A Simulation Study comparing MCMC, QML and GMM Estimation of the Stochastic Volatility Model

Carl Nilsson
carl.nilsson.949@student.lu.se

supervised by
Peter Jochumzen

October 13, 2016
Abstract

The stochastic volatility (SV) model is an alternative to GARCH models to model time varying volatility. In this thesis the basic stochastic volatility model and three different estimation methods are described—namely, Bayesian Markov chain Monte Carlo (MCMC) methods, quasi maximum-likelihood (QML) and generalized method of moments (GMM).

To compare these estimation methods a large scale simulation study is conducted where many different parameter values and sample sizes are compared. Since both the latter two methods are non-likelihood based, our hypothesis is that the likelihood based MCMC would perform better. The conclusion of the study is that this is the case, MCMC turns out to be more efficient than QML and GMM by quite a large margin, especially for estimating the latent volatilities.

Keywords: Monte Carlo simulation, simulation study, stochastic volatility, Markov chain Monte Carlo, quasi-maximum likelihood, generalized method of moments, Gibbs sampling, Metropolis-Hastings
## Contents

1 Introduction  
   1.1 Stochastic Volatility Models  
      1.1.1 Basic properties  
      1.1.2 State space form  
      1.1.3 Estimation  
   1.2 Purpose  
   1.3 Software Used  
   1.4 Outline  

2 Estimation Methods for the SV model  
   2.1 Markov Chain Monte Carlo  
   2.2 Quasi Maximum Likelihood  
   2.3 Generalized Method of Moments  
   2.4 Results  

3 A Simulation Study Comparing MCMC to QML and GMM  
   3.1 Parameter estimates  
   3.2 Volatility estimates  
   3.3 Conclusions and ideas for further research  

References  

A Bayesian Methods  
   A.1 Priors  

B Simulation Methods  
   B.1 Accept-Reject Method  
   B.2 Monte Carlo Integration  
   B.3 Importance Sampling  
   B.4 Markov Chains  
   B.5 The Metropolis-Hastings Algorithm  
   B.6 Gibbs Sampling  
      B.6.1 The two-stage Gibbs sampler  
      B.6.2 The multistage Gibbs sampler  

C Probability Distributions  
   C.1 Normal  
   C.2 Gamma  
   C.3 Inverse Gamma  
   C.4 Beta  

iii
List of Figures

2.1 Simulated data from the SV model with $T = 1000$, and parameters $\mu = -9$, $\phi = 0.97$ and $\sigma = 0.15$. The upper frame plots the simulated logreturns $\{y_t\}$ and the lower frame the simulated conditional volatilities $\{\sigma_t\}$. .......................... 7
2.2 The prior density $f(\mu)$ ................................................................. 9
2.3 The prior density $f(\phi)$ for $\alpha = 35$ and $\beta = 1.5$ .......................... 9
2.4 The prior density $f(\sigma^2)$. ......................................................... 10
2.5 The density of log $\chi^2$ versus the mixture of ten normals. ................. 12
2.6 Trace and density plots of 100,000 posterior draws after 25,000 discarded burn-in sweeps for the parameters $\mu$, $\phi$ and $\sigma$ for the simulated series with $\mu = -9$, $\phi = 0.97$ and $\sigma = 0.15$ .............................. 14
2.7 The posterior means of $\sigma_t$ (with a 90% posterior credible interval) from the MCMC algorithm compared to the simulated volatilities. ............................. 15
2.8 True log $(\chi^2)$ density vs normal approximation $\mathcal{N}(-1.27036, \pi^2/2)$ ... 16
2.9 QML and Kalman filter estimates of the simulated volatilities with a 90% confidence interval. ................................................................. 17
2.10 Kalman filter using the parameter estimates from GMM for the simulated series with a 90% confidence interval. ................................. 21
2.11 The simulated volatilities compared to the estimates from MCMC, QML and GMM. ................................................................. 22
B.1 The $\mathcal{B}(2.7, 6.3)$ density. .......................................................... 40
B.2 Generation of $X \sim \mathcal{B}(2.7, 6.3)$ with Accept-Reject sampling. ........... 42
B.3 Example of a Monte Carlo integration. .............................................. 44
B.4 Metropolis-Hastings algorithm, $\mathcal{B}$ example – Drawings .................. 47
B.5 Metropolis-Hastings algorithm, $\mathcal{B}$ example – Density compared with direct simulation ................................................................. 48
B.6 Gibbs sampler of the bivariate normal $(\rho = 0.5)$ with 50 steps and no burn-in starting from $(2, -2)$ ...................................................... 50

List of Tables

2.1 The weights, means and variances $\{q_k, m_k, v^2_k\}$ of the normal mixtures from Omori, Chib, Shephard, and Nakajima (2007). .................. 12
2.2 Summary of the estimates of the simulated SV series. ....................... 22
3.1 Parameter specification ................................................................. 23
3.2 Estimated means and RMSE for $T = 500$ ..................................... 24
3.3 RMSE ratio for $T = 500$ ............................................................... 25
3.4 Estimated means and RMSE for $T = 2500$ ..................................... 26
3.5 RMSE ratio for $T = 2500$ .......................... 26
3.6 Estimated means and RMSE for $T = 5000$ ......................... 27
3.7 RMSE ratio for $T = 5000$ .................................. 27
3.8 The change in RMSE($\hat{\theta}$) as $T$ grows. ......................... 28
3.9 RMSE($\hat{\sigma}_t$) for $T = 500$ ................................ 29
3.10 RMSE($\hat{\sigma}_t$) for $T = 2500$ ............................... 29
3.11 RMSE($\hat{\sigma}_t$) for $T = 5000$ ............................... 30
3.12 RMSE($\hat{\sigma}_t$) ratio .......................................... 30

List of Algorithms

2.1 SV mixture sampler. ............................................. 12
2.2 R function for the SV moment conditions. ......................... 20
2.3 Fit the SV model using the R package gmm. ....................... 20
B.1 Generic R code for rejection sampling .......................... 39
B.2 R code for rejection sampling – beta example .................... 41
B.3 R code for Monte Carlo integrating $g(x)$ in (B.6) .................. 43
B.4 Metropolis-Hastings ............................................. 46
B.5 R code for M-H algorithm – beta example ....................... 47
B.6 The Gibbs sampler. .............................................. 50
1 Introduction

There are some properties that are common for most or almost all types of financial return series; thorough reviews of these stylized facts are given by Taylor (2005), Campbell, Lo, and MacKinlay (1997), Bollerslev, Engle, and Nelson (1994), Ding, Granger, and Engle (1993) and Jorion and Goetzmann (1999).

The empirical distributions display fat tails (leptokurtosis) and skewness; the returns have little (small markets) or no autocorrelation (larger markets) whereas squared returns are highly autocorrelated. One way to model financial return series that takes these stylized facts into account is by the autoregressive conditional heteroskedasticity (ARCH) model of Engle (1982) and the generalized ARCH (GARCH) model of Bollerslev (1986) and their extensions (see Bollerslev et al., 1994 for a survey).

Let \( y_t \) be a random variable that has a mean and variance conditional on the information set \( \mathcal{F}_{t-1} \), i.e., all information available at time \( t-1 \). For example, the series \( \{y_t\} \) may be the error term from an econometric model: 

\[
y_t = r_t - \mu_t,
\]

where \( \mu_t \equiv E[r_t | \mathcal{F}_{t-1}] \). We could assume 

\[
y_t | \mathcal{F}_{t-1} \sim N(0, \sigma^2_t),
\]

which is the standard case, or more generally 

\[
y_t | \mathcal{F}_{t-1} \sim D(0, \sigma^2_t),
\]

where \( D \) stands for distribution and could be a leptokurtic distribution, e.g., Student’s t-distribution.

A GARCH\((p,q)\) process is formulated as 

\[
\sigma^2_t = \omega + \alpha_1 y^2_{t-1} + \cdots + \alpha_q y^2_{t-q} + \beta_1 \sigma^2_{t-1} + \cdots + \beta_p \sigma^2_{t-p} \\
= \omega + \sum_{i=1}^q \alpha_i y^2_{t-i} + \sum_{j=1}^p \beta_j \sigma^2_{t-j} \quad (1.1)
\]

where \( \omega > 0 \) and \( \alpha_i \geq 0 \) for \( i = 1, \ldots, q \), \( \beta_j \geq 0 \), for \( j = 1, \ldots, p \), which is necessary and sufficient for \( \sigma^2_t \) to be positive. If \( p \) is equal to zero, the GARCH\((p,q)\) process reduces to an ARCH\((q)\) process and if both \( p \) and \( q \) are equal to zero, \( y_t \) is just a white noise and the conditional variance \( \sigma^2_t \) is constant and equal to \( \omega \).

The GARCH\((1,1)\)—also proposed independently by Taylor (1986) (who called it an ARMACH process)—is the most widely employed ARCH/GARCH model in practice,\(^1\) and is simply stated as 

\[
\sigma^2_t = \omega + \alpha y^2_{t-1} + \beta \sigma^2_{t-1}. \quad (1.2)
\]

Another way to model financial time series is by Stochastic Volatility (SV) models, which is the focus of this thesis and will be described in the next section. In the volatility equation of the SV model (equivalent to (1.1) for GARCH) an additional white noise error term is included. This makes them more realistic and gives them a better ability to capture stylized facts of empirical data, although

---

\(^1\) See, for example, Engle (2004, p. 330), Kennedy (2010, p. 125) and Teräsvirta (2009, p. 20).
much harder to estimate (see, e.g., Kim, Shephard, & Chib, 1998). One field where SV models have been used extensively and successfully is for pricing financial derivatives, although, almost exclusively with continuous specifications of the processes. However, it is often argued that a discrete SV representation is a better approximation to the continuous form model than a GARCH representation.\(^2\)

### 1.1 Stochastic Volatility Models

This section introduces the discrete time Stochastic Volatility (SV) class of models.\(^3\) If a very broad definition is used, GARCH models could be considered a subclass of the SV models. However, in this thesis—and in what seems to be the norm in the SV literature\(^4\)—a narrower definition is employed, and a distinction between GARCH and SV models is made. The difference is that the SV model includes an additional error term, specified in the volatility equation, which is unobservable (non-measurable), thus making the process latent and estimation non-trivial.

Early references includes Clark (1973), Taylor (1982), Rosenberg (1972) (all reprinted in Shephard, 2005, which also includes a lengthy and up to date introduction to SV models by the editor) and Taylor (1986) (recently released in a second edition with a new preface: Taylor, 2008). Other introductions may be found in Taylor, 2005 and in Andersen, Davis, Kreiß, and Mikosch (2009). See also Taylor (1994), Shephard (1996) and Ghysels et al. (1996) for surveys on SV models and their statistical properties.

The basic log-normal SV model models the (unobserved) conditional variance of \(\{y_t\}\) as a log-AR(1) series, for \(t = 1, \ldots, T\), as

\[
\begin{align*}
y_t &= \sigma_t \varepsilon_t, \quad \text{with} \quad \varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1) \\
h_t &= \mu + \phi(h_{t-1} - \mu) + \sigma_{\eta} \eta_t, \quad \text{with} \quad \eta_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1), \\
h_0 &\sim \mathcal{N}\left(\mu, \frac{\sigma_{\eta}^2}{1 - \phi^2}\right)
\end{align*}
\]  

where \(\varepsilon_t\) and \(\eta_t\) are independent white noise processes, \(h_t \equiv \log \sigma_t^2\), and \(h_0\) is drawn from the stationary distribution of \(h_t\). The return series \(y_t\) is stationary and ergodic (see Section B.4) if the persistence parameter \(\phi\) is restricted to \(|\phi| < 1\), and the volatility of log-volatility parameter \(\sigma_{\eta}\) is restricted to \(\sigma_{\eta} > 0\).

