i \delta(x_{t+1}^k - \tilde{x}_{t+1}^i)$.
7:     Update the old particles to particles generated in the step 6, the weight of each particle is $1/M$
8: End
```

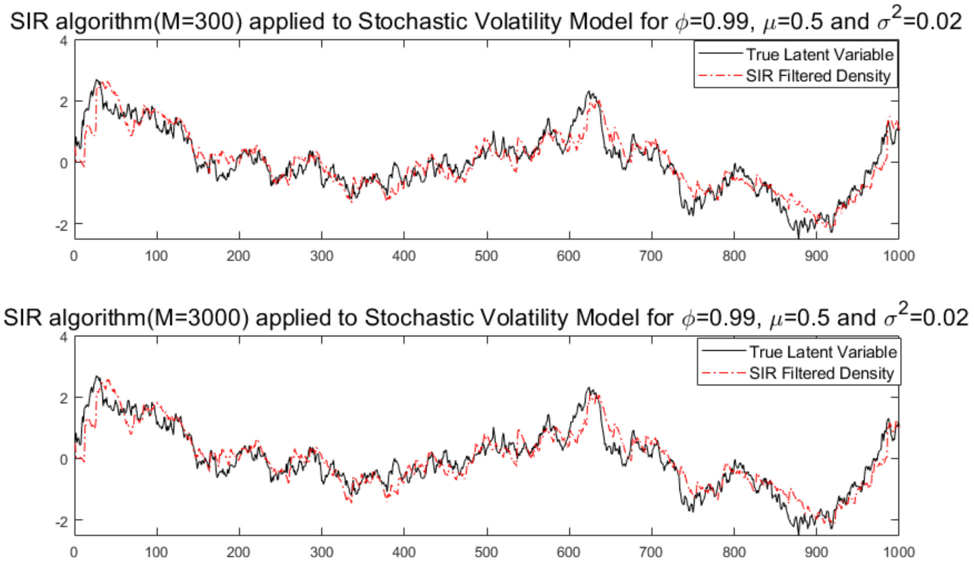

Figure 10: Simulation results from SIR with different number of particles, it demonstrates even with 3000 particles, the SIR algorithm produce a good estimate of true latent variable.

5.2 Auxiliary Particle Filter

It is one of the well-known weakness of the particle filters that the mixture structure will difficulties to adapt the SIR[26]. The problem in many case can be improved with performing particle filter in a higher dimension. The goal of the auxiliary particle filter is to sample from the posterior as before. The major idea is similar to the SIR, but instead making blind proposals transitions from some distributions directly, Auxiliary particle filter making proposal transitions incorporates with the observations y_{t+1} from some density $g(\cdot)$. There are a wide choice of the proposal density which is based on the different time series model and here is just the generic form. The auxiliary particle firstly approximate the posterior $p(x_{t+1}|\mathcal{F}_{t+1})$ by:

$$g(x_{t+1}, k|\mathcal{F}_{t+1}) \propto p(y_{t+1}|\mu_{t+1}^k)p(x_{t+1}|x_t) \tag{76}$$

where μ_{t+1}^k is the mean of the transition $a_{t+1}|a_t^k$. The mean can be computed numerically, via making M transitions for particle $k = 1, \cdots, M$ from the simulation or use the theoretical value. The we approximate the prediction density as:

$$g(k|\mathcal{F}_{t+1}) \propto \int p(y_{t+1}|\mu_{t+1}^k)p(x_{t+1}|x_t^k)dx_{t+1} = p(y_{t+1}|\mu_{t+1}^k) \tag{77}$$

The fist step is to sample from the approximated posterior $g(x_{t+1}, k|\mathcal{F}_{t+1})$, which is achieved by sampling an auxiliary variable k from the prediction density $g(k|\mathcal{F}_{t+1})$ and then sampling an transition $x_{t+1}|x_t^k$. The probability of an auxiliary variable k being sampled $\lambda_k \propto g(k|\mathcal{F}_{t+1})$ is also known an the first stage weight. Then we sampled from these first weighted particles, the intuitions are straightforward, that the higher likelihood particle will be more favoured to be sampled. Then we perform sampling again M times for these particles and make a transition, where the index j is the number of the draw at the second stage and k is the correspond to the particle been select from the first stage. Equivalently, the above process sampled from the approximated posterior, $g(a_{t+1}, k|\mathcal{F}_{t+1})$ are repeat M times. Finally, the probability was restored to the original distribution, which divide the approximated posterior by the approximated probability and compute the measurement density as the normalized second stage weight:

$$\omega_j = \frac{p(y_{t+1}|a^j_{t+1})}{g(y_{t+1}|\mu^{kj}_{t+1})}, \qquad \pi_j = \frac{\omega_j}{\sum_{i=1}^M \omega_i} \tag{78}$$

The sample generate from the above the multinomial distribution will have the same density as the target posterior $p_{x_{t+1}|\mathcal{F}_{t+1}}$ we desire.

The hope of the APF is that the second stage weights are less variable compare to original SIR method and have a quicker convergence. The method of making proposals which had a high conditional likelihood will reduce the costs of sampling many low likelihood particles, as a consequence, the low likelihood particles is less likely to be sampled at the second stage, which improves efficiency. In order to help our reader to follow, the general framework of APF is presented the algorithm as following:

Algorithm 7 ASIR algorithm

```
1: Sample a set of initial particles, $x_0^k \sim p(x_0), k = 1, \cdots, M$, where the weight of each particles is $1/M$
2: for $t = 1, \cdots, T-1$;
3:     Sample the particle $x_t^k \sim p(x_t|\mathcal{F}_t)$ for $k = 1, \cdots, M$; do
4:     Compute the first stage weight for these particles $\omega^k_{t|t+1} = g(y_{t+1}|\mu^k_t)\pi^k_t, \pi^k_{t|t+1} = \frac{\omega^k_{t|t+1}}{\sum_{i=1}^N \omega^i_{t|t+1}}$.
5:     Sample $\tilde{x}^k_t \sim \sum_{i=1}^N \omega^i_{t|t+1}\delta(x_t - x^i_t)$.
6:     Sample a transition for these particles and we have $x^k_{t+1} \sim g(x_{t+1}|\tilde{x}^k_t; y_{t+1})$.
7:     Compute the second stage weight,
           $$\omega^k_{t+1} = \frac{p(y_{t+1}|x^k_{t+1})p(x^k_{t+1}|\tilde{x}^k_t)}{g(y_{t+1}|\tilde{x}^k_t)g(x^k_{t+1}|\tilde{x}^k_t; y_{t+1})}, \pi^k_{t+1} = \frac{\omega^k_{t+1}}{\sum_{i=1}^N \omega^i_{t+1}}. \tag{79}$$
8:     Sample M particles from of the mixture of particles, $x^k_{t+1} \sim \sum_{i=1}^M \pi^i_{t+1}\delta(x^k_{t+1} - \tilde{x}^i_{t+1})$
9:     Update the old particles to particles generated in the step 6, the weight of each particle is $1/M$
```

The SIR method can be considered as a special case of Generalized ASIR, for instance, if we set $g(x_{t+1}|x_t) = p(x_{t+1}|x_t)$ and $g(y_{t+1}|x_t) = 1$.

5.3 Auxiliary Particle filtering to Stochastic volatility model

The choice of the proposed density is non-unique(this work is adapted from [26]), in this case, the proposed density is similar to the density used in the rejection sampling, with assumptions of the log concavity of the measurement density, i.e. $g(y_{t+1}|x_{t+1}, \mu^k_{t+1}) \geq p(y_{t+1}|x_{t+1})$. Therefore,

$$\begin{aligned} p(x_{t+1}, k|y_{t+1}) &\propto p(y_{t+1}|x_{t+1})p(x_{t+1}|x^k_t) \\ &\leq g(y_{t+1}|x_{t+1}, \mu^k_{t+1})p(x_{t+1}|x^k_t) \\ &= g(y_{t+1}|\mu^k_{t+1})g(x_{t+1}|x^k_t, y_{t+1}; \mu^k_{t+1}) \\ &\propto g(x_{t+1}, k|y_{t+1}) \end{aligned} \tag{80}$$

Therefore either rejection sampling or APF method can be applied, firstly we sample auxiliary variable k from the likelihood $g(y_{t+1}|\mu^k_{t+1})$ and then draw a_{t+1} from $g(a_{t+1}|a^k_t, y_{t+1}; \mu^k_{t+1})$. Then we compute the second stage weight as

$$\omega^j = \frac{p(y_{t+1}|x_{t+1})}{g(y_{t+1}|x_{t+1}; \mu^k_{t+1})} \tag{81}$$

For SV model, the log-likelihood $p(y_{t+1}|x_{t+1})$ is concave in $x_{t+1}$and we have $\mu^k_{t+1} = \phi x_t + (1-\phi)\mu$, where we have

$$\log g(y_{t+1}|x_{t+1};\mu^k_{t+1}) = \text{const} - \frac{1}{2}x_{t+1} - \frac{y^2}{2}\exp(-\mu^k_{t+1})\left\{1-(x_{t+1}-\mu^k_{t+1})\right\} \tag{82}$$

which implies,

$$g(y_{t+1}|x_{t+1};\mu^k_{t+1}) = \mathcal{N}\left[\mu^k_{t+1} + \frac{\sigma^2}{2}\left\{y^2\exp(-\mu^k_{t+1}) - 1, \sigma^2\right\}\right] = \mathcal{N}(\mu^{*k}_{t+1}, \sigma^2) \tag{83}$$

Similarly we have:

$$g(y_{t+1}|\mu^k_{t+1}) = \exp\left\{\frac{1}{2\sigma^2}(\mu^{*k2}_{t+1} - \mu^{k2}_{t+1})\right\}\exp\left\{-\frac{y^2}{2}\exp(-\mu^k_{t+1})(1+\mu^k_{t+1})\right\} \tag{84}$$

and accordingly the unnormalized second weight for particle is:

$$\omega_j = \exp\left(-\frac{y^2}{2}\left[\exp(-x_{t+1}) - \exp(-\mu^k_{t+1})\left\{1-(x_{t+1}-\mu^k_{t+1})\right\}\right]\right) \tag{85}$$

and then perform sampling from multinomial sampling.