---

\(^2\)See for example Melino and Turnbull (1990) and Wiggins (1987).

\(^3\)Continuous time SV models are used extensively in finance starting with Hull and White (1987). For a connection between discrete and continuous SV models, see, e.g., Ghysels, Harvey, and Renault (1996) or Shephard (2005).

\(^4\)See the papers and books cited below.
1.1.1 Basic properties

Since $\eta_t$ is normal, $h_t$ follows a normal AR(1) process with

$$E[h_t] = \mu, \quad \text{Var}(h_t) = \sigma_h^2 = \frac{\sigma^2}{1-\phi^2}.$$  

Using the properties of the log-normal distribution\(^5\), if $h_t$ is stationary all moments exist and for any positive number $r$ (Taylor, 2005, (11.7), (11.26) and (11.27)):

$$E[|y_t|^r] = E[(\sigma_t|\varepsilon_t|)^r] = E[\sigma_t^r] E[|\varepsilon_t|^r]$$ \hspace{1cm} (1.4)

where

$$E[\sigma_t^r] = \exp\left(\frac{r}{2}\mu + \frac{r^2}{8}\sigma_h^2\right)$$ \hspace{1cm} (1.5)

and

$$E[|\varepsilon_t|^r] = \frac{2^{r/2}}{\sqrt{\pi}} \Gamma\left(\frac{r}{2} + 1\right)$$ \hspace{1cm} (1.6)

where $\Gamma()$ is the gamma function\(^6\) and if $r$ is a positive integer

$$\Gamma(r) = (r - 1)!$$

From which we can deduce that

$$\text{Var}(y_t) = E[\sigma_t^2\varepsilon_t^2] = E[\sigma_t^2] E[\varepsilon_t^2] = E[\sigma_t^2] = e^{\mu + \sigma_h^2/2}$$ \hspace{1cm} (1.7)

and calculate the kurtosis

$$\text{kurt}(y_t) = \frac{E[y_t^4]}{E[\sigma_t^4]^2} = \frac{3e^{2\mu + 2\sigma_h^2}}{e^{2\mu + \sigma_h^2}} = 3e^{\sigma_h^2} > 3,$$ \hspace{1cm} (1.8)

meaning that the SV model has fatter tails than the corresponding normal distribution.

The squared returns $y_t^2$ are positively autocorrelated. To calculate the autocorrelations we first note that (Shephard, 1996, p. 23)

$$E[y_t^2 y_{t-s}^2] = E[\sigma_t^2\sigma_{t-s}^2] E[\varepsilon_t^2] E[\varepsilon_{t-s}^2] = E[\sigma_t^2\sigma_{t-s}^2] = e^{2\mu + \sigma_h^2 + 2\sigma_h^2\phi^s}.$$  

\(^5\) If $\log X \sim \mathcal{N}(\mu, \sigma^2)$, then $X \sim \log\mathcal{N}(\mu, \sigma^2)$, $E[X^r] = \exp(r\mu + r^2\sigma^2/2)$, $\text{Var}(X) = e^{2\mu + \sigma^2}(e^{\sigma^2} - 1)$. So, if $h_t = \log \sigma_t^2 \sim \mathcal{N}(\mu, \sigma_h^2)$, then $\sigma_t^2 \sim \log\mathcal{N}\left(\frac{\mu}{2}, \frac{\sigma_h^2}{2}\right)$.

\(^6\) See (C.5) for a definition.
The autocovariances are given by
\[
\text{Cov}(y_t^2, y_{t-s}^2) = E[y_t^2 y_{t-s}^2] - E[y_t^2] E[y_{t-s}^2]
\]
\[
= e^{2\mu + \sigma_h^2 + \sigma_h^2 \phi^s} - e^{2\mu + \sigma_h^2}
\]
\[
= e^{2\mu + \sigma_h^2} (e^{\sigma_h^2 \phi^s} - 1)
\]
then, the autocorrelations of \( y_t^2 \) can be written as
\[
\frac{\text{Cov}(y_t^2, y_{t-s}^2)}{\text{Var}(y_t^2)} = \frac{e^{2\mu + \sigma_h^2} (e^{\sigma_h^2 \phi^s} - 1)}{e^{2\mu + \sigma_h^2} (3e^{\sigma_h^2} - 1)} = \frac{e^{\sigma_h^2 \phi^s} - 1}{3e^{\sigma_h^2} - 1}.
\]
which can be negative, unlike for GARCH models, if \( \phi < 0 \).

1.1.2 State space form

Harvey, Ruiz, and Shephard (1994) square and take the logarithm of the mean equation of the SV model (1.3), \( y_t = \sigma_t \varepsilon_t \), so that
\[
\log y_t^2 = h_t + \log \varepsilon_t^2
\]
\[
= E[\log \varepsilon_t^2] + h_t + \xi_t
\]
(1.9)
where \( \xi_t \) is a zero-mean non-normal white noise process defined as \( \xi_t = \log \varepsilon_t^2 - E[\log \varepsilon_t^2] \). Since we have assumed \( \varepsilon_t \sim \mathcal{N}(0, 1) \), then \( \log \varepsilon_t^2 \sim \log \chi_1^2 \), with pdf (see Omori, Chib, Shephard, & Nakajima, 2007, p. 428)
\[
f(\log \varepsilon_t^2) = \frac{1}{\sqrt{2\pi}} \exp \left\{ \frac{\log \varepsilon_t^2 - \log \varepsilon_t^2}{2} \right\}, \quad \log \varepsilon_t^2 \in \mathbb{R}
\]
with \( E[\log \varepsilon_t^2] \approx -1.27036 \) and \( \text{Var}(\log \varepsilon_t^2) = \pi^2/2 \). (1.3) can then be written as a (non-normal) linear state-space model
\[
\log y_t^2 = 1.27036 + h_t + \xi_t, \quad \text{with } \xi_t \overset{\text{i.i.d.}}{\sim} \left(0, \frac{\pi^2}{2}\right)
\]
\[
h_t = \mu + \phi(h_{t-1} - \mu) + \sigma_\eta \eta_t, \quad \text{with } \eta_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1).
\]
(1.10)

1.1.3 Estimation

One of the limitations of SV models is that the likelihood function
\[
l(y_1, \ldots, y_T; \theta) = \int_{h_1} \cdots \int_{h_T} f(y_1, \ldots, y_T|h_1, \ldots, h_T, \theta)f(h_1, \ldots, h_T|\theta) \, dh_1 \cdots dh_T
\]
(1.11)
unlike for GARCH models, is intractable. To evaluate the $T$ dimensional integral we have to either use numerical methods or some kind of approximation, or find another way to estimate the SV model. In this thesis we will try three different approaches:

MCMC We use simulation methods, namely Markov chain Monte Carlo (MCMC) methods to estimate the model by taking a Bayesian point of view.

QML We rewrite $y_t = \sigma_t \epsilon_t$ in state space form, i.e., $\log y_t^2 = h_t + \log \epsilon_t^2$, and maximize the (quasi) likelihood function under the (false) assumption of normality.

GMM Finally, we estimate the parameters of the SV model $\theta$ using generalized methods of moments (GMM).

1.2 Purpose

The main purpose of this thesis is to conduct a large scale simulation study, comparing three different estimation methods for the stochastic volatility model, and how they perform for different parameter specifications and sample sizes. We will compare both parameter estimates and estimates of the latent (non-observable) volatilities.

Hypothesis Our hypothesis is that Markov chain Monte Carlo methods is the most efficient estimator of the SV model of the three compared, and in case the hypothesis is true, we are interested in learning how much more efficient it is compared to QML and GMM.

1.3 Software Used

The main program/programming language used for this thesis is the open source and multi-platform R language (R Core Team, 2016). I have tried to include as much of the code as possible and as deemed necessary, all code is available from the author on request. Mathematica was used for some calculations.

This thesis was typeset in \LaTeX, an extension of Donald Knuth’s TeX by Leslie Lamport.

\footnote{http://www.r-project.org.}
1.4 Outline

The rest of the thesis is outlined as follows. In Section 2 we describe the estimation methods for the SV model. Section 3, which is the main part of the thesis, conducts a simulation study comparing the finite sample estimates of MCMC, QML and GMM for estimating the basic SV model.

An appendix is included with reviews of some useful concepts. Appendix A gives a very brief introduction to Bayesian statistics and in particular Bayes’ formula and concepts such as prior and posterior distributions. Appendix B defines and describes some simulation methods that will be useful for MCMC estimation of the SV model, such as rejection sampling, the Metropolis-Hastings algorithm and Gibbs sampling. Appendix C is a review of a few common probability distributions used in the thesis.
2 Estimation Methods for the SV model

This section gives a brief introduction to the estimation methods that will be used to estimate the stochastic volatility model in the simulation study in Section 3. First we use the R package stochvol (Kastner, 2016a, 2016b) to simulate a series of length $T = 1000$ from the SV model in (1.3), with $\mu = -9$, $\phi = 0.97$ and $\sigma_\eta = 0.15$ that will be estimated using QML, GMM and MCMC throughout this section, see Figure 2.1.

```r
set.seed(1)  # set the seed to ensure reproducibility
library(stochvol)
sim <- svsim(1000, mu = -9, phi = 0.97, sigma = 0.15)
```

![Simulated data from the SV model](image)

**Figure 2.1** Simulated data from the SV model with $T = 1000$, and parameters $\mu = -9$, $\phi = 0.97$ and $\sigma_\eta = 0.15$. The upper frame plots the simulated logreturns $\{y_t\}$ and the lower frame the simulated conditional volatilities $\{\sigma_t\}$.

2.1 Markov Chain Monte Carlo

Shephard, 2002 for \( t \)-distributed errors and jumps, Chib, Nardari, & Shephard, 2006 for multivariate SV models and Omori et al., 2007 for correlation between the error terms) takes a different approach using a mixture of normal distributions to estimate the reformulated \( \log y_t^2 \) that allows them to use a multi-move algorithm, which is more efficient.

The difficulty in estimating the SV model described in Section 1.1, i.e.,

\[
\begin{align*}
y_t &= \sigma_t \varepsilon_t, \quad \text{with } \varepsilon_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1) \\
h_t &= \mu + \phi (h_{t-1} - \mu) + \sigma_\eta \eta_t, \quad \text{with } \eta_t \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, 1), \\
h_0 &\sim \mathcal{N}\left(\mu, \sigma^2_\eta \frac{1}{1 - \phi^2}\right)
\end{align*}
\]

with Bayesian and Markov chain Monte Carlo methods (see Appendix A and B for a short introduction to Bayesian and MCMC methods, respectively) lies in that the likelihood function is intractable. This makes it impossible to directly analyze the posterior density \( \pi(\theta | y) \). However, this can be overcome by using data augmentation (Tanner & Wong, 1987), augmenting the parameters \( \theta = (\mu, \phi, \sigma_\eta^2) \) with the latent log volatilities \( h = (h_1, \ldots, h_T)^T \) to form a large parameter vector of length \( 3 + T \), \( (\theta, h) \in \Theta \times \mathcal{H} \). The joint posterior is given by Bayes theorem:

\[
\pi(\theta, h | y) \propto f(y | h)f(h | \theta)f(\theta).
\]

which can be sampled with MCMC methods without computing the likelihood \( f(y | \theta) \). From the joint posterior, the marginal \( \pi(\theta | y) \) can be used to perform inference on the parameters of the SV model, while the marginal \( \pi(h | y) \) can be used for inference on the unknown volatilities and for prediction. To compute these marginal distributions a Markov chain is constructed with invariant distribution \( \pi \) (see Section B.4).

**Prior distributions** To perform Bayesian inference we need to specify a prior distribution\(^8\) \( f(\theta) \), and following Kim et al. (1998), independent priors are chosen \( f(\mu, \phi, \sigma_\eta^2) = f(\mu)f(\phi)f(\sigma_\eta^2) \). A normal (see Section C.1) prior is employed for the level \( \mu \in \mathbb{R} \), with the hyperparameters chosen so that it is rather uninformative

\[
\mu \sim \mathcal{N}(0, 100).
\]

See Figure 2.2.

For the persistence \( \phi \in (-1, 1) \), a beta (see Section C.4) prior is chosen

\[
\frac{\phi + 1}{2} \sim \mathcal{B}(\alpha, \beta),
\]

\(^8\) See Section A.1.
implying
\[ f(\phi) = \frac{1}{2B(\alpha, \beta)} \left( \frac{\phi + 1}{2} \right)^{\alpha-1} \left( \frac{1 - \phi}{2} \right)^{\beta-1}, \quad \alpha, \beta > \frac{1}{2} \] (2.4)
such that \( \phi \) has support on the interval \((-1,1)\), the prior mean and variance are \( E[\phi] = 2\alpha/(\alpha + \beta) - 1 \) and \( \text{Var}(\phi) = 4\alpha\beta/((\alpha + \beta)^2(\alpha + \beta + 1)) \). We set \( \alpha = 35 \) and \( \beta = 1.5 \), implying a prior mean \( E[\phi] = 0.92 \) and standard deviation \( \sqrt{\text{Var}(\phi)} = 0.065 \), and very little weight for values of \( \phi < 0.5 \) (actually \( \Pr(\phi < 0.90) = 30\% \) and \( \Pr(\phi < 0.70) = 1\% \)). See Figure 2.3. This is done since the choice of hyperparameters is rather influential in small samples (Kastner, 2016a), and we use the prior information that for daily financial data series \( \phi \) has been found to be around 0.90 to 0.99 (see Jacquier et al., 1994, and references therein).

Instead of specifying a conjugate prior (see Subsection A.1) for \( \sigma_\eta \in \mathbb{R}^+ \), following Kastner and Frühwirth-Schnatter (2014) (citing Frühwirth-Schnatter &

\[ \text{I} \text{e., that it is inverse-gamma distributed (see Section C.3) } \sigma_\eta^2 \sim IG(a,b), \text{ as in e.g., Jacquier et al. (1994) and Kim et al. (1998).} \]
Wagner, 2010 for motivation), it is specified as the product of the hyperparameter $B_\sigma$ with a $\chi^2_1$ distribution which is a special case of the gamma distribution (see Section C.2)

$$\sigma^2_\eta \sim B_\sigma \times \chi^2_1 = G \left( \frac{1}{2}, \frac{B_\sigma}{2} \right)$$ (2.5)

$$f(\sigma^2_\eta) = \frac{1}{\Gamma(\frac{1}{2})} \left( \frac{B_\sigma}{2} \right)^{1/2} \frac{1}{\sqrt{\sigma^2_\eta}} \exp \left( -\sigma^2_\eta \frac{B_\sigma}{2} \right)$$ (2.6)

The prior mean and variance are $E[\sigma^2_\eta] = 1/B_\sigma$ and $\text{Var}(\sigma^2_\eta) = 2/B^2_\sigma$, respectively.

We choose $B_\sigma = 1$, so that $\sigma^2_\eta \sim \chi^2_1$, $E[\sigma^2_\eta] = 1$ and $\text{Var}(\sigma^2_\eta) = 2$, see Figure 2.4.

![Figure 2.4](image)

**Figure 2.4** The prior density $f(\sigma^2_\eta)$.

**Single-move sampler** As mentioned above the MCMC analysis of the SV model started with Jacquier et al. (1994), and is based on $(T + 3)$ full conditional distributions

$$h_t | y, h_{\setminus t}, \theta, \quad \text{for } t = 1, 2, \ldots, T$$

$$\mu | y, h, \phi, \sigma^2_\eta$$

$$\phi | y, h, \mu, \sigma^2_\eta$$

$$\sigma^2_\eta | y, h, \mu, \phi$$

(2.7)

where $h_{\setminus t}$ denotes all elements of $h$ excluding $h_t$. The latent variables $h$ are sampled by a sequence of Metropolis-Hastings (see Section B.5) independence draws.

Shephard and Kim (1994) and Kim et al. (1998) notes that the single-move sampler converges very slowly since the components of $h|y, \theta$ are highly correlated, and develops another multi-move sampler which improves mixing in the Markov chain and is shown by Kim et al. (1998) to be more efficient.
**Multi-move sampler** The multi-move sampler builds on the state-space form of the SV model, \( y_t^* = h_t + \epsilon_t \) (where \( y_t^* = \log y_t^2 \) and \( \epsilon_t = \log \epsilon_t^2 \)) described in Section 1.1.2. Since we have assumed \( \epsilon_t \) to be normal then \( \epsilon_t = \log \epsilon_t^2 \) is \( \log \chi^2 \) distributed, with density

\[
f(\epsilon_t) = \frac{1}{\sqrt{2\pi}} \exp \left( \frac{\epsilon_t - \exp(\epsilon_t)}{2} \right), \quad \epsilon_t \in \mathbb{R}.
\]

Instead of approximating \( \log \epsilon_t^2 \) with a normal distribution \( \mathcal{N}(-1.2704, \pi^2/2) \) as in QMLE (see Figure 2.8), Kim et al. (1998) use a mixture representation where the density is in the form of a mixture of normal distributions

\[
g(\epsilon_t) = \sum_{k=1}^{K} q_k f_N(\epsilon_t | m_k, v_k^2), \quad \epsilon_t \in \mathbb{R} \tag{2.8}
\]

where \( q_k \) denotes the component probability of the \( k \)th normal density, and \( f_N(\epsilon_t | m_k, v_k^2) \) denotes the density of a normal distribution with mean \( m_k \) and variance \( v_k^2 \).

The mixture distribution could, equivalently, be written in terms of a component indicator variable \( s_t \in \{1, \ldots, K\} \),

\[
\epsilon_t | s_t = k \sim \mathcal{N}(m_k, v_k^2)
\]

\[
\Pr(s_t = k) = q_k \tag{2.9}
\]

which is the formulation Kim et al. (1998) uses in their MCMC mixture sampler.

The constants \( \{q_k, m_k, v_k^2\} \) are optimized to make \( g(\epsilon_t) \) match the first four moments of the \( \log \chi^2 \) density \( f(\epsilon_t) \) (these constants are model independent so there is no added computational cost). Kim et al. (1998) sets \( K = 7 \), while Omori et al. (2007) sets \( K = 10 \), see Table 2.1. As can be seen in Figure 2.5 the fit is very close to the true density. Compare with the normal approximation used in the QML estimation in Figure 2.8 (which is equivalent to setting \( K = 1 \) and approximating \( \epsilon_t \) with \( \mathcal{N}(-1.2704, \pi^2/2) \)). Given the indicator variable at each time \( t, s_t \), the state space form of the SV model is Gaussian

\[
y_t | s_t = k, h_t \sim \mathcal{N}(h_t + m_k, v_k^2) \tag{2.10}
\]

which makes it possible to construct a multi-move, two block sampler to improve mixing in the Markov chain,

\[
(\theta, h) | y^*, s
\]

\[
s | y^*, h, \theta \tag{2.11}
\]
Table 2.1 The weights, means and variances \( \{q_k, m_k, v^2_k\} \) of the normal mixtures from Omori et al. (2007).

<table>
<thead>
<tr>
<th>( k )</th>
<th>( q_k )</th>
<th>( m_k )</th>
<th>( v^2_k )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.00609</td>
<td>1.92677</td>
<td>0.11265</td>
</tr>
<tr>
<td>2</td>
<td>0.04775</td>
<td>1.34744</td>
<td>0.17788</td>
</tr>
<tr>
<td>3</td>
<td>0.13057</td>
<td>0.73504</td>
<td>0.26768</td>
</tr>
<tr>
<td>4</td>
<td>0.20674</td>
<td>0.02266</td>
<td>0.40611</td>
</tr>
<tr>
<td>5</td>
<td>0.22715</td>
<td>-0.85173</td>
<td>0.62699</td>
</tr>
<tr>
<td>6</td>
<td>0.18842</td>
<td>-1.97278</td>
<td>0.98583</td>
</tr>
<tr>
<td>7</td>
<td>0.12047</td>
<td>-3.46788</td>
<td>1.57469</td>
</tr>
<tr>
<td>8</td>
<td>0.05591</td>
<td>-5.55246</td>
<td>2.54498</td>
</tr>
<tr>
<td>9</td>
<td>0.01575</td>
<td>-8.68384</td>
<td>4.16591</td>
</tr>
<tr>
<td>10</td>
<td>0.00115</td>
<td>-14.65</td>
<td>7.33342</td>
</tr>
</tbody>
</table>

Figure 2.5 The density of \( \log \chi^2 \) versus the mixture of ten normals.

Kim et al. (1998) implements the multi-move mixture sampler as Algorithm 2.1 for the posterior density \( \pi(s, h, \phi, \sigma^2, \eta, \mu | y^*) \) following the implementation for general state space models by Carter and Kohn (1994) and Shephard (1994).

Algorithm 2.1 SV mixture sampler.

1. Initialize \( s = (s_1, \ldots, s_T)^\top, \phi, \sigma^2, \) and \( \mu \);
2. Sample \( h \) from \( h | y^*, s, \phi, \sigma^2, \mu \) with a Metropolis-Hastings step;
3. Sample \( s \) from \( s_t \sim \text{Pr} \{ s_t = k | y^*_t, h_t \} \propto q_k f_{N}(y^*_t | h_t + m_k, v^2_k) \);
4. Update \( \phi, \sigma^2, \eta, \mu \);
5. Goto 2.

The R package stochvol (Kastner, 2016a, 2016b) is used to estimate the MCMC
SV model, which use the algorithms described in Kastner and Frühwirth-Schnatter (2014).\footnote{The sampler of Kastner and Frühwirth-Schnatter (2014) builds on that of Kim et al. (1998) in Algorithm 2.1, except that they sample $h$ all without a loop (AWOL) to improve computational speed and that they sample the model both using a centered and non-centered parameterization in the same algorithm using ancillarity-sufficiency interweaving (Yu & Meng, 2011).}

We first draw 25 000 burn-in iterations that are discarded and then 100 000 draws of which we save every fifth. The \texttt{R} code used is:

```r
library(stochvol)
mcmc <- svsample(y = sim$y, draws=1e5, burnin = 25000, thinpara=5, thinlatent=5, priormu = c(0, 100), priorphi = c(35, 1.5), priorsigma = 1)
```

The posterior parameter draws are plotted in Figure 2.6 as well as their densities. The posterior draws from $h_1, h_2, \ldots, h_{1000}$ are rewritten as $\sigma_t = e^{h_t/2}$ for $\sigma_1, \sigma_2, \ldots, \sigma_{1000}$, the means are plotted in Figure 2.7 as well as a 90\% posterior credible interval (the Bayesian equivalent of a confidence interval). The parameter estimates (actually the means of the posterior draws) and a comparison to QML and GMM are found in Section 2.4.
Figure 2.6 Trace and density plots of 100 000 posterior draws after 25 000 discarded burn-in sweeps for the parameters $\mu$, $\phi$ and $\sigma_\eta$ for the simulated series with $\mu = -9$, $\phi = 0.97$ and $\sigma_\eta = 0.15$. 
2.2 Quasi Maximum Likelihood

Standard references for state space models are Harvey (1989) and Durbin and Koopman (2012); for a textbook level introduction, see Commandeur and Koopman (2007). The stochastic volatility model has been estimated using quasi maximum likelihood (QML) by Harvey et al. (1994) and Ruiz (1994).