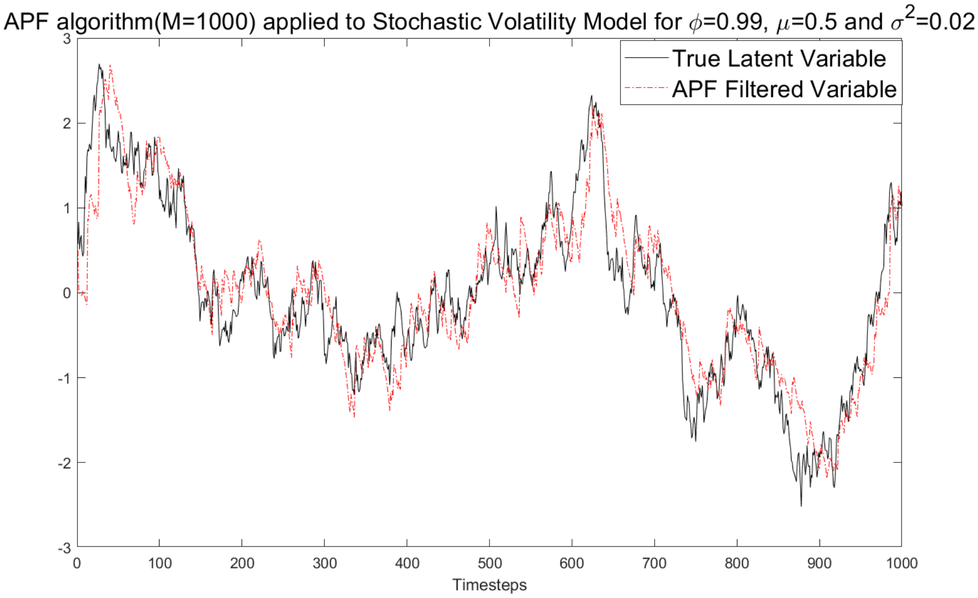

Figure 11: Simulation result for Auxiliary particle filter from the above method with 1000 particles used in total.

5.4 Likelihood Estimation

For Auxiliary Sampling Importance Re-sampling estimator of the prediction density $p(y_t|y_{1:t-1})$ is (this work is adapted from [27]):

$$\hat{\mathbf{p}}_N(y_{t+1}|y_{1:t}) = \frac{1}{M}\left(\sum_{k=1}^{N}\omega^k_{t+1}\right)\left(\sum_{k=1}^{N}\omega^k_{t|t+1}\right) \tag{86}$$

where ω^k_{t+1} is the second stage weight in APF and $\omega^k_{t|t+1}$ is the first stage weight. For SIR Algorithm, $\omega^k_t = p(y_t|x^k_t)$ and the first stage weight is $\omega^k_{t|t+1} = \pi^k_t$. After summing all k, the marginal probability will simply becomes to 1 by definition. Since the particles are generated from each simulation, the likelihood function will also varies since they are depending on the states we generated. To perform the parameter estimation, the likelihood function need to be maximized, where required the likelihood estimator is unbiased. Therefore, we need to demonstrate the likelihood function is unbiased so we can estimate the parameters via MLE. To show the proof, we firstly define some terminology as the following:

$$\begin{aligned} \hat{\mathbf{p}}_N(x_t|\mathcal{F}_t) &= \sum_{k=1}^{N} \pi_t^k \delta(x_t - x_t^k) \\ \hat{\mathbf{g}}_N(x_t|\mathcal{F}_{t+1}) &= \sum_{k=1}^{N} \pi_{t|t+1}^k \delta(x_t - x_t^k), \text{where} \quad x_t^k \sim \hat{\mathbf{p}}_N(x_t|\mathcal{F}_t) \end{aligned} \tag{87}$$

where $\pi_{t|t+1}^k$ and π_t^k is from the ASIR algorithm as above, from Bayes theorem , the proposal first stage weight can be written as:

$$\hat{\mathbf{g}}_N(x_{t+1}|\mathcal{F}_{t+1}) = \int g(x_{t+1}|\tilde{x}_t; y_{t+1})\hat{\mathbf{g}}_N(\tilde{x}_t|\mathcal{F}_{t+1})d\tilde{x}_t \tag{88}$$

where the first term corresponds to probability of a particle being selected according to some rules and then making a proposal move, which adapted to the observations at $t+1$. The second term $\hat{\mathbf{g}}_N(x_t|\mathcal{F}_{t+1})$ is the empirical density that a particle being selected, based on the next step observation y_{t+1}. The first stage weight $\hat{\mathbf{g}}_N(x_{t+1}|\mathcal{F}_{t+1})$ provide an approximated posterior from the proposal density that we can use in the step 3. Furthermore, the unnormlized weight of the particle can be written as

$$\omega_{t|t+1}(x_t) = g(y_{t+1}|x_t)\pi_t, \omega_{t+1}(x_{t+1}; x_t) = \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t)}{g(y_{t+1}|x_t)g(x_{t+1}|x_t; y_{t+1})} \tag{89}$$

The intuition of this equation is we need to restore the probability to be original posterior, therefore, we have dived the first stage weight. Therefore, we can rewrite the first step unnormalized weight at Algorithm 1 as $\omega_{t|t+1}^k = \omega_{t|t+1}(x_t^k)\pi_t^k$ and similar for $\omega_{t+1}^k = \omega_{t+1}(x_{t+1}^k; x_t^k)$. Therefore, the prediction density with approximating the posterior from the last time step as the empirical density from the simulation can be written as following:

$$\begin{aligned} p(y_{t+1}|\mathcal{F}_t) &= \int\int p(y_t|x_t)p(x_t|x_{t-1})p(x_{t-1}|\mathcal{F}_{t-1})dx_t dx_{t-1} \\ &\approx \int\int p(y_t|x_t)p(x_t|x_{t-1})\hat{\mathbf{p}}_N(x_{t-1}|\mathcal{F}_{t-1})dx_t dx_{t-1} \end{aligned} \tag{90}$$

With substituting four equation above and the following equation can be obtained:

$$\begin{aligned} p(y_{t+1}|\mathcal{F}_t) &\approx \int\int \frac{p(y_t|x_t)p(x_t|x_{t-1})}{g(y_{t+1}|x_t)g(x_{t+1}|x_t; y_{t+1})} g(y_{t+1}|x_t)g(x_{t+1}|x_t; y_{t+1})\hat{\mathbf{p}}_N(x_t|\mathcal{F}_t)dx_{t+1}dx_t \\ &= \int\int \omega_{t+1}(x_{t+1}; x_t)g(x_{t+1}|x_t; y_{t+1})\omega_{t|t+1}(x)dx_{t+1}dx_t \end{aligned} \tag{91}$$

In Monte-Carlo, while performing integral for Monte-Carlo integration we approximate the integral as summation of random number, for example:

$$\int xp(x)dx = \frac{1}{M}\sum x, \quad \text{where} \quad x \sim p(x), M \to \infty \tag{92}$$

Do it for the dx_{t-1}, therefore, we can obtain that

$$p(y_{t+1}|\mathcal{F}_t) \approx \left\{\sum_{k=1}^{N} \omega_{t-1|t}^k\right\} \int \hat{\mathbf{g}}_N(x_{t+1}|\mathcal{F}_{t+1})\omega_{t+1}(x_{t+1}; x_t)dx_{t+1} \tag{93}$$

where $\hat{\mathbf{g}}_N(x_{t+1}|\mathcal{F}_{t+1})$ is obtaining from the step 3 in the algorithm. With using g as a density , samples x_{t+1}^k can be drawn from $\hat{\mathbf{g}}_N(x_{t+1}|\mathcal{F}_{t+1})$ and therefore, $\omega_{t+1}(x_{t+1}^k, x_t^k)$ can be computed with using the law of the large number for MC integration, i.e.