Using the state space form of the SV model in (1.10) and approximating \( \xi_t \sim N(0, \pi^2/2) \), the parameters \( \theta = (\mu, \phi, \sigma_\eta^2) \in \mathbb{R} \times (-1, 1) \times \mathbb{R}_+ \) can be estimated by maximizing the quasi\(^{11}\) log-likelihood function

\[
\ell_Q(y|\theta) = -\frac{T}{2} \log(2\pi) - \frac{1}{2} \sum_{t=1}^{T} \left( \log F_t + \frac{v_t^2}{F_t} \right)
\]

(2.12)

where \( v_t = \log y_t^2 - h_t + 1.27036 \) is the one-step-ahead prediction error and \( F_t = \text{Var}(v_t) \) is its variance. As seen in Figure 2.8, \( \log \varepsilon_t^2 \) is poorly approximated by the normal distribution, which leads to inefficient estimates in small samples. However, QMLE is consistent and asymptotically normal (Taylor, 2005, p. 286).

The SV model can be estimated with QML methods in R with numerous different state space and Kalman filtering packages, see Tusell (2011) for a comprehensive comparison of the packages available for R. We tried \texttt{dlm} (Petris, 2010; Petris, Petrone, & Campagnoli, 2009) and \texttt{FKF} (Luethi, Erb, & Otziger, 2014) and found

\(^{11}\)Quasi since \( \xi_t \) is not actually normally distributed, and thus \( \ell(y|\theta) \) is not a log-likelihood function.
Figure 2.8 The true log($\chi^2_1$) density in blue and the normal approximation $\mathcal{N}(-1.27036, \frac{\pi^2}{2})$ in dashed red.

The latter to be much faster for estimating the SV model so it will be used for the simulation study in this thesis. The FKF package uses the following notation for the state space model:

\[
\begin{align*}
y_t &= c_t + Z_t \cdot \alpha_t + G_t \cdot \varepsilon_t, & \varepsilon_t &\sim \mathcal{N}(0, I_d) \quad \text{(measurement equation)} \\
\alpha_t &= d_t + T_t \cdot \alpha_{t-1} + H_t \cdot \eta_t, & \eta_t &\sim \mathcal{N}(0, I_m) \quad \text{(transition equation)} \\
\alpha_0 &\sim \mathcal{N}(a_0, P_0)
\end{align*}
\]

where $\alpha_t$ denotes the state variable. For the SV model in state space form $\alpha_t = h_t$, $c_t = -1.27$, $Z_t = 1$, $G_tG_t^\top = \pi^2/2$, $d_t = \mu(1-\phi)$, $T_t = \phi$, $H_tH_t^\top = \sigma^2_\eta$, $a_0 = \mu$ and $P_0 = \sigma^2_\eta/(1-\phi^2)$. We create an R function defining the SV model in state space form

```r
SVss = function (parm) {
  a0 = c(parm[1])
  dt = matrix(parm[1]*(1 - parm[2]))
  ct = matrix(-1.27)
  Tt = matrix(parm[2])
  Zt = matrix(1)
  HHt = matrix(parm[3]^2)
  GGt = matrix(pi^2/2)
  return (list(a0 = a0, P0 = P0, ct = ct, dt = dt, Zt = Zt, Tt = Tt,
                HHt = HHt))
}
```

and create an objective function

```r
library(FKF)
opt <- function(theta, yt) {
```

16
which we pass to `optim()` to get the QML estimates

```r
qml.fit <- optim(par = c(mu = -9, phi = 0.90, sigmaeta = 0.3), fn = objective, yt = matrix(log(sim$y^2), nrow = 1), hessian = TRUE, method = "L-BFGS-B", lower = c(-Inf, -0.9999, 0.001), upper = c(Inf, 0.9999, Inf))
```

The parameter estimates are then used to estimate $h$ by a Kalman filter with `fkf()`

```r
sp <- SVss(qml.fit$par)
QML.filter <- fkf(a0 = sp$a0, P0 = sp$P0, dt = sp$dt, ct = sp$ct, Tt = sp$Tt, Zt = sp$Zt, HHt = sp$HHt, GGt = sp$GGt, yt = rbind(log(sim$y^2)))
```

The Kalman filter of the estimated parameters is plotted in Figure 2.9. The parameter estimates and a comparison to MCMC and GMM are found in Section 2.4.

![Figure 2.9 QML and Kalman filter estimates of the simulated volatilities with a 90% confidence interval.](image)

### 2.3 Generalized Method of Moments

Generalized method of moments (GMM) was proposed by Hansen (1982), see also Hansen, Heaton, and Yaron (1996) and for the choice of the covariance matrix
see Newey and West (1987), Newey and West (1994) and Andrews (1991). The stochastic volatility model was estimated using GMM by Melino and Turnbull (1990) and further investigated by Jacquier et al. (1994) and Andersen and Sørensen (1996).

The advantage of a moment based approach is that the integration problem with evaluating the likelihood can be avoided. However, in general, a finite number of moments contains less information than the density so the estimation will be less efficient than for likelihood based methods, especially for estimation of the SV model since, as mentioned by Jacquier et al. (1994), the score function cannot be computed to suggest which moments should be used for GMM estimation. Another drawback is that GMM can be expected to work poorly when $\phi$ is close to unity (which is common for financial data) since it only works when $h_t$ is stationary (Shephard, 1996). Since GMM only provides estimates of the parameters $\theta$ but not of the latent volatilities $\{\sigma_t\}_{t=1,...,T}$ they have to be estimated using another method. To estimate the volatilities we use the parameter estimates from GMM and run a Kalman filter (as described in 2.2) with those to obtain the filtered estimates of the volatilities.

Let $m(\theta)$ denote the vector of analytical moments, $\overline{m}_T(\theta)$ the vector of sample moments and define $g_T(\theta) = \overline{m}_T(\theta) - m(\theta)$. The GMM estimator of $\theta$, $\hat{\theta}_T$ minimizes the distance between the sample moments and analytical moments over the parameter space $\Theta$

$$\hat{\theta}_T = \arg\min_{\theta \in \Theta} g_T(\theta)^\top W g_T(\theta)$$

where $W$ is a weighing matrix.

The optimal weighting matrix $W^*$ is defined as

$$W^* = \left( \lim_{T \to \infty} \text{Var} \left( \sqrt{T} g_T(\theta_0) \right) \equiv \Omega(\theta_0) \right)^{-1}$$

which can be estimated by an heteroskedasticity and auto-correlation consistent (HAC) matrix

$$\hat{\Omega}(\theta) = \sum_{s=-(n-1)}^{n-1} k_h(s) \hat{\Gamma}_s(\theta^*),$$

where $k_h(s)$ is a kernel, $h$ is the bandwidth which can be chosen using the procedures proposed by Newey and West (1987) and Andrews (1991),

$$\hat{\Gamma}_s(\theta^*) = \frac{1}{T} \sum_t g_T(\theta^*)^\top g_T(\theta^*)^\top$$

and $\theta^*$ is a convergent estimate of $\theta_0$.

The version of GMM used in this thesis is the iterative GMM of Hansen et al. (1996):
1. Compute $\theta^{(0)} = \arg\min_{\theta} g_T(\theta)^\top g_T(\theta)$;
2. Compute the HAC matrix $\hat{\Omega}(\theta^{(0)})$;
3. Compute $\theta^{(1)} = \arg\min_{\theta} g_T(\theta)^\top \left[ \hat{\Omega}(\theta^{(0)}) \right]^{-1} g_T(\theta)$;
4. If $\|\theta^{(0)} - \theta^{(1)}\| < tol$ go to 5., else $\theta^{(0)} = \theta^{(1)}$ and go to 2.;
5. Define the ITGMM estimator $\hat{\theta}$ as $\theta^{(1)}$.

where the heteroskedasticity and auto-correlation consistent (HAC) matrix $\hat{\Omega}(\theta)$ is defined as in Newey and West (1987). The bandwidth used is the Quadratic Spectral kernel proposed by Andrews (1991).

Following Andersen and Sørensen (1996), we use the 24 moments defined below (see also Section 1.1.1)

$m_1(\theta) = E[|y_t|] = \left(\frac{2}{\pi}\right)^{1/2} E[\sigma_t]$
$m_2(\theta) = E[y_t^2] = E[\sigma_t^2]$
$m_3(\theta) = E[|y_t|^3] = 2 \left(\frac{2}{\pi}\right)^{1/2} E[\sigma_t^3]$
$m_4(\theta) = E[y_t^4] = 3 E[\sigma_t^4]$
$m_{5:14}(\theta) = E[y_t y_{t-i}] = \frac{2}{\pi} E[\sigma_t \sigma_{t-i}], \quad \text{for } i = 1, \ldots, 10$
$m_{15:24}(\theta) = E[y_t^2 y_{t-i}^2] = E[\sigma_t^2 \sigma_{t-i}^2], \quad \text{for } i = 1, \ldots, 10$

where for any positive integer $i$ and positive constants $r$ and $s$,

$E[\sigma_t^r] = \exp \left( \frac{r}{2} \mu + \frac{r^2}{8} \sigma_h^2 \right)$

and

$E[\sigma_t^r \sigma_{t-i}^s] = E[\sigma_t^r] E[\sigma_t^s] \exp \left( \frac{r s \phi^i \sigma_h^2}{4} \right)$.

Then the moment conditions are

$$g_T(\theta) = \begin{bmatrix}
\frac{1}{T-10} \sum_{t=11}^{T} |y_t| - (2/\pi)^{1/2} \exp \left( \frac{\mu}{2} + \frac{\sigma_h^2}{8} \right) \\
\frac{1}{T-10} \sum_{t=11}^{T} y_t^2 - \exp \left( \mu + \frac{\sigma_h^2}{2} \right) \\
\vdots \\
\frac{1}{T-10} \sum_{t=11}^{T} y_t^2 y_{t-10}^2 - \exp \left( \mu + \frac{\sigma_h^2}{2} \right)^2 \exp \left( \phi^{10} \sigma_h^2 \right)
\end{bmatrix} (2.13)$$
The moment conditions are defined in an R function in Algorithm 2.2, which is then used in Algorithm 2.3 which uses the R package gmm (Chaussé, 2010) to estimate the parameters $\theta$.

**Algorithm 2.2** R function for the SV moment conditions.

```r
## define moment equations as in AS96 pp. 350--51
sv.moments = function (theta, y){
  mu = theta[1]
  phi = theta[2]
  sigmatau = theta[3]
  sig2h = sigmatau^2/(1-phi^-2)
  # the moments:
  moments <- c(
    m1 = sqrt(2/pi) * exp(mu/2 + sig2h /8) ,
    m2 = exp(mu + sig2h/2),
    m3 = 2 * sqrt(2/pi) * exp(3*mu/2 + 9*sig2h /8),
    m4 = 3 * exp(2*mu + 2*sig2h),
    m5to14 = 2/pi * exp(2*(mu/2 + sig2h /8) + phi^-(-1:10) * sig2h/4),
    m15to24 = exp(2*(mu + sig2h/2) + 4 * phi^-(-1:10) * sig2h/4)
  )
  # transform data:
  nolags <- cbind ( abs (y),y^2 , abs (y)^3 , y ^4)
  nolags <- nolags[ -(1:10) ,]
  lags <- tslag (y, 1:10 , trim = TRUE ) * as.vector ( y[ -(1:10) ])
  gmm.data <- data.frame ( nolags , abs ( lags ), lags ^2)
  # moment conditions g^T( theta):
  return (t(t(gmm.data) - moments))
}
```

**Algorithm 2.3** Fit the SV model using the R package gmm.

```r
library ( gmm )
gmm.fit <- gmm (g = sv.moments , x = y, t0 = c(mu=-10, phi= 0.9,
    sigmatau= 0.2), type = "iterative", vcov = 'HAC', optfct="nlminb", lower=c(-Inf, -0.9999, 0.00001), upper = c(Inf, 0.9999 , Inf))
```

The FKF package (see Section 2.2) is then used to run a Kalman filter with the GMM parameter estimates to estimate the volatilities $\{\sigma_t\}$, see Figure 2.10.

The parameter estimates and a comparison to MCMC and GMM are found in Section 2.4.
2.4 Results

Figure 2.11 plots the simulated volatilities $\sigma_t$ together with the estimated volatilities from MCMC, QML and GMM. Table 2.2 summarizes the parameter estimations and the root mean square error (RMSE) for the estimated volatilities

$$\text{RMSE}(\sigma_t) = \left( \frac{1}{T} \sum_{t=1}^{T} (\sigma_t^{\text{true}} - \hat{\sigma}_t)^2 \right)^{1/2}$$  \hspace{1cm} (2.14)

where $\sigma_t^{\text{true}}$ denotes the simulated volatility for period $t$ and $\hat{\sigma}_t$ its estimate.

As can be seen in Table 2.2 the estimates of $\mu$ are all fairly similar and close to the true value of $-9$. QML actually comes closest but for $\phi$ and $\sigma_\eta$ QML performs the worst, while MCMC and GMM performs similarly for this sample. However, when we compare the estimates of the volatilities $\sigma_t$ plotted in Figure 2.11 we see that MCMC outperforms them both by a large margin, the Kalman filters gives too smooth estimates. This is confirmed by the RMSE, which is much lower for MCMC.

The comparison in this section was informal, with just one simulated sample, to introduce the estimation methods and some of their differences. In the next section we will conduct a large scale simulation study where many parameter specifications and sample sizes will be compared.
Figure 2.11 The simulated volatilities compared to the estimates from MCMC, QML and GMM.

Table 2.2 Summary of the estimates of the simulated SV series.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>$\theta^{true}$</th>
<th>$\hat{\theta}$</th>
<th>SE</th>
<th>5%</th>
<th>95%</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMC</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu$</td>
<td>$-9$</td>
<td>$-8.977$</td>
<td>$0.106$</td>
<td>$-9.145$</td>
<td>$-8.809$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.97</td>
<td>0.927</td>
<td>0.034</td>
<td>0.864</td>
<td>0.971</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>0.15</td>
<td>0.184</td>
<td>0.051</td>
<td>0.110</td>
<td>0.276</td>
</tr>
<tr>
<td>RMSE($\sigma_t$)</td>
<td>17.72</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>QML</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu$</td>
<td>$-9$</td>
<td>$-8.995$</td>
<td>$0.090$</td>
<td>$-9.143$</td>
<td>$-8.847$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.97</td>
<td>0.839</td>
<td>0.091</td>
<td>0.690</td>
<td>0.988</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>0.15</td>
<td>0.288</td>
<td>0.121</td>
<td>0.089</td>
<td>0.487</td>
</tr>
<tr>
<td>RMSE($\sigma_t$)</td>
<td>24.28</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>GMM</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>$\mu$</td>
<td>$-9$</td>
<td>$-8.952$</td>
<td>$0.037$</td>
<td>$-9.056$</td>
<td>$-8.935$</td>
</tr>
<tr>
<td>$\phi$</td>
<td>0.97</td>
<td>0.933</td>
<td>0.095</td>
<td>0.683</td>
<td>0.996</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>0.15</td>
<td>0.127</td>
<td>0.102</td>
<td>0.120</td>
<td>0.455</td>
</tr>
<tr>
<td>RMSE($\sigma_t$)</td>
<td>24.29</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note: RMSE $\times 10^4$ is displayed for legibility.
3 A Simulation Study Comparing MCMC to QML and GMM

In this section we conduct a simulation study, comparing the MCMC, QML and GMM estimators for the stochastic volatility model. We simulate samples from nine different parameter specifications (see Table 3.1) and from three different sample sizes ($T = 500$, $T = 2500$ and $T = 5000$). For each specification $N = 100$ samples are simulated, each of which are estimated by MCMC, QML and GMM.

Table 3.1 The nine different parameter specifications used in the simulation study. Note that $\mu$ is always set to $\mu = -9$, so that $E[\sigma^2_t] = e^{\mu + \sigma_h^2/2(1-\phi^2)} \approx 0.00012$, which yields an expected yearly standard deviation of approximately 18% (assuming daily data).

<table>
<thead>
<tr>
<th>$CV^2$</th>
<th>$\phi$</th>
<th>0.90</th>
<th>0.97</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.10</td>
<td>$\sigma_\eta$</td>
<td>0.135</td>
<td>0.075</td>
<td>0.044</td>
</tr>
<tr>
<td>0.46</td>
<td>$\sigma_\eta$</td>
<td>0.269</td>
<td>0.150</td>
<td>0.087</td>
</tr>
<tr>
<td>1.00</td>
<td>$\sigma_\eta$</td>
<td>0.363</td>
<td>0.202</td>
<td>0.117</td>
</tr>
</tbody>
</table>

Since $h_t \equiv \log \sigma_t^2 \sim \mathcal{N}(\mu, \sigma_h^2)$, where $\sigma_h^2 = \sigma_\eta^2/(1-\phi^2)$, then (from the properties of the log-normal distribution, see footnote 5)

$$\sigma_t^2 \sim \mathcal{N}(e^{\mu + \sigma_h^2/2}, e^{2\mu + \sigma_h^2 (e^{\sigma_h^2} - 1)})$$

The (squared) coefficient of variation for $\sigma_t^2$ is then

$$CV^2 = \frac{\text{Var}(\sigma_t^2)}{(E[\sigma_t^2])^2} = \frac{e^{2\mu + \sigma_h^2 (e^{\sigma_h^2} - 1)}}{e^{2\mu + \sigma_h^2}} = e^{\sigma_h^2} - 1 \quad (3.1)$$

The (squared) coefficient of variation has been used as a measure of which parameter specifications are relevant (see e.g., Jacquier et al., 1994 and Andersen & Sørensen, 1997). From their own estimations and a review of previous studies Jacquier et al. (1994) finds that $CV^2$ varies between 0.43 and 1.61, with values between 0.5 and 1 the most common, and with values of $\phi$ ranging from around 0.90 to 0.995 for daily data (and between 0.7 to 0.97 for weekly data). Six of the nine parameter specifications used in our study are in this range of common empirical values ($CV^2 = 0.46$ and $CV^2 = 1.00$). We have also included three specifications for the very low coefficient of variation $CV^2 = 0.10$ to see how the estimators handle those specifications. We use three different specifications of $\phi$ for each $CV^2$, namely $\phi = 0.90$, $\phi = 0.97$, and $\phi = 0.99$ from which we calculated the $\sigma_\eta$ values to be used. We have $\mu$ fixed at $\mu = -9$, so that the yearly standard deviation is roughly 18%, assuming daily log-returns $y_t$. 


3.1 Parameter estimates

As a measure of efficiency of the estimators we use the root mean square error (RMSE) defined as

$$\text{RMSE}(\hat{\theta}) = \left( \frac{1}{N} \sum_{i=1}^{N} (\theta_{\text{true}} - \hat{\theta}^{(i)})^2 \right)^{1/2} \tag{3.2}$$

where \(\theta_{\text{true}} = (\mu_{\text{true}}, \phi_{\text{true}}, \sigma_{\eta_{\text{true}}})^\top\) denotes the true values of the parameters and \(\hat{\theta}^{(i)}\) denotes the estimated \(\theta\) for the \(i\)th sample.

In Table 3.2 (for \(T = 500\)), Table 3.4 (for \(T = 2500\)) and Table 3.6 (\(T = 5000\)) we display the mean values of the parameter estimates for \(N = 100\) samples for each specification as well as the corresponding RMSE (in parenthesis). The corresponding Table 3.3, Table 3.5 and Table 3.7, for \(T = 500\), \(T = 2500\) and \(T = 5000\), respectively, displays the efficiency of MCMC in terms of QML and GMM, for each parameter, which we calculate as the RMSE ratio, i.e., the ratio of the RMSE of QML and GMM over MCMC. Table 3.8 displays how RMSE changes as the sample size grows.

Table 3.2 The means and RMSE of the MCMC, QML and GMM parameter estimates for \(N = 100\) samples (for each specification) of length \(T = 500\).

<table>
<thead>
<tr>
<th>(CV^2 = 0.1)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>(-9)</td>
<td>(0.90)</td>
<td>(0.135)</td>
<td>(-9.00)</td>
<td>(0.97)</td>
<td>(0.075)</td>
<td>(-9)</td>
<td>(0.99)</td>
<td>(0.044)</td>
</tr>
<tr>
<td>MCMC</td>
<td>(-9.002)</td>
<td>(0.892)</td>
<td>(0.129)</td>
<td>(-8.999)</td>
<td>(0.919)</td>
<td>(0.112)</td>
<td>(-8.981)</td>
<td>(0.918)</td>
<td>(0.100)</td>
</tr>
<tr>
<td>QML</td>
<td>(-8.998)</td>
<td>(0.889)</td>
<td>(0.117)</td>
<td>(-8.990)</td>
<td>(0.891)</td>
<td>(0.126)</td>
<td>(-8.981)</td>
<td>(0.889)</td>
<td>(0.118)</td>
</tr>
<tr>
<td>GMM</td>
<td>(-9.040)</td>
<td>(0.689)</td>
<td>(0.051)</td>
<td>(-9.042)</td>
<td>(0.722)</td>
<td>(0.033)</td>
<td>(-9.022)</td>
<td>(0.707)</td>
<td>(0.039)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(CV^2 = 0.46)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>(-9)</td>
<td>(0.90)</td>
<td>(0.269)</td>
<td>(-9)</td>
<td>(0.97)</td>
<td>(0.150)</td>
<td>(-9)</td>
<td>(0.99)</td>
<td>(0.087)</td>
</tr>
<tr>
<td>MCMC</td>
<td>(-9.007)</td>
<td>(0.804)</td>
<td>(0.353)</td>
<td>(-8.981)</td>
<td>(0.884)</td>
<td>(0.220)</td>
<td>(-8.998)</td>
<td>(0.954)</td>
<td>(0.131)</td>
</tr>
<tr>
<td>QML</td>
<td>(-8.999)</td>
<td>(0.778)</td>
<td>(0.414)</td>
<td>(-8.971)</td>
<td>(0.742)</td>
<td>(0.315)</td>
<td>(-8.990)</td>
<td>(0.911)</td>
<td>(0.172)</td>
</tr>
<tr>
<td>GMM</td>
<td>(-9.054)</td>
<td>(0.876)</td>
<td>(0.137)</td>
<td>(-9.043)</td>
<td>(0.681)</td>
<td>(0.062)</td>
<td>(-9.061)</td>
<td>(0.864)</td>
<td>(0.046)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(CV^2 = 1)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
<th>(\mu)</th>
<th>(\phi)</th>
<th>(\sigma_\eta)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>(-9)</td>
<td>(0.90)</td>
<td>(0.363)</td>
<td>(-9)</td>
<td>(0.97)</td>
<td>(0.202)</td>
<td>(-9)</td>
<td>(0.99)</td>
<td>(0.117)</td>
</tr>
<tr>
<td>MCMC</td>
<td>(-9.021)</td>
<td>(0.900)</td>
<td>(0.364)</td>
<td>(-8.971)</td>
<td>(0.937)</td>
<td>(0.251)</td>
<td>(-8.962)</td>
<td>(0.968)</td>
<td>(0.156)</td>
</tr>
<tr>
<td>QML</td>
<td>(-9.027)</td>
<td>(0.905)</td>
<td>(0.335)</td>
<td>(-8.967)</td>
<td>(0.931)</td>
<td>(0.255)</td>
<td>(-8.955)</td>
<td>(0.935)</td>
<td>(0.208)</td>
</tr>
<tr>
<td>GMM</td>
<td>(-9.122)</td>
<td>(0.891)</td>
<td>(0.205)</td>
<td>(-9.094)</td>
<td>(0.937)</td>
<td>(0.104)</td>
<td>(-9.041)</td>
<td>(0.921)</td>
<td>(0.070)</td>
</tr>
</tbody>
</table>