$$\begin{aligned} p(y_{t+1}|\mathcal{F}_t) &\approx \left\{\sum_{k=1}^{N} \omega_{t-1|t}^k\right\} E_g(\omega_{t+1}(x_t^k, x_{t+1}^k)) \\ &= \left(\sum_{k=1}^{N} \frac{\omega_{t+1}^k}{N}\right)\left(\sum_{k=1}^{N} \omega_{t|t+1}^k\right) \end{aligned} \tag{94}$$

Suppose now at time t, the particles can be generated from the filtering density $p(x_t|\mathcal{F}_t)$, the probability of each particle x_t^k being generated is π_t^k and we denotes the collection these particles as $\mathcal{A}_t = \left\{x_t^k; \pi_t^k\right\}, \forall k = 1, \cdots, N$.

We further need to prove the mean estimator of the prediction density $p(y_{t+1}|\mathcal{F}_t)$ can be represented as computing the mean of the adapted likelihood of a particle being selected pending on the future observation, $p(y_{t+1}|x_t^k)$, i.e.

$$E[\hat{\mathbf{p}}_N(y_{t+1}|\mathcal{F}_t)|\mathcal{A}_t] = \sum_{k=1}^{N} p(y_t|x_{t-1}^k)\pi_{t-1}^k \tag{95}$$

To start with, we take expectation for the likelihood estimator:

$$E[\hat{\mathbf{p}}_N(y_{t+1}|\mathcal{F}_t)|\mathcal{A}_t] = E\left[\sum_{k=1}^{N} \frac{\omega_{t+1}^k}{N}|\mathcal{A}_t\right]\left(\sum_{k=1}^{N} \omega_{t|t+1}^k\right) \tag{96}$$

since the weights $\omega_{t|t+1}^k$ is known and it can be generated from the particles. With using the above equation, we can transform then back into the integral form as:

$$E[\hat{\mathbf{p}}_N(y_{t+1}|\mathcal{F}_t)|\mathcal{A}_t] = \left\{\sum_{k=1}^{N} \omega_{t-1|t}^k\right\} \int\int g(x_{t+1}|\tilde{x}_t; y_{t+1})\hat{\mathbf{g}}_N(\tilde{x}_t|\mathcal{F}_{t+1})\omega_{t+1}(x_{t+1}; x_t)dx_{t+1}d\tilde{x}_t \tag{97}$$

where $\tilde{x}_t$ is drawn from the density $\hat{\mathbf{p}}_N(x_t|\mathcal{F}_t)$. Use Monte-Carlo integration, substitute Equation 89 and Equation 87 and the integral can therefore be rewritten as:

$$\begin{aligned} E[\hat{\mathbf{p}}_N(y_{t+1}|\mathcal{F}_t)|\mathcal{A}_t] &= \left\{\sum_{k=1}^{N} \omega_{t|t+1}^k\right\} \int \sum_{k=1}^{N} \omega_{t+1}(x_{t+1}; x_t^k) g(x_{t+1}|x_t^k; y_t) \frac{\omega_{t|t+1}(x_t^k)}{\left(\sum_{j=1}^{N} \omega_{t|t+1}(x_t^j)\right)} dx_t \\ &= \int \sum_{k=1}^{N} \omega_{t|t+1}(x_t^k) g(x_{t+1}|x_t^k; y_t)\omega_{t|t+1}(x_t^k)dx_{t+1} \end{aligned} \tag{98}$$

Taking the summation out and performing integration first will not change the result and Equation89 is used to transform the integral into the original probability back :

$$\begin{aligned} E[\hat{\mathbf{p}}_N(y_{t+1}|\mathcal{F}_t)|\mathcal{A}_t] &= \sum_{k=1}^{N} \int \frac{p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^k)}{g(y_{t+1}|x_t)g(x_{t+1}|x_t^k; y_{t+1})} g(y_{t+1}|x_t^k)g(x_{t+1}|x_t^k; y_{t+1})\pi_t^k dx_{t+1} \\ &= \sum_{k=1}^{N} \int p(y_{t+1}|x_{t+1})p(x_{t+1}|x_t^k)\pi_t^k dx_{t+1} \\ &= \sum_{k=1}^{N} p(y_{t+1}|x_t^k)\pi_t^k \end{aligned} \tag{99}$$

Furthermore, it need to be proved, that for any moment $t - l$, the result is still correct, i.e. :

$$E[\hat{\mathbf{p}}_N(y_{t-l:t}|\mathcal{F}_{t-l-1})|\mathcal{A}_{t-l-1}] = \sum_{k=1}^{N} p(y_{t-l:t}|x_{t-l-1}^k)\pi_{t-l-1}^k \tag{100}$$

The result can be proved via induction, the proof can be separated into two parts. Firstly, it is assumed that the above equation for h is true. We then need to prove it is true for $h + 1$, i.e.:

$$E[\hat{\mathbf{p}}_N(y_{t-l-1:t}|\mathcal{F}_{t-l-2})|\mathcal{A}_{t-l-2}] = E[E[\hat{\mathbf{p}}_N(y_{t-l:t}|\mathcal{F}_{t-l-1})|\mathcal{A}_{t-l-1}]\hat{\mathbf{p}}_N(y_{t-l-1:t}|\mathcal{F}_{t-l-2})|\mathcal{A}_{t-l-2}] \tag{101}$$

The intuition for induction is, since the particles at each stage is generated from the previous stage, therefore, if we use the particle generated before, they should follow the same distribution at the next instant. Furthermore, using the definitions from previous equations and again use Monte-Carlo integration:

$$E[\hat{\mathbf{p}}_N(y_{t-l-1:t}|\mathcal{F}_{t-l-2})|\mathcal{A}_{t-l-2}] = E\left[\sum_{k=1}^{N} p(y_{t-l:t}|x_{t-l-1}^k)\pi_{t-l-1}^k \sum_{z=1}^{N} \frac{\omega_{t-l-1}^z}{N} \sum_{i=1}^{N} \omega_{t-l-2|t-l-1}^i |\mathcal{A}_{t-l-2}\right] \tag{102}$$

In the particle filter, the probability of particle can be computed from the step 4 for time at $t-h-1$ and from the previous step the particle mass was known and also for weight at $t-h-2$, therefore :

$$\begin{aligned} E[\hat{\mathbf{p}}_N(y_{t-l-1:t}|\mathcal{F}_{t-l-2})|\mathcal{A}_{t-l-2}] &= E\left[\sum_{k=1}^{N} p(y_{t-l:t}|x^k_{t-l-1})\frac{\omega^k_{t-l-1}}{\sum_{j=1}^{N}\omega^j_{t-l-1}}|\mathcal{A}_{t-l-2}\sum_{z=1}^{N}\frac{\omega^z_{t-l-1}}{N}\sum_{i=1}^{N}\omega^i_{t-l-2|t-l-1}\right] \\ &= E\left[\frac{1}{N}\sum_{k=1}^{N} p(y_{t-l:t}|x^k_{t-l-1})\omega^k_{t-l-1}|\mathcal{A}_{t-l-2}\right]\left[\sum_{i=1}^{N}\omega^i_{t-l-2|t-l-1}\right] \end{aligned} \tag{103}$$

The expectation part, again can be evaluated in a similar way to integral as before:

$$\begin{aligned} E[\hat{\mathbf{p}}_N(y_{t-l-1:t}|\mathcal{F}_{t-l-2})|\mathcal{A}_{t-l-2}] = \left[\sum_{i=1}^{N}\omega^i_{t-l-2|t-l-1}\right]\int & p(y_{t-l:t}|x_{t-l-1})\omega_{t-l-1}(x_{t-l-1};\tilde{x}_{t-l-2}) \\ &\times g(x_{t-l-1}|\tilde{x}_{t-l-2},y_{t-l-1})\hat{\mathbf{g}}_N(\tilde{x}_{t-l-2}|\mathcal{F}_{t-l-1})d\tilde{x}_{t-l-2}dx_{t-l-1} \end{aligned} \tag{104}$$

The calculation is similar as the previous :

$$\begin{aligned} E[\hat{\mathbf{p}}_N(y_{t-l-1:t}|\mathcal{F}_{t-l-2})|\mathcal{A}_{t-l-2}] &= \sum_{k=1}^{N}\pi^k_{t-l-2}\int p(y_{t-l:t}|x_{t-l-1})p(y_{t-l-1}|x_{t-l-1})p(x_{t-l-1}|x^k_{t-l-2})dx_{t-l-1} \\ &= \sum_{k=1}^{N}\pi^k_{t-l-2}p(y_{t-l-1:t}|x^k_{t-l-2}) \end{aligned} \tag{105}$$

as required. Now select $l = t-1$, it can be proved that it is true for $t = 0$:

$$E[\hat{\mathbf{p}}_N(y_{1:t}|\mathcal{F}_0)|\mathcal{A}_0] = \sum_{k=1}^{N} p(y_{1:t}|x^k_0)\pi^k_0 \tag{106}$$

where $\mathcal{F}_0$ is the observation obtained at time 0, however, there is no observation at time 0, so the dependence can be ignored. In the particle filter, we have set $x \sim p(x_0)$ and $\pi^k_0 = \frac{1}{N}$, therefore, at the limit of large N,

$$E[\hat{\mathbf{p}}_N(y_{1:t})|\mathcal{A}_0] = \sum_{k=1}^{N} p(y_{1:t}|x^k_0)\pi^k_0 = \int p(y_{1:t}|x_0)p(x_0)dx_0 = p(y_{1:t}) \tag{107}$$

We have demonstrate that as $N \to \infty$, the estimator of the likelihood is an unbiased estimator, with computing the mean of the prediction log-likelihood, the estimator will towards to the true likelihood function. Therefore, if we can perform inference for the parameter estimator via maximized the log-likelihood, i.e.