Note: RMSE in parenthesis.
Table 3.3 The mean and range of the ratio of QML to MCMC, and GMM to MCMC, of RMSE for the estimates in Table 3.2 ($T = 500$).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RMSE Ratio</th>
<th>Mean</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>QML</td>
<td>1.115</td>
<td>[1.02, 1.30]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>1.120</td>
<td>[1.06, 1.18]</td>
</tr>
<tr>
<td>$\phi$</td>
<td>QML</td>
<td>1.477</td>
<td>[0.49, 2.71]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>4.937</td>
<td>[1.13, 12.16]</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>QML</td>
<td>2.158</td>
<td>[1.37, 3.59]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>1.842</td>
<td>[0.96, 2.90]</td>
</tr>
</tbody>
</table>

From Table 3.2 and Table 3.3 we see that MCMC is more efficient than both QML and GMM on average when $T = 500$ for all parameters. The RMSE of $\mu$ are quite similar for all three methods but slightly better for MCMC across all specifications. For $\phi$ there is larger variation, MCMC strongly dominates GMM, and is more efficient than QML on average and in all but one specification, namely the slightly unrealistic case with $CV^2 = 0.1$ and $\phi = 0.9$ (so that $\sigma_\eta = 0.135$), where QML is twice as efficient as MCMC (however both $\mu$ and $\sigma_\eta$ are more efficiently estimated by MCMC for the same specification). For $\sigma_\eta$, MCMC is better than average than both QML and GMM (roughly twice as efficient as both) and better than QML for all specification and better than GMM for all specifications but one, namely for $CV^2 = 0.1$ and $\phi = 0.99$, GMM is 4% more efficient than MCMC (however, for the same specification when $T = 2500$ and $T = 5000$, MCMC is more than twice as efficient).
Table 3.4 The means and RMSE of the MCMC, QML and GMM parameter estimates for \( N = 100 \) samples (for each specification) of length \( T = 2500 \).

<table>
<thead>
<tr>
<th>Method</th>
<th>( \mu )</th>
<th>( \phi )</th>
<th>( \sigma_\eta )</th>
<th>( \mu )</th>
<th>( \phi )</th>
<th>( \sigma_\eta )</th>
<th>( \mu )</th>
<th>( \phi )</th>
<th>( \sigma_\eta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMC</td>
<td>-9.002</td>
<td>0.887</td>
<td>0.135</td>
<td>-9.005</td>
<td>0.978</td>
<td>0.044</td>
<td>-9.005</td>
<td>0.978</td>
<td>0.060</td>
</tr>
<tr>
<td></td>
<td>(0.044)</td>
<td>(0.038)</td>
<td>(0.036)</td>
<td>(0.077)</td>
<td>(0.019)</td>
<td>(0.023)</td>
<td>(0.077)</td>
<td>(0.019)</td>
<td>(0.023)</td>
</tr>
<tr>
<td>QML</td>
<td>-9.002</td>
<td>0.889</td>
<td>0.132</td>
<td>-9.004</td>
<td>0.889</td>
<td>0.157</td>
<td>-9.004</td>
<td>0.889</td>
<td>0.157</td>
</tr>
<tr>
<td></td>
<td>(0.051)</td>
<td>(0.011)</td>
<td>(0.057)</td>
<td>(0.086)</td>
<td>(0.101)</td>
<td>(0.122)</td>
<td>(0.086)</td>
<td>(0.101)</td>
<td>(0.122)</td>
</tr>
<tr>
<td>GMM</td>
<td>-9.008</td>
<td>0.912</td>
<td>0.081</td>
<td>-9.009</td>
<td>0.959</td>
<td>0.036</td>
<td>-9.009</td>
<td>0.959</td>
<td>0.036</td>
</tr>
<tr>
<td></td>
<td>(0.047)</td>
<td>(0.115)</td>
<td>(0.087)</td>
<td>(0.083)</td>
<td>(0.115)</td>
<td>(0.050)</td>
<td>(0.083)</td>
<td>(0.115)</td>
<td>(0.050)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Method</th>
<th>( \mu )</th>
<th>( \phi )</th>
<th>( \sigma_\eta )</th>
<th>( \mu )</th>
<th>( \phi )</th>
<th>( \sigma_\eta )</th>
<th>( \mu )</th>
<th>( \phi )</th>
<th>( \sigma_\eta )</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMC</td>
<td>-9.006</td>
<td>0.889</td>
<td>0.282</td>
<td>-9.008</td>
<td>0.964</td>
<td>0.159</td>
<td>-9.023</td>
<td>0.987</td>
<td>0.093</td>
</tr>
<tr>
<td></td>
<td>(0.056)</td>
<td>(0.026)</td>
<td>(0.037)</td>
<td>(0.098)</td>
<td>(0.013)</td>
<td>(0.024)</td>
<td>(0.165)</td>
<td>(0.006)</td>
<td>(0.017)</td>
</tr>
<tr>
<td>QML</td>
<td>-9.990</td>
<td>0.852</td>
<td>0.313</td>
<td>-9.008</td>
<td>0.926</td>
<td>0.232</td>
<td>-9.021</td>
<td>0.974</td>
<td>0.112</td>
</tr>
<tr>
<td></td>
<td>(0.065)</td>
<td>(0.132)</td>
<td>(0.134)</td>
<td>(0.102)</td>
<td>(0.060)</td>
<td>(0.107)</td>
<td>(0.165)</td>
<td>(0.039)</td>
<td>(0.063)</td>
</tr>
<tr>
<td>GMM</td>
<td>-9.009</td>
<td>0.917</td>
<td>0.196</td>
<td>-9.025</td>
<td>0.974</td>
<td>0.078</td>
<td>-9.034</td>
<td>0.993</td>
<td>0.042</td>
</tr>
<tr>
<td></td>
<td>(0.063)</td>
<td>(0.060)</td>
<td>(0.107)</td>
<td>(0.104)</td>
<td>(0.071)</td>
<td>(0.108)</td>
<td>(0.180)</td>
<td>(0.013)</td>
<td>(0.062)</td>
</tr>
</tbody>
</table>

Note: RMSE in parenthesis.

Table 3.5 The mean and range of the ratio of QML to MCMC, and GMM to MCMC, of RMSE for the estimates in Table 3.4 (\( T = 2500 \)).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RMSE Ratio (Method/MCMC)</th>
<th>Mean</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \mu )</td>
<td>QML</td>
<td>1.093</td>
<td>[1.00, 1.21]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>1.075</td>
<td>[1.02, 1.12]</td>
</tr>
<tr>
<td>( \phi )</td>
<td>QML</td>
<td>3.171</td>
<td>[0.29, 6.22]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>3.352</td>
<td>[2.02, 6.13]</td>
</tr>
<tr>
<td>( \sigma_\eta )</td>
<td>QML</td>
<td>2.965</td>
<td>[1.39, 5.33]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>3.333</td>
<td>[2.03, 4.96]</td>
</tr>
</tbody>
</table>

26
Table 3.6 The means and RMSE of the MCMC, QML and GMM parameter estimates for $N = 100$ samples (for each specification) of length $T = 5000$.

<table>
<thead>
<tr>
<th>Method</th>
<th>$\mu$</th>
<th>$\phi$</th>
<th>$\sigma_\eta$</th>
<th>$\mu$</th>
<th>$\phi$</th>
<th>$\sigma_\eta$</th>
<th>$\mu$</th>
<th>$\phi$</th>
<th>$\sigma_\eta$</th>
</tr>
</thead>
<tbody>
<tr>
<td>MCMC</td>
<td>-8.999</td>
<td>0.889</td>
<td>0.135</td>
<td>-9.007</td>
<td>0.959</td>
<td>0.087</td>
<td>-9.007</td>
<td>0.959</td>
<td>0.087</td>
</tr>
<tr>
<td>QUML</td>
<td>-9.004</td>
<td>0.889</td>
<td>0.144</td>
<td>-9.005</td>
<td>0.889</td>
<td>0.160</td>
<td>-9.005</td>
<td>0.889</td>
<td>0.160</td>
</tr>
<tr>
<td>GMM</td>
<td>-9.002</td>
<td>0.906</td>
<td>0.102</td>
<td>-9.008</td>
<td>0.955</td>
<td>0.062</td>
<td>-9.008</td>
<td>0.955</td>
<td>0.062</td>
</tr>
</tbody>
</table>

Note: RMSE in parenthesis.

Table 3.7 The mean and range of the ratio of QML to MCMC, and GMM to MCMC, of RMSE for the estimates in Table 3.6 ($T = 5000$).

<table>
<thead>
<tr>
<th>Parameter</th>
<th>RMSE Ratio (Method/MCMC)</th>
<th>Mean</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\mu$</td>
<td>QML</td>
<td>1.090</td>
<td>[1.05, 1.37]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>1.046</td>
<td>[0.99, 1.11]</td>
</tr>
<tr>
<td>$\phi$</td>
<td>QML</td>
<td>4.416</td>
<td>[0.38, 15.62]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>3.788</td>
<td>[1.99, 8.44]</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>QML</td>
<td>3.879</td>
<td>[1.27, 12.05]</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>4.067</td>
<td>[2.63, 6.11]</td>
</tr>
</tbody>
</table>

A similar pattern is seen for the longer simulated series, $T = 2500$ (in Table 3.4 & Table 3.5) and $T = 5000$ (in Table 3.6 & Table 3.7). QML is still more efficient than MCMC for estimating $\phi$ (but not $\mu$ and $\sigma_\eta$) for the specification $\phi = 0.90$ and $CV^2 = 0.1$, but MCMC is more efficient than both QML and GMM on average and for all the eight other specifications. On average, MCMC is about three times as efficient as QML and GMM for $\phi$ and $\sigma_\eta$ when $T = 2500$ and about four
times as efficient when $T = 5000$, while the efficiency of $\mu$ does not differ that much—MCMC is 9% more efficient than QML for both $T = 2500$ and $T = 5000$, and 8% respectively 5% better than GMM for $T = 2500$ and $T = 5000$.

### Table 3.8 The change in RMSE($\hat{\theta}$) as $T$ grows.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>MCMC</th>
<th>QML</th>
<th>GMM</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T=500$</td>
<td>$T=2500$</td>
<td>$T=5000$</td>
<td>$T=500$</td>
</tr>
<tr>
<td>$\mu$</td>
<td>2.115</td>
<td>2.937</td>
<td>2.154</td>
</tr>
<tr>
<td>$\phi$</td>
<td>4.905</td>
<td>8.806</td>
<td>3.173</td>
</tr>
<tr>
<td>$\sigma_\eta$</td>
<td>2.741</td>
<td>4.288</td>
<td>2.329</td>
</tr>
</tbody>
</table>

From Table 3.8 we see, as expected, that the efficiency of estimating all the three parameters increases for all the three methods as the sample size $T$ grows.

### 3.2 Volatility estimates

To evaluate the estimated volatilities we use the total root mean square error (RMSE) over all the $N = 100$ samples.

$$\text{RMSE}(\hat{\sigma}_t) = \left(\frac{1}{N \cdot T} \sum_{i=1}^{N} \sum_{t=1}^{T} (\sigma_{t}^{\text{true}}(i) - \hat{\sigma}_t^{(i)}))^2\right)^{1/2} (3.3)$$

where $\sigma_{t}^{\text{true}}(i)$ denotes the simulated volatility for period $t$ in sample $i$, and $\hat{\sigma}_t^{(i)}$ denotes the estimate of that volatility, which for MCMC is the mean from the posterior distribution, while for QML and GMM it comes from Kalman filters run with the parameter estimates. We also run Kalman filters using the true values of the parameters as a reference. Table 3.9, Table 3.10, and Table 3.11 displays the RMSE ($\times 10000$) of the estimated volatilities, for $T = 500$, $T = 2500$ and $T = 5000$, respectively. Table 3.12 summarizes the previous tables with the RMSE ratio in terms of MCMC, i.e., the efficiency of the MCMC estimates compared to the QML and GMM (and Kalman filters of the true parameters).
Table 3.9 RMSE(\(\hat{\sigma}_t\)) of Kalman filters for the QML and GMM estimates as well as for the true parameters compared to MCMC for \(N = 100\) samples of length \(T = 500\).

<table>
<thead>
<tr>
<th>(CV^2)</th>
<th>Method</th>
<th>0.90</th>
<th>0.97</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>MCMC</td>
<td>16.34</td>
<td>12.35</td>
<td>10.24</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>18.04</td>
<td>18.04</td>
<td>12.50</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>18.42</td>
<td>22.03</td>
<td>12.26</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>17.09</td>
<td>19.85</td>
<td>10.34</td>
</tr>
<tr>
<td>0.46</td>
<td>MCMC</td>
<td>33.66</td>
<td>26.34</td>
<td>20.17</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>44.63</td>
<td>33.09</td>
<td>23.79</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>46.32</td>
<td>40.86</td>
<td>22.33</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>44.15</td>
<td>36.95</td>
<td>22.36</td>
</tr>
<tr>
<td>1</td>
<td>MCMC</td>
<td>36.56</td>
<td>38.85</td>
<td>16.36</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>47.76</td>
<td>49.26</td>
<td>23.85</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>59.82</td>
<td>61.11</td>
<td>32.45</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>49.66</td>
<td>53.44</td>
<td>23.57</td>
</tr>
</tbody>
</table>

*Note: RMSE \(\times 10^4\) is displayed for legibility.*

Table 3.10 RMSE(\(\hat{\sigma}_t\)) of Kalman filters for the QML and GMM estimates as well as for the true parameters compared to MCMC for \(N = 100\) samples of length \(T = 2500\).

<table>
<thead>
<tr>
<th>(CV^2)</th>
<th>Method</th>
<th>0.90</th>
<th>0.97</th>
<th>0.99</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>MCMC</td>
<td>16.85</td>
<td>36.56</td>
<td>8.89</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>19.15</td>
<td>47.76</td>
<td>16.24</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>19.37</td>
<td>59.82</td>
<td>16.63</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>18.72</td>
<td>49.66</td>
<td>14.28</td>
</tr>
<tr>
<td>0.46</td>
<td>MCMC</td>
<td>24.70</td>
<td>20.23</td>
<td>15.15</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>32.95</td>
<td>30.05</td>
<td>24.11</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>33.06</td>
<td>31.58</td>
<td>24.49</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>32.42</td>
<td>30.13</td>
<td>19.22</td>
</tr>
<tr>
<td>1</td>
<td>MCMC</td>
<td>34.88</td>
<td>31.98</td>
<td>18.46</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>44.71</td>
<td>46.17</td>
<td>31.50</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>47.68</td>
<td>52.12</td>
<td>41.14</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>45.14</td>
<td>49.99</td>
<td>31.09</td>
</tr>
</tbody>
</table>

*Note: RMSE \(\times 10^4\) is displayed for legibility.*
Table 3.11 RMSE($\hat{\sigma}_t$) of Kalman filters for the QML and GMM estimates as well as for the true parameters compared to MCMC for $N = 100$ samples of length $T = 5000$.

<table>
<thead>
<tr>
<th>$CV^2$</th>
<th>Method</th>
<th>$\phi$</th>
<th>$\phi$</th>
<th>$\phi$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>0.90</td>
<td>0.97</td>
<td>0.99</td>
</tr>
<tr>
<td>0.1</td>
<td>MCMC</td>
<td>15.08</td>
<td>12.49</td>
<td>9.98</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>16.27</td>
<td>17.12</td>
<td>15.27</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>16.17</td>
<td>16.94</td>
<td>13.72</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>15.94</td>
<td>16.38</td>
<td>12.90</td>
</tr>
<tr>
<td>0.46</td>
<td>MCMC</td>
<td>26.87</td>
<td>20.97</td>
<td>15.30</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>33.14</td>
<td>29.76</td>
<td>19.65</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>33.21</td>
<td>31.55</td>
<td>21.27</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>33.03</td>
<td>30.10</td>
<td>19.69</td>
</tr>
<tr>
<td>1</td>
<td>MCMC</td>
<td>33.71</td>
<td>26.28</td>
<td>17.73</td>
</tr>
<tr>
<td></td>
<td>QML</td>
<td>45.09</td>
<td>39.84</td>
<td>28.04</td>
</tr>
<tr>
<td></td>
<td>GMM</td>
<td>46.19</td>
<td>41.18</td>
<td>36.59</td>
</tr>
<tr>
<td></td>
<td>True parameters</td>
<td>46.13</td>
<td>40.04</td>
<td>28.33</td>
</tr>
</tbody>
</table>

Note: RMSE $\times 10^4$ is displayed for legibility.

Table 3.12 A summary of the relative performance from Tables 3.9, 3.10 and 3.11, comparing the filters of QML, GMM and the true parameters to MCMC.

<table>
<thead>
<tr>
<th>RMSE ratio (Method/MCMC)</th>
<th>$T = 500$ Mean Range</th>
<th>$T = 2500$ Mean Range</th>
<th>$T = 5000$ Mean Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM</td>
<td>1.482 [1.107,1.984]</td>
<td>1.600 [1.150,2.229]</td>
<td>1.437 [1.072,2.064]</td>
</tr>
</tbody>
</table>

As we can see from Table 3.9, Table 3.10, Table 3.11 and Table 3.12, for estimating the volatilities $\sigma_t$, MCMC has the lowest RMSE for all specifications and sample sizes—even beating out the Kalman filter of the true parameters. MCMC is roughly 29% more efficient on average than QML, 48% more efficient than GMM and 30% more efficient than the true parameters for $T = 500$, 45% more efficient than QML, 60% more efficient than GMM and 40% more efficient than the true parameters for $T = 2500$, 37% more efficient than QML, 44% more efficient than GMM and 35% more efficient than the true parameters for $T = 5000$.

3.3 Conclusions and ideas for further research

As a conclusion, MCMC performs the best (i.e., has the lowest RMSE) for all parameter specifications and sample sizes regarding estimation of the volatilities.
and for almost all specifications regarding the parameter estimates and by a large margin on average. The one exception is the slightly unrealistically low $CV^2 = 0.1$ (not found in the empirical series investigated by Jacquier et al., 1994) with $\phi = 0.90$ for which QML is better than MCMC for estimating $\phi$ (but not $\mu$ or $\sigma_\eta$). The reason for this is unknown to the author and would require further research, but the effect seems real and does not go away with increased sample sizes.

There exist many more estimation methods than those presented in this thesis, both Bayesian and frequentist. For example, further studies could compare MCMC to simulated method of moments (SMM) and efficient method of moments (EMM)—which can be viewed as an improved GMM, importance sampling (IS) approaches—which can be viewed as an improvement of QML—and extensions, such as improved importance sampling (IIS), efficient importance sampling (EIS) and the Bayesian single site sampler (SSS) and multi-move sampler (MMS).

This simulation study only covers the most basic univariate SV model. There are numerous extensions, e.g., removal of the normality assumption (usually replacing it with a Student’s t-distribution), correlated error terms for a leverage effect, and extensions to the multivariate case. It would be very interesting to conduct simulation studies comparing the efficiency in estimating the different extensions with different methods as well.

---

12 Descriptions and references to all of the estimation methods mentioned in this section can be found in Bos (2012).
References


## A Bayesian Methods

This section will be a short introduction to Bayesian inference; it is based on Marin and Robert (2014) and Robert (2001).

*Bayes’ Theorem* tells us that the conditional probabilities of two events $A$ and $B$, $\Pr(A|B)$ and $\Pr(B|A)$, are related by (given that $\Pr(B) \neq 0$) (Robert, 2001, pp. 8–9; Casella & Berger, 2001, pp. 20–23)

$$
\Pr(A|B) = \frac{\Pr(B|A) \Pr(A)}{\Pr(B|A) \Pr(A) + \Pr(B|A^c) \Pr(A^c)} = \frac{\Pr(B|A) \Pr(A)}{\Pr(B)}.
$$

In a similar manner conditional distributions can be inverted. Given two random variables $x$ and $y$ with conditional distribution $f(x|y)$ and marginal distribution $g(y)$, the conditional distribution of $y$ given $x$ is

$$
g(y|x) = \frac{f(x|y)g(y)}{\int f(x|y)g(y) \, dy}.
\tag{A.1}
$$

Given the data $y = (y_1, \ldots, y_T)^\top$ from a density $f(y|\theta)$ with parameters $\theta = (\theta_1, \ldots, \theta_p)^\top \in \Theta$, the associated likelihood function is

$$
\ell(\theta|y) = \prod_{t=1}^T f(y_t|\theta).
$$

In contrast to the frequentist likelihood approach, the Bayesian approach modifies the likelihood into a *posterior distribution* with a *prior distribution* (the apriori knowledge), which is a probability distribution on the parameter vector $\theta$ with density $\pi(\theta)$—the posterior distribution can be computed using Bayes’ formula in (A.1)

$$
\pi(\theta|y) = \frac{\ell(\theta|y)\pi(\theta)}{\int \ell(\theta|y)\pi(\theta) \, d\theta} \propto \ell(\theta|y)\pi(\theta)
\tag{A.2}
$$

Or in words: the posterior distribution $\pi(\theta|y)$ is proportional to the likelihood $\ell(\theta|y)$ multiplied by the prior distribution $\pi(\theta)$.

### A.1 Priors

There are different kinds of prior distributions. *Conjugate* priors are distributions to which the posterior distribution belongs to the same distributional family.

---

13 The symbol "$\propto$" is read as “proportional to” and is used for two functions as $h \propto g$, meaning that $h(x) = c \cdot g(x) \forall x$, for some $c \neq 0$. If $h(x)$ is a PDF and $h \propto g$, then $g$ uniquely determines $h$ and $h(x) = g(x)/\int g(x) \, dx$. 