$$\hat{\theta} = \arg\max_{\theta}\left\{\sum_{t=1}^{T}\log\hat{\mathbf{L}}_M(\theta)\right\} = \arg\max_{\hat{\theta}}\left\{\sum_{t=1}^{T}\log\left[\left(\sum_{k=1}^{M}\frac{\omega^k_{t+1}}{M}\right)\left(\sum_{k=1}^{M}\omega^k_{t|t+1}\right)\right]\right\} \tag{108}$$

The only assumptions of the model can be applied is the observation and the state can be indexed, meanwhile, and the set of parameters, θ are fixed. From the Equation 68, the log-likelihood function of SIR can be computed via summing all prediction densities, where the prediction density is delivered from each iteration at the particle filter. The estimator of the likelihood for SIR algorithm can therefore be written as:

$$\log\hat{\mathbf{L}}_M(\theta,\mathcal{F}_t) = \sum_{t=1}^{T}\log\hat{\mathbf{p}}(y_t|\theta;\mathcal{F}_{t-1}) = \sum_{t=1}^{T}\log\left(\frac{1}{M}\sum_{k=1}^{M}\omega^k_t\right) \tag{109}$$

where $\omega^k_t = p(y_{t+1}|\tilde{x}^k_{t+1})$.

Due to limited amount time, we have not yet write code for performing maximum likelihood estimation. The method is simple, just compute the mean likelihood estimation given by the above equation and determine the parameter maximise the mean likelihood across many iterations.

6 Conclusion

In this project, we reviewed the the state space model with its applications of stochastic volatility model. The first two methods provide a reasonable good estimates for the true parameters and the state, however, they also have their weakness, for instance, when signal noise ratio is small(small σ_η), the filtered result is very noisy and can not really be used. The monte-carlo method, though produce optimal estimator for state and parameters, it converges very slowly and computationally inefficient and not really flexible. The particle filter, as a new method, with very flexible structure is a promising method. Due to limited amount of time, the likelihood estimation have not yet applied. Furthermore, if more time allowed, the vectorization of the code need to improve the simulation speed. Overall, we do appreciate this project that provides a better understanding of machine learning and time series.

7 Appendix

7.1 Generating AR1 process from OU process

For an Ornstein–Uhlenbeck process:

$$dx_t = \theta(\mu - x_t)dt + \sigma dW_t \tag{110}$$

The strong solution is:

$$x_t = x_0 e^{-\theta t} + \mu(1 - e^{-\theta t}) + \sigma \int_0^t e^{\theta s} dWs \tag{111}$$

We can simulate the process discreetly via:

$$x_{t_i} = e^{-k\Delta t} x_{t_{i-1}} + \theta(1 - e^{-k\Delta t}) + \sqrt{\frac{\sigma^2}{2k}(1 - e^{-2k\Delta t})}\epsilon \tag{112}$$

where ϵ is a standard normal. It is AR1 process if we take $\phi = e^{-k\Delta t}, \theta = \mu$ and $\sigma_\eta = \sqrt{\frac{\sigma^2}{2k}(1 - e^{-2k\Delta t})}$. After we set initial X_0, we can iterate to compute a stationary AR1 process, which is simple to run at Matlab.

7.2 Autocorrelation in stochastic volatility model

Despite the fact y_t is independent to each other, but if we apply log transformation, we found it is ARMA(1,1) process. To prove this, we can write:

$$\ln y_t^2 = x_t + \ln \epsilon_t^2 \tag{113}$$

From this we can already comment that at least $\ln y_t^2$ is AR(1) process. If ϵ_t follows normal distribution, then the mean and variance are known as -1.27 and 4.93 respectively, we can write the expression as

$$\ln y_t^2 = -1.27 + x_t + \xi_t \tag{114}$$

where $\xi_t = \ln \epsilon_t^2 + 1.27$ and is a i.i.d with zero mean and constant variance, and a logarithm of normal distribution, therefore we can write down the auto-covariance function as a summation of two process, i.e.

$$\gamma(h; \ln y_t^2) = \gamma(h; x_t) + \gamma(h; \xi_t) \tag{115}$$

where we know,

$$\gamma(0; \xi_t) = 4.93, \quad \gamma(h, \xi_t) = 0, \quad h = 1, 2, \cdots, \tag{116}$$

$$\gamma(h; x_t) = \phi^h \sigma_x^2. \tag{117}$$

Therefore, we can derive the auto-correlation function as

$$\rho(h; \ln y_t^2) = \frac{\phi^h \sigma_x^2}{4.93 + \sigma_x^2} \tag{118}$$

The auto-correlation function suggest an exponential decay and can be verified via using ACF and PACF in mat-lab. The result roughly agreed with the first few time lags, which exhibits exponentially decayed and from ACF given one observations.

7.3 Kalman smoother

The aim of filtering is to estimate the state vector, $\boldsymbol{\alpha_t}$, based on all information available till to t. For smoothing, the mechanism is similar and it takes account of the informations at later time than t. The smooth estimator, known as simply the smoother and denotes as $\mathbf{a_{t|T}}$. The smoother contains more informations than filter, from the past to the later future, therefore, in general it will have a smaller Mean square error than filter.

There are three basic algorithms in a linear models. *Fix-point smoothing* algorithm computes smoothed estimator of the state vector at some fixed time point, i.e. it gives $\mathbf{a}_{\boldsymbol{\tau}\mathbf{|t}}$ for any $t > \tau$. *Fix-lag smoothing* computes smoothed estimators for a fixed time delay, i.e., it gives $\mathbf{a_{t-j|t}},\ j = 1, \cdots, M$, where M is the maximum lag. *Fix-time interval* concerns computing the full set of the smoothed estimator for a fixed span of data, i.e. $\mathbf{a_{t|T}}, \forall\ t < T$, which is most widely used method method. The fixed-interval smoothing algorithm can used the following recursion equations, starting from the final quantity, $\mathbf{a_T}$ and $\mathbf{P_T}$, is given by Kalman filter and work backwards:

$$\mathbf{a_{t|T}} = \mathbf{a_t} + \mathbf{P^*_t}(\mathbf{a_{t+1|T}} - \mathbf{T_{t+1}a_t} - \mathbf{c_{t+1}}) \tag{119}$$

and

$$\mathbf{P_{t|T}} = \mathbf{P_t} + \mathbf{P^*_t}(\mathbf{P_{t+1|T}} - \mathbf{P_{t+1|t}})\mathbf{P^{*'}_t} \tag{120}$$

where $\mathbf{P^*_t} = \mathbf{P_t T'_{t+1} P^{-1}_{t+1|t}}$, where $\mathbf{a_{T|T}} = \mathbf{a_T}, \mathbf{P_{T|T}} = \mathbf{P_T}$.

7.3.1 Derivation of Kalman Smoother

From Kalman filter, we can obtain the likelihood for the state $\boldsymbol{\alpha_t}$, $p(\boldsymbol{\alpha_t}|\mathbf{y_{1:t}})$ and also the predictive distribution $p(\boldsymbol{\alpha_t}|\mathbf{y_{1:t-1}})$. Use the marginal distribution, we have(Notice it is always benefit to think graphically, the joint distribution can be think as the predictive probability of y_{t+1} given α_{t+1},transition probability from α_t to α_{t+1} times probability of α_t given a set of data from y, where the dominator is simply the normalization constant. The second step simply times the dominator and the numerator with $p(\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}})$):

$$\begin{aligned}
p(\boldsymbol{\alpha_t}|\mathbf{y_{1:n}}) &= \int p(\boldsymbol{\alpha_{t+1}}, \boldsymbol{\alpha_t}|\mathbf{y_{1:n}})d\boldsymbol{\alpha_{t+1}} \\
&= \int \frac{p(\boldsymbol{y_{t+1:n}}|\boldsymbol{\alpha_{t+1}})p(\boldsymbol{\alpha_{t+1}}|\boldsymbol{\alpha_t})p(\boldsymbol{\alpha_t}|\boldsymbol{y_{1:t}})}{p(\mathbf{y_{t+1:n}}|\mathbf{y_{1:t}})}d\boldsymbol{\alpha_{t+1}} \\
&= \int \frac{p(\boldsymbol{\alpha_{t+1}}|\boldsymbol{\alpha_t})p(\boldsymbol{\alpha_t}|\boldsymbol{y_{1:t}})}{p(\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:t}})} \times \frac{p(\boldsymbol{y_{t+1:n}}|\boldsymbol{\alpha_{t+1}})p(\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}})}{p(\mathbf{y_{t+1:n}}|\mathbf{y_{1:t}})}d\boldsymbol{\alpha_{t+1}} \\
&= \int p(\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t+1}}; \boldsymbol{y_{1:t}}) \times p(\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}})d\boldsymbol{\alpha_{t+1}}
\end{aligned} \tag{121}$$

The joint density of α_t and α_{t+1} are multivariate-Gaussian and we can again write as following:

$$\begin{pmatrix} \boldsymbol{\alpha_t} \\ \boldsymbol{\alpha_{t+1}} \end{pmatrix} |\mathbf{y_{1:t}} \sim \mathcal{N}\left(\begin{pmatrix} \mathbf{a_t} \\ \mathbf{a_{t+1|t}} \end{pmatrix}, \begin{pmatrix} \mathbf{P_t} & \mathbf{C^*_t} \\ \mathbf{C^*_t} & \mathbf{P_{t+1|t}} \end{pmatrix}\right) \tag{122}$$

We can compute the correlation between $\boldsymbol{\alpha_{t+1}}$ and $\boldsymbol{\alpha_t}$, $\mathbf{C^*_t}$ with the updating equation via:

$$C^*_t = E\left[(\boldsymbol{\alpha_t} - \mathbf{a_t})(\boldsymbol{\alpha_{t+1}} - \mathbf{a_{t+1}})'\right] = E\left[(\boldsymbol{\alpha_t} - \mathbf{a_t})(\boldsymbol{T_t}\boldsymbol{\alpha_t} - \mathbf{T_t a_t} - \mathbf{R_t}\eta_\mathbf{t})'\right] = \mathbf{P_t T'} \tag{123}$$

Use the similar approach as before, $E[Z|x] = \mu_Z + \Sigma_{zx}\Sigma^{-1}_{xx}(x - \mu_X)$, we can very easily obtain that:

$$E\left[\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t+1}}, \boldsymbol{y_{1:t}}\right] = \mathbf{a_t} + \mathbf{P_t T' P^{-1}_{t+1|t}}\left(\boldsymbol{\alpha_{t+1}} - \mathbf{a_{t+1|t}}\right). \tag{124}$$

We can therefore obtain $E[\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}}] = \boldsymbol{a_{t+1}}|\boldsymbol{n}$ and we therefore can compute conditional expectation with using the tower-property of conditional expectation $E(\boldsymbol{\alpha_t}|\boldsymbol{y_{1:n}}) = E_{\alpha_t}\left[E_{\alpha_{t+1},y_{1:t}}[\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t+1}}, \mathbf{y_{1:t}}]|\mathbf{y_{1:n}}\right]$:

$$E(\boldsymbol{\alpha_t}|\boldsymbol{y_{1:n}}) = E_{\alpha_t}\left[E_{\alpha_{t+1},y_{1:t}}[\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t+1}}, \mathbf{y_{1:t}}]|\mathbf{y_{1:n}}\right] = \mathbf{a_t} + \mathbf{P_t T'_t P^{-1}_{t+1|t}}(\mathbf{a_{t+1|n}} - \mathbf{a_{t+1|t}}) \tag{125}$$

Use the law of the total expectation, i.e.

$$\begin{aligned}
V_X(X) &= E[X^2] - E[X]^2 \\
&= E_Y[E_X[X^2|Y]] - \{E_Y[E_X[X|Y]]\}^2 \\
&= E_Y[V_x[X|Y] + E_X[X|Y]^2] - \{E_Y[E_X[X|Y]]\}^2 \\
&= V_Y(E_X[X|Y]) + E_Y[V_X[X|Y]]
\end{aligned} \tag{126}$$

With using this property, we can compute $V[\boldsymbol{\alpha_t}|\boldsymbol{y_{1:n}}]$ via using the result from the portioned Gaussian(Notice, since it is taking expectation so only the random variable part count,i.e. subject to α):

$$\begin{aligned} V[\boldsymbol{\alpha_t}|\boldsymbol{y_{1:n}}] &= V_{\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}}}[E[\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t+1}},\boldsymbol{y_{1:t}}]] + E_{\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}}}[V[\boldsymbol{\alpha_t}|\boldsymbol{\alpha_{t+1}},\boldsymbol{y_{1:t}}]] \\ &= V_{\boldsymbol{\alpha_{t+1}}|\boldsymbol{y_{1:n}}}[\mathbf{P_t T' P^{-1}_{t+1|t}}\alpha_{\mathbf{t+1}}] + (\mathbf{P_t} + \mathbf{P_t T'_t P^{-1}_{t+1|t} T' P_t} \\ &= \mathbf{P_t T' P^{-1}_{t+1|t} P_{t+1|n} P^{-1}_{t+1|t} T' P_t} + \mathbf{P_t T' P^{-1}_{t+1|t}}\alpha_{t+\mathbf{1}}] + \mathbf{P_t} + \mathbf{P_t T'_t P^{-1}_{t+1|t} T' P_t} \end{aligned} \tag{127}$$

as required.

7.4 General Proof of detailed balance from Gibbs sampling

In order to demonstrate Gibbs sampling will provide the stationary distribution, if the detailed balance condition is provided. Consider a two-dimensional density(can be generalised to higher dimensional space, it is similar!),

$$\pi(\boldsymbol{\theta}) = \pi(\theta_1, \theta_2) = \pi(\theta_1)\pi(\theta_2|\theta_1) = \pi(\theta_2)\pi(\theta_1|\theta_2) \tag{128}$$

For Gibbs sampling, transition kernel was used to update the probability distribution, for 2-d case:

$$q(\boldsymbol{\theta}^{i+1}|\boldsymbol{\theta}^i) = \pi(\theta_2^{i+1}|\theta_1^{i+1})\pi(\theta_1^{i+1}|\theta_2^i) \tag{129}$$

where i corresponds to the number of sweeps. The detailed balance condition requires that,

$$q(\boldsymbol{\theta}^{i+1}|\boldsymbol{\theta}^i)\pi(\boldsymbol{\theta}^i) = q(\boldsymbol{\theta}^i|\boldsymbol{\theta}^{i+1})\pi(\boldsymbol{\theta}^{i+1}) \tag{130}$$

If the detailed balanced condition is satisfied, we have in-variance of the distribution after each sampling. This can be showed by integrating the both side of detailed balance Equation (16) with respect to $\boldsymbol{\theta}^i$, the right hand side simply integrate to 1 as a general result, i.e.

$$\int q(\boldsymbol{\theta}^{i+1}|\boldsymbol{\theta}^i)\pi(\boldsymbol{\theta}^i)d\boldsymbol{\theta}^i = \pi(\boldsymbol{\theta}^{i+1}) \tag{131}$$

In-variance is a weaker condition than detailed balance and we have showed that the detail balance will imply in-variance. Therefore, the distribution will need to satisfy the in-variance condition. The Gibbs sampling satisfy the invariance condition, we can show that as the following:

$$\begin{aligned} \int q(\boldsymbol{\theta}^{i+1}|\boldsymbol{\theta}^i)\pi(\boldsymbol{\theta}^i)d\boldsymbol{\theta}^i &= \int\int \pi(\theta_2^{i+1}|\theta_1^{i+1})\pi(\theta_1^{i+1}|\theta_2^i)\pi(\theta_1^i,\theta_2^i)d\theta_1^i d\theta_2^i \\ &= \pi(\theta_2^{i+1}|\theta_1^{i+1})\int \pi(\theta_1^{i+1}|\theta_2^i)\left[\int \pi(\theta_1^i|\theta_2^i)d\theta_1^i\right]\pi(\theta_2^i)d\theta_2^i \\ &= \pi(\theta_2^{i+1}|\theta_1^{i+1})\pi(\theta_1^{i+1}) = \pi(\theta_1^{i+1},\theta_2^{i+1}) \end{aligned} \tag{132}$$

and invariance, therefore, we only need to verify $\pi(\boldsymbol{\theta}^i)$ is equivalent to $\pi(\boldsymbol{\theta}^{i+1})$. Meanwhile, we can also prove the Gibbs sampling satisfies the detailed balance condition, firstly, we applied the Gibbs sampling with different orders(we can show they are equivalent using Bayes rule and re-labelling the indices), i.e.

$$q(\boldsymbol{\theta}^{i+1}|\boldsymbol{\theta}^i)\pi(\boldsymbol{\theta}^i) = \frac{1}{2}\pi(\theta_2^{i+1}|\theta_1^{i+1})\pi(\theta_1^{i+1}|\theta_2^i)\pi(\theta_1^i|\theta_2^i)\pi(\theta_2^i) + \frac{1}{2}\pi(\theta_1^{i+1}|\theta_2^{i+1})\pi(\theta_2^{i+1}|\theta_1^i)\pi(\theta_2^i|\theta_1^i)\pi(\theta_1^i) \tag{133}$$

With Bayes Rule, noting $\pi(\theta_1^{i+1},\theta_2^i) = \pi(\theta_1^{i+1}|\theta_2^i)\pi(\theta_2^i) = \pi(\theta_2^i|\theta_1^{i+1})\pi(\theta_1^{i+1})$, and then we have:

$$\begin{aligned} q(\boldsymbol{\theta}^{i+1}|\boldsymbol{\theta}^i)\pi(\boldsymbol{\theta}^i) &= \frac{1}{2}\pi(\theta_2^{i+1}|\theta_1^{i+1})\pi(\theta_1^{i+1})\pi(\theta_1^i|\theta_2^i)\pi(\theta_2^i|\theta_1^{i+1}) + \frac{1}{2}\pi(\theta_1^{i+1}|\theta_2^{i+1})\pi(\theta_2^{i+1})\pi(\theta_1^i|\theta_2^{i+1})\pi(\theta_2^i|\theta_1^i) \\ &= \frac{1}{2}\pi(\theta_1^{i+1},\theta_2^{i+1})\pi(\theta_1^i|\theta_2^i)\pi(\theta_2^i|\theta_1^{i+1}) + \frac{1}{2}\pi(\theta_1^{i+1},\theta_2^{i+1})\pi(\theta_1^i|\theta_2^{i+1})\pi(\theta_2^i|\theta_1^i) \\ &= \pi(\boldsymbol{\theta}^{i+1})q(\boldsymbol{\theta}^i|\boldsymbol{\theta}^{i+1}) \end{aligned} \tag{134}$$

which we have demonstrated detailed balance, which implies the existence of stationary distribution. If the Markov chain is aperiodic, we can ensure the Gibbs sampling is a valid scheme and will converge to the distribution we desire.

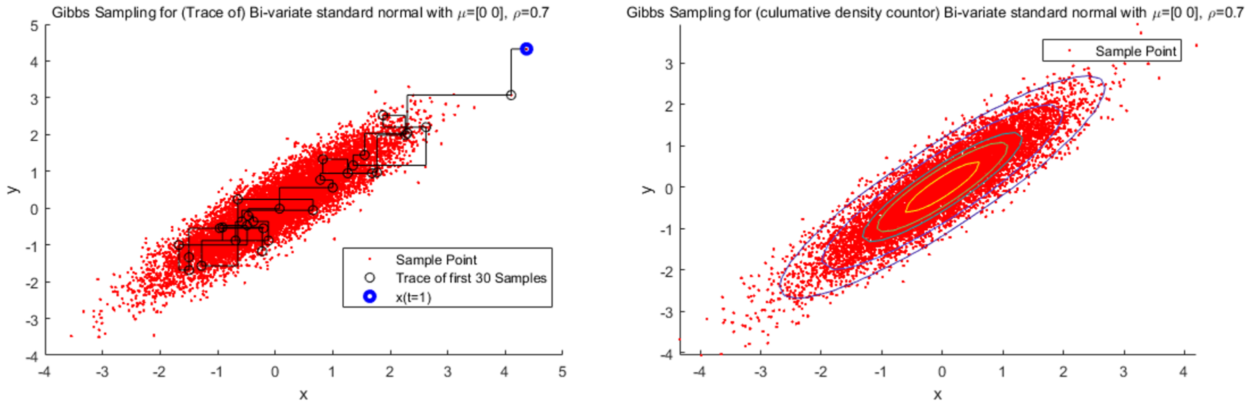

Figure 12: Gibbs sampling illustrations with an initial point at $(5, 5)$, the sampled point start at a quite far point from the desired mean, and then through our iteration, it converges bi-variate as we desired distribution. The contour line corresponds to theoretical cumulative density, which we can see agreement between these results, implies gibbs sampling indeed give us desired distribution.

7.4.1 Gibbs sampling for the Bi-variate Normal

Before progress to SV model, we will illustrate how Gibbs sampling work on Bi-variate normal to make us familiar with the method. Suppose we want to obtain samples from a Bi-variate distribution, i.e.

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 1 & \rho \\ \rho & 1 \end{pmatrix}\right] \tag{135}$$

Suppose we c an only generate samples from a uni-variate distribution. From the result from partitioned Gaussian(we can either do it from marginal density or use the result of reverting block matrix), we can derive the conditional probability as

$$(X|Y = y) \sim \mathcal{N}(\rho y, (1 - \rho^2)) \quad \text{and} \quad (Y|X = x) \sim \mathcal{N}(\rho x, (1 - \rho^2)) \tag{136}$$

and we can simulate this distribution from these marginal distribution via following rules.

Firstly, construct a Markov Chain, set initial condition $X^{(0)} = x_0$ and then draw a sample from the above distribution, therefore, we will have the following process:$X^{(0)} = x_0 \to Y^0 \to X^{(1)} \to Y^{(1)} \to \cdots \to X^{(k)} \to Y^{(k)} \to \cdots$, therefore, at each k, we know that x_k, y_k is normally distributed around:

$$\begin{aligned} (Y^{(k)}|X^{(k)} = x_k) &\sim \mathcal{N}(\rho x_k, 1 - \rho^2) \\ (X^{(k)}|Y^{(k-1)} = y_{k-1}) &\sim \mathcal{N}(\rho y_{k-1}, 1 - \rho^2) \end{aligned} \tag{137}$$

The joint distribution $P(X^{(k)}, Y^{(k)})$ can be written as:

$$\begin{pmatrix} X^{(k)} \\ Y^{(k)} \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \rho^{2k}x_0 \\ \rho^{2k+1}x_0 \end{pmatrix}, \begin{pmatrix} 1 - \rho^{4k} & \rho(1 - \rho^{4k}) \\ \rho(1 - \rho^{4k}) & 1 - \rho^{4k+2} \end{pmatrix}\right] \tag{138}$$

From the above equation, we can suggest as long as the $|\rho| < 1$, at the long-time limit, the joint distribution will always converges to stationary distribution, regardless the initial condition.

For the state space model, the main objective is obtain the posterior density of the parameters $p(\theta|y_{1:n})$ or latent density $p(x_{0:n}|y_{1:n})$. For stochastic volatility model, it is generally easier to get the joint density $p(\theta, x_{0:n}|y_{1:n})$ and then take the average to compute the marginal to obtain the posterior $p(\theta|y_{1:n})$.

Derivation from general form

Suppose we have a general bi-variate distribution

$$\begin{pmatrix} X \\ Y \end{pmatrix} \sim \mathcal{N}\left[\begin{pmatrix} \mu_1 \\ \mu_2 \end{pmatrix}, \begin{pmatrix} \sigma_1^2 & \rho\sigma_1\sigma_2 \\ \rho\sigma_1\sigma_2 & \sigma_2^2 \end{pmatrix}\right] \tag{139}$$

Use the result from the partitioned Gaussian, we can write the conditional distributions as follows:

$$\begin{aligned} X|Y &\sim \mathcal{N}\left(\mu_1 + \rho\frac{\sigma_1}{\sigma_2}(y-\mu_2), (1-\rho^2)\sigma_1^2\right) \\ Y|X &\sim \mathcal{N}\left(\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x-\mu_1), (1-\rho^2)\sigma_2^2\right) \end{aligned} \tag{140}$$

Therefore, if we sampling $X^{(k)}$ from distribution from Markov Chain, we can express the X and Y as:

$$\begin{aligned} X^k &= \mu_1 + \rho\frac{\sigma_1}{\sigma_2}(y^{k-1}-\mu_2) + \sqrt{(1-\rho^2)\sigma_1^2}\xi_t \\ Y^k &= \mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x^k-\mu_1) + \sqrt{(1-\rho^2)\sigma_2^2}\eta_t \end{aligned} \tag{141}$$

Replace the x^k into the second equation into the first equation,

$$\begin{aligned} X^k &= \mu_1 + \rho\frac{\sigma_1}{\sigma_2}\left[\mu_2 + \rho\frac{\sigma_2}{\sigma_1}(x^{k-1}-\mu_1) + \sqrt{(1-\rho^2)\sigma_2^2}\eta_t - \mu_2\right] + \sqrt{(1-\rho^2)\sigma_1^2}\xi_t \\ &= \mu_1 + \rho^2(x^{k-1}-\mu_1) + \rho\sqrt{(1-\rho^2)\sigma_1^2}\eta_t + \sqrt{(1-\rho^2)\sigma_1^2}\xi_t, \end{aligned} \tag{142}$$

where ξ_t and η_t is standard normal with zero mean and unit variance. Therefore we can derive the mean $E(X^{(k)}) = \mu_1 + \rho^2(x^{k-1}-\mu_1)$ and variance $\mathrm{Var(X^{(k)})} = (1-\rho^4)\sigma_1^2$. Similarly, we can compute for Y^k, i.e.,

$$Y^k = \mu_2 + \rho^2(y^{k-1}-\mu_2) + \rho\sqrt{(1-\rho^2)\sigma_2^2}\eta_t + \sqrt{(1-\rho^2)\sigma_2^2}\xi_t, \tag{143}$$

and also the mean $E(Y^{(k)}|X^{(k)}) = \mu_2 + \rho^2(x^{(k)}-\mu_2)$ and also the variance $\mathrm{Var(Y^{(k)}|X^{(k)})} = (1-\rho^4)\sigma_2^2$, which is the general form. The above Markov chain can be treated as AR (1) process. For AR(1) process, from Equation (8) and Equation (9), we are able to derive the mean of the equation. Furthermore, we can also derive the variance of AR(1) process, since the series is finite, we can write the summation of geometric series of AR(1) as:

$$\mathrm{Var}(x_k) = \frac{1-\rho^{4k}}{1-\rho^4} \times 1 - \rho^4 = 1-\rho^{4k}, \qquad \mathrm{Var}(y_k) = \frac{1-\rho^{4k+2}}{1-\rho^4} \times 1 - \rho^4 = 1-\rho^{4k+2} \tag{144}$$

The co-variance between $X^{(k)}, Y^{(k)}$ can be treated as lag-1 in AR(1) process, with using computing co-variance $\gamma(X^{(k)}Y^{(k)})$ from Equation (7) and also replacing formula with Equation (8) in as:

$$\mathrm{Cov}(X^{(k)}, Y^{(k)}) = \rho\mathrm{Var}(X^{(k)}) \tag{145}$$

Therefore, we obtained the joint probability distribution as required Equation (4).

7.4.2 Detail Balance Condition and Stationary distribution

A Markov process is called a reversible Markov process if it satisfies the detail balance equation. If the Markov process is regular, then the reversible Markov process will guarantee the Markov process posses stationary distribution. The detailed balance equation implies that there is no probability current between two state, therefore, stationary. The detail-balance equation can be written for joint probability $p(x_k, x_{k+1})$as:

$$p(x_k|x_{k+1})p(x_{k+1}) = p(x_{k+1}|x_k)p(x_k) \tag{146}$$

The above Gibbs sampling follows detailed balance. Firstly, we can see that the variance and mean will converge to stationary states, i.e. $\pi(x) \sim \mathcal{N}(0,1), \pi(y) \sim \mathcal{N}(0,1)$. For the Gibbs we know the current value is only dependent on the previous value, as Equation (5) by noting both $\sigma_1 = \sigma_2 = 1$. By recognising the constant term is identical, take transformation of log and we have

$$\begin{aligned} -\frac{1}{2}\frac{(y-\rho x)^2}{1-\rho^2} - \frac{1}{2}x^2 &= -\frac{1}{2}\frac{y^2 - 2\rho xy + \rho^2x^2 + (1-\rho^2)x^2}{1-\rho^2} = -\frac{1}{2}\left(\frac{1-\rho^2}{1-\rho^2}y^2 + \frac{\rho^2y^2 - 2\rho xy + x^2}{1-\rho^2}\right) \\ &= -\frac{1}{2}\frac{(x-\rho y)^2}{1-\rho^2} - \frac{1}{2}y^2 \end{aligned} \tag{147}$$

Therefore, this further guarantee the Gibbs sampling will governs the stationary distribution.

7.5 Proves of MH converges to stationary distribution

Wcan demonstrate that the $\pi(\theta)$ is a stationary distribution via showing that the transition probability $p_{\mathrm{MH}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})$ satisfied detailed balance condition, i.e.

$$p_{\mathrm{MH}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})\pi(\boldsymbol{\theta}^{(t-1)}) = p_{\mathrm{MH}}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \tag{148}$$

where $p_{\mathrm{MH}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)}) = \alpha(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})+p_{\mathrm{A}}(\boldsymbol{\theta}^{(t-1)})\delta(\boldsymbol{\theta}\boldsymbol{-}\boldsymbol{\theta}^{(t-1)})$, similarly for $p_{\mathrm{MH}}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})$, where $p_{\mathrm{A}}(\boldsymbol{\theta}^{(t-1)})$ denotes the probability of the variable $\boldsymbol{\theta}$ remain the same as the last instant $\boldsymbol{\theta^t} = \boldsymbol{\theta}$:

$$p_{\mathrm{A}}(\boldsymbol{\theta}^{(t-1)}) = 1 - \int \alpha(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})d\boldsymbol{\theta}^{(t-1)} \tag{149}$$

To demonstrate the Metropolis-Hasting algorithm satisfy the detail balance condition, we can relabel the second term since it involves the delta-function so we can swap the variables also with some manipulation we can obtain Detailed balance conditions, i.e. :

$$\begin{aligned}
p_{\mathrm{MH}}(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})\pi(\boldsymbol{\theta}^{(t-1)}) &= \left[\alpha(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)}) + p_{\mathrm{A}}(\boldsymbol{\theta}^{(t-1)})\delta(\boldsymbol{\theta} - \boldsymbol{\theta}^{(t-1)})\right]\pi(\boldsymbol{\theta}^{(t-1)}) \\
&= \min\left(1, \frac{q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}{q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})\pi(\boldsymbol{\theta}^{(t-1)})}\right) q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})\pi(\boldsymbol{\theta}^{(t-1)}) \\
&\qquad\qquad\qquad + p_{\mathrm{A}}(\boldsymbol{\theta})\delta(\boldsymbol{\theta}^{(t-1)} - \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\
&= \min\left(q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})\pi(\boldsymbol{\theta}^{(t-1)}), q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})\right) \\
&\qquad\qquad\qquad + p_{\mathrm{A}}(\boldsymbol{\theta})\delta(\boldsymbol{\theta}^{(t-1)} - \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\
&= \min\left(\frac{q(\boldsymbol{\theta}|\boldsymbol{\theta}^{(t-1)})\pi(\boldsymbol{\theta}^{(t-1)})}{q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})}, 1\right) q(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\
&\qquad\qquad\qquad + p_{\mathrm{A}}(\boldsymbol{\theta})\delta(\boldsymbol{\theta}^{(t-1)} - \boldsymbol{\theta})\pi(\boldsymbol{\theta}) \\
&= p_{\mathrm{MH}}(\boldsymbol{\theta}^{(t-1)}|\boldsymbol{\theta})\pi(\boldsymbol{\theta})
\end{aligned} \tag{150}$$

We have demonstrated that the detailed balanced equation, therefore metropolis-Hasting algorithm will converge to stationary distribution.

7.6 Generating mixture of normal using student-t distribution

Here is a little example of how Metropolis-Hasting sampling, firstly we want to generate a mixture of two Gaussian density. Directly we may not directly sample it from the distribution, i.e.(transformation of variable or straight MCMC sampling), however, we can Metropolis-Hastings to sampled, the procedure is following the algorithm above. Notice, the difficulty of the Metropolis-Hastings Algorithm is choosing the appropriate proposal(see reference paper). Now if we want to generate samples from mixture of normal distribution $\pi(\theta) = \rho\mathcal{N}(x|\mu_1, 1) + (1-\rho)N(x|\mu_2, 1)$, and we choose student-t distribution as our transition density, but we assume each samples is independent from the proposal distribution, i.e. $q(x_t|x_{t-1}) = q(x_t) = 1/(1+t^2)$, and repeat the iteration until it converge the stationary distribution, we can obtain the following figure:

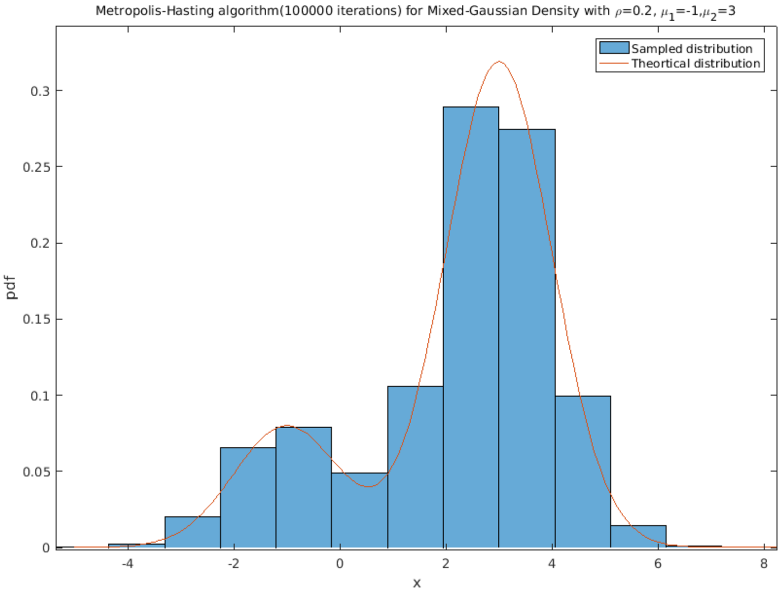

Figure 13: Samples generate from Metropolis-Hastings algorithm for mixtures of Gaussian density. Compared two distribution we found they are in agreement.

7.7 Convergence of the approximated posterior in the particle filter

Theorem 1: If the required density is proportional to the posterior $p(y|x)p(x)$ and we are able to draw samples $x^i \sim p(x)$, where $i = 1, \cdots, M$. If $p(y|x)$ is tractable, then the discrete distribution over x^k with probability mass of $p(y|x^k)/\sum_{i=1}^{M} p(y|x^i)$ on x^k tends in distribution to the required density as $M \to \infty$.

To prove it , true posterior density and cumulative distribution function is defined as following:

$$p(x|y) = \frac{p(y|x)p(x)}{p(y)} \text{ and } F(x|y) \tag{151}$$

We further define the cumulative distribution function for the approximating empirical distribution function as:

$$\hat{\mathbf{F}}_M(y|x) = \frac{\frac{1}{N}\sum_{i=1}^{M} p(y|x^i)\mathbb{1}_{(x>x_i)}}{\frac{1}{N}\sum_{i=1}^{M} p(y|x^i)}, \tag{152}$$

where $\mathbb{1}$ is an indicator function, which only have return values if the conditions $(\cdot)$ is satisfied, for this case:

$$\mathbb{1}(x > x_i) := \begin{cases} 1 & \text{if } \mathrm{x} > \mathrm{x_i}, \\ 0 & \text{if } \mathrm{x} \leq \mathrm{x_i}. \end{cases} \tag{153}$$

As $N \to \infty$, use the property of margin probability $p(y) = \int p(y|x)dx$ and the condition $x \sim p(x)$, we know the summation of the likelihood $\frac{1}{M}\sum_{i=1}^{M} p(y|x^i)$ will converge to $p(y)$ in probability, from the law of the large number, similarly we can also obtain $\frac{1}{N}\sum_{i=1}^{M} p(y|x^i)\mathbb{1}_{(x>x_i)}$ converges to $F(x|y)P(y)$ in probability as $N \to \infty$. Therefore we have $\hat{\mathbf{F}}_M(y|x) \to F(y|x)$ in probability.

8 Self-assessment

Since my supervisor is very busy(less than 3 hours meeting time in total), I treasure all the meeting time. Every week I produce a weekly report that can be easily followed, writing Matlab algorithm to check whether I understand the material from the paper and also compare the result. This enable me keep employed and learning, at some stage I can check whether I understand what I am doing, but certainly, I am not an expert and still may make some sorts of mistakes. I wrote down all single steps at the paper and take notes I don't understand to meeting. This project have a good amount of the code and please do not under-thinking the time it will take.

References

[1] Alessio Farhadi & Dimitri Vvedensky, Risk, randomness, crashes and quants, Contemporary Physics 44:3 237-257, DOI: 10.1080/0010751031000077396

[2] On sequential Monte Carlo sampling methods for Bayesian filtering A Doucet, S Godsill, C Andrieu, Statistics and computing 10 (3), 197-208

[3] Durbin, James, and Siem Jan Koopman. ”Time series analysis by state space methods” Vol. 38. Oxford University Press, 2012.

[4] Edward Ionides, Sequential Monte Carlo methods for inferring transmission dynamics from pathogen genetic sequences, Mathematisches Forschungsinstitut Oberwolfach workshop on Design and analysis of infectious disease studies, Nov. 13, 2013

[5] Anna M. and Paul S., scribe ”Filtering. State space models. Kalman filter.”, MIT, `https://ocw.mit.edu/courses/economics/14-384-time-series-analysis-fall-2013/lecture-notes/MIT14_384F13_lec21.pdf`

[6] Schön, Thomas B., Adrian Wills, and Brett Ninness. ”System identification of nonlinear state-space models.” Automatica 47.1 (2011): 39-49.

[7] Matei, Marius. ”Assessing volatility forecasting models: why GARCH models take the lead.” Romanian Journal of Economic Forecasting 12.4 (2009): 42-65.

[8] Schaefer, Stephen M. “Robert Merton, Myron Scholes and the Development of Derivative Pricing.” The Scandinavian Journal of Economics, vol. 100, no. 2, 1998, pp. 425–445. JSTOR, JSTOR, www.jstor.org/stable/3440891.

[9] Nelson, Daniel B. ”Conditional heteroskedasticity in asset returns: A new approach.” Econometrica: Journal of the Econometric Society (1991): 347-370.

[10] Taylor, Stephen J. ”Modelling financial time series”. world scientific, 2008.

[11] Faragher, Ramsey. ”Understanding the basis of the Kalman filter via a simple and intuitive derivation.” IEEE Signal processing magazine 29.5 (2012): 128-132.

[12]] Bishop, Gary, and Greg Welch. ”An introduction to the Kalman filter.” Proc of SIGGRAPH, Course 8.27599-3175 (2001): 59.

[13] Harvey, Andrew C. ”Time Series Model: Second Edition”, Harvester Wheatsheaf, ISBN: 0-7450-1199-3

[14] Platanioti, K., E. J. McCoy, and D. A. Stephens. A review of stochastic volatility: univariate and multivariate models. working paper, 2005.

[15] Nasrabadi, Nasser M. ”Pattern recognition and machine learning.” Journal of electronic imaging 16.4 (2007): 049901.

[16] Wackerly, Dennis; Mendenhall, William; Scheaffer, Richard L. (2008). ”Mathematical Statistics with Applications (7 ed)” Belmont, CA, USA: Thomson Higher Education. ISBN 0-495-38508-5.

[17] Kim, Sangjoon, Neil Shephard, and Siddhartha Chib. ”Stochastic volatility: likelihood inference and comparison with ARCH models.” The review of economic studies 65.3 (1998): 361-393.

[18] Gamerman, Dani, and Hedibert F. Lopes. Markov chain Monte Carlo: stochastic simulation for Bayesian inference. Chapman and Hall/CRC, 2006.

[19] Chib, Siddhartha, and Edward Greenberg. ”Understanding the metropolis-hastings algorithm.” The american statistician 49.4 (1995): 327-335.

[20] Metropolis, Nicholas, et al. ”Equation of state calculations by fast computing machines.” The journal of chemical physics 21.6 (1953): 1087-1092.

[21] Gelman, Andrew, and Donald B. Rubin. ”Inference from iterative simulation using multiple sequences.” Statistical science 7.4 (1992): 457-472.

[22] Pitt, Michael K., and Neil Shephard. ”Analytic convergence rates and parameterization issues for the Gibbs sampler applied to state space models.” Journal of Time Series Analysis 20.1 (1999): 63-85.

[23] Yu, Yaming, and Xiao-Li Meng. ”To center or not to center: That is not the question—an Ancillarity–Sufficiency Interweaving Strategy (ASIS) for boosting MCMC efficiency.” Journal of Computational and Graphical Statistics 20.3 (2011): 531-570.

[24] , Doucet, Arnaud & Johansen, Adam. (2009). A Tutorial on Particle Filtering and Smoothing: Fifteen Years Later. Handbook of Nonlinear Filtering. 12.

[25] N. J. Gordon, D. J. Salmond and A. F. M. Smith, ”Novel approach to nonlinear/non-Gaussian Bayesian state estimation,” in IEE Proceedings F - Radar and Signal Processing, vol. 140, no. 2, pp. 107-113, April 1993. doi: 10.1049/ip-f-2.1993.0015

[26] Pitt, Michael K., and Neil Shephard. “Filtering via Simulation: Auxiliary Particle Filters.” Journal of the American Statistical Association, vol. 94, no. 446, 1999, pp. 590–599. JSTOR, JSTOR, www.jstor.org/stable/2670179.

[27] Michael K. Pitt, Ralph dos Santos Silva, Paolo Giordani, Robert Kohn, On some properties of Markov chain Monte Carlo simulation methods based on the particle filter, Journal of Econometrics, Volume 171, Issue 2, 2012, Pages 134-151, ISSN 0304-4076, https://doi.org/10.1016/j.jeconom.2012.06.004. (http://www.sciencedirect.com/science/article/pii/S0304407612001510) Keywords: Auxiliary variables; Adapted filtering; Bayesian inference; Simulated likelihood)