---
Examples of conjugate priors\footnote{See, for example, http://en.wikipedia.org/wiki/Conjugate\_prior#Table\_of\_conjugate\_distributions for a long list.} are: (1) when the likelihood is $\mathcal{N}(\mu, \sigma^2)$ and $\sigma^2$ is known, then $\mu \sim \mathcal{N}(\mu_0, \sigma_0^2)$, (2) if the likelihood is $\mathcal{N}(\mu, \sigma^{-2})$ and $\mu$ is known, then $\sigma^{-2} \sim \mathcal{G}(\alpha, \beta)$, from which it follows that, (3) if the likelihood is $\mathcal{N}(\mu, \sigma^2)$ and $\mu$ is known, then $\sigma^2 \sim \mathcal{IG}(\alpha, \beta)$.

Noninformative or diffuse priors are used when no previous information is available, e.g., a uniform distribution can be used to give equal weight to all values. Sometimes improper priors are used, meaning that it does not integrate to one (so that it is not a real density).

## B Simulation Methods

*Markov Chain Monte Carlo* (MCMC) is a computer intensive method of sampling from (usually high dimensional) distributions. As the name suggests it is based on *Markov chains*, and *Monte Carlo simulations* (generating pseudo-random numbers using a computer)—a method pioneered in the very early days of computers by Stanislaw Ulam and John von Neumann, working on the Manhattan project.

Robert and Casella (2004, Def. 7.1, p. 268) defines an MCMC method of simulating from a distribution $f$ as “any method that produces an ergodic\footnote{The *ergodic theorem* is the usual name for the Markov chain version of the Law of Large Numbers (LLN). See Robert and Casella (2004, Chapter 6) for related theorems and proofs.} Markov chain $\{X(t)\}$ whose stationary distribution is $f$.”


Before introducing MCMC we will look at some other related simulation methods.

### B.1 Accept-Reject Method

One way to simulate from a distribution indirectly is by using the accept-reject method (also know as the acceptance-rejection method or rejection sampling).

For accept-reject sampling to work the target density $f$ (the density we are interested in) has to be known up to a multiplicative constant. A known density $q$, which we can simulate from, called the instrumental or candidate density, is used to actually generate the sample. There are two requirements for rejection sampling to work: (i) There exists a finite known constant $M$, such that $M \geq f(x)/q(x)$
for all $x$ (i.e., $M = \sup_{x}[f(x)/q(x)]$). (ii) The support of $f$ and $q$ are compatible (Robert & Casella, 2010, p. 51).

These steps will simulate a random variable $X \sim f$ using rejection sampling:

1. Generate $Y \sim q(\cdot)$, and $U \sim U_{[0,1]}$
2. Accept $X = y$ if $u \leq \alpha$, where $\alpha$ is the acceptance probability
   \[
   \alpha = \frac{f(y)}{Mq(y)}
   \]
3. Else return to 1.

When Step 4 is accepted it counts as one drawing (the rejected candidates are just thrown away and will not be used). These steps are repeated to a desirable number of drawings. Algorithm B.1 shows an implementation in R for one iteration of rejection sampling, taken from Robert and Casella (2010, p. 51).

**Algorithm B.1** Generic R code for rejection sampling (from Robert & Casella, 2010, p. 51).

```r
u=runif(1)*M
y=randq(1) # Replace randq() with the appropriate distribution
while (u>f(y)/q(y)){
u=runif(1)*M
y=randq(1) }
```

Robert and Casella (2010, p. 52) show why the method works; the cdf of the generated random variable is

\[
\Pr \left( Y \leq x \left| U \leq \frac{f(Y)}{Mq(Y)} \right. \right) = \frac{\Pr(Y \leq x, U \leq f(Y))}{\Pr(U \leq f(Y))} = \frac{\int_{-\infty}^{x} \int_{0}^{f(y)/[Mq(y)]} du q(y) dy}{\int_{-\infty}^{+\infty} \int_{0}^{f(y)/[Mq(y)]} du q(y) dy}
\]

[The uniform integral is equal to its upper limit]

\[
= \frac{\int_{-\infty}^{x} f(y) q(y) dy}{\int_{-\infty}^{+\infty} f(y) q(y) dy} = \frac{\int_{-\infty}^{+\infty} f(y) dy}{\int_{-\infty}^{+\infty} f(y) dy} = \Pr(X \leq x)
\]
which is the cdf that we wanted.

A small example follows to make things a little more concrete (due to Robert & Casella, 2010, Example 2.7). We want to generate a beta random variable, say $X \sim B(2.7, 6.3)$ and we use the uniform $U_{[0,1]}$ as the candidate distribution. See Figure B.1 for a plot of the density enclosed in a box, representing $Y$, i.e., values drawn from $U_{[0,1]}$ on the horizontal axis, which is read of on the vertical axis as $f(y)$. Drawings outside of the density in Figure B.1 are rejected, and those inside accepted. The probability of acceptance is the probability of $U \leq f(Y)/(Mq(Y))$, or

$$
\Pr \left( U \leq \frac{f(Y)}{Mq(Y)} \right) = \int_{-\infty}^{+\infty} \int_{0}^{f(Y)/(Mq(Y))} duq(y) dy = \int_{-\infty}^{+\infty} \frac{f(Y)}{Mq(Y)} q(y) dy = \frac{1}{M} \int_{-\infty}^{+\infty} f(Y) dy = \frac{1}{M}
$$

We thus calculate the constant $M$ (remember, it is defined as the upper-bound $f(x)/q(x) \leq M$—which is the maximum of the beta density). This can be calculated in R with the `optimize` function

```r
M <- optimize(f=function(x){dbeta(x,2.7,6.3)}, maximum=TRUE, interval=c(0,1))$objective
```

which gives the maximum at 2.669744. The probability of acceptance is thus $1/2.67 = 0.37$ (which is quite close to what one might guess by eyeballing Figure B.1).
Next we want to adjust the general Algorithm B.1 for this example; see Algorithm B.2. Note, that $U^* \sim \mathcal{U}_{[0,M]}$ is equivalent to $U \sim \mathcal{U}_{[0,1]}$.


```r
Nsim = 1000
M <- optimize(f = function(x) { dbeta(x, 2.7, 6.3) }, maximum = TRUE, interval = c(0, 1))$objective
y <- runif(Nsim)  # generation from proposal dist
u <- runif(n = Nsim, min = 0, max = M)  # uniform over (0,M)

x <- y[u < dbeta(y, 2.7, 6.3)]  # accepted subsample
```

Instead of simulating from a uniform distribution, we could choose a proposal distribution more similar to the target distribution. We try by using the $\mathcal{B}(2,6)$ as a proposal in the example above. Optimization by

```r
M <- optimize(f = function(x) { dbeta(x, 2.7, 6.3) / dbeta(x, 2, 6) }, maximum = TRUE, interval = c(0, 1))$objective
```

yields $M = 1.671808$—which is lower than before—and the corresponding acceptance rate of 0.5981—compared to 0.3746 using the uniform proposal distribution. Figure B.2 shows 1000 simulations each using the different proposal distributions; on the left with $\mathcal{U}_{[0,1]}$ as proposal (370/1000 were accepted), and on the right with $\mathcal{B}(2,6)$ (617/1000 were accepted).

### B.2 Monte Carlo Integration

This section follows Robert and Casella (2004, Chapter 3) very closely. Suppose there is an integral (typically of high dimension, making it hard to solve analytically) of the form

$$\int_X h(x) \, dx.$$

If it is decomposable into the function $g(x)$ and the density $f(x)$ defined on $X$, it can be rewritten as an expectation

$$\int_X h(x) \, dx = \int_X g(x) f(x) \, dx = E_f[g(x)]. \quad \text{(B.1)}$$

If an i.i.d. sample $x_1, x_2, \ldots, x_n$ is drawn from a distribution with density $f(x)$, then the empirical average $\bar{g}_n$ is used to approximate the expectation, which converges almost surely if one could apply a strong law of large numbers

$$\bar{g}_n = \frac{1}{n} \sum_{i=1}^{n} g(x_i) \xrightarrow{a.s.} E_f[g(x)]. \quad \text{(B.2)}$$
This is referred to as *Monte Carlo integration* and could be used to approximate analytically infeasible integrals. The asymptotic variance of $\bar{g}_n$ is given by

$$\text{Var} (\bar{g}_n) = \frac{1}{n} \int_x (g(x) - E_f [g(x)])^2 f(x) \, dx$$  \hspace{1cm} (B.3)

which itself is estimated by

$$v_n \equiv \frac{1}{n^2} \sum_{j=1}^{n} (g(x_j) - \bar{g}_n)^2.$$  \hspace{1cm} (B.4)

If the Central Limit Theorem is applicable, then

$$\frac{\bar{g}_n - E_f [g(x)]}{\sqrt{v_n}} \sim \mathcal{N}(0, 1).$$  \hspace{1cm} (B.5)

As an example, we consider the integral $\int_0^1 g(x) f(x) \, dx$ where $g(x) = (\cos(50x) + \sin(20x))^2$ and $f(x) = 1$ (this is Example 3.4 in Robert & Casella, 2004 and Example 3.3 in their 2010 book),

$$\int_0^1 ((\cos(50x) + \sin(20x))^2) \, dx.$$  \hspace{1cm} (B.6)
This integral can actually be solved analytically\footnote{Mathematica gives the answer \[ \frac{8240 - 105 \sin(40) + 42 \sin(100) + 280 \cos(30) - 120 \cos(70)}{8400} \approx 0.96520. \]} but it could be interesting to approximate it with simulations.

We draw \( n \) i.i.d. random variables from a uniform distribution on \([0, 1]\), \( U_1, U_2, \ldots, U_n \) and take the average of \( g(u_i) \) to approximate the integral, i.e., we use (B.2), \( \sum g(u_i)/n \approx \int_0^1 g(x) \, dx \). Algorithm B.3 (taken from Robert & Casella, 2010, p. 66) gives an R implementation. The upper part of Figure B.3 displays the function \( g(x) \) as defined above, and the lower part displays the running average of the simulations (a new average for each new simulation \( i \) using all simulations up to \( i \)). The 10,000’th average (i.e, of the last simulated value) is 0.96861, which is quite close to the expectation 0.96520 (see Footnote 16).

\begin{verbatim}
# Specifying the function:
g=function(x){(cos(50*x)+sin(20*x))^2}

# graphical parameters:
par(mar=c(2,2,2,1),mfrow=c(2,1))
curve(g,lwd=2)
integrate(g,0,1)
x=g(runif(10^4))

# (runif generates random uniform drawings)

# Estimated integral and SE:
estint=cumsum(x)/(1:10^4)
# (cumsum gives the cumulated sum)
esterr=sqrt(cumsum((x-estint)^2))/(1:10^4)

# The plot:
plot(estint,type="l",lwd=2,ylim=mean(x)+20*c(-esterr[10^4],
esterr[10^4]))
# 2 SE:
lines(estint+2*esterr,col="gold",lwd=2)
lines(estint-2*esterr,col="gold",lwd=2)

# Line for the true integral:
abline(h=0.96520, untf = FALSE, col="blue")
\end{verbatim}
B.3 Importance Sampling

If we rewrite (B.1) by multiplying and dividing by the proposal distribution \( q(x) \)

\[
E_f[g(x)] = \int_X g(x)f(x) \, dx \\
= \int_X g(x) \frac{f(x)}{q(x)} q(x) \, dx = E_q \left[ g(x) \frac{f(x)}{q(x)} \right]
\]

we can, similarly to (B.2), approximate the expectation with an average

\[
E_f[g(x)] = E_q \left[ g(x) \frac{f(x)}{q(x)} \right] \approx \frac{1}{n} \sum_{i=1}^{n} g(x_i) \frac{f(x_i)}{q(x_i)}
\]

(B.8)

where \( X_1, X_2, \ldots, X_n \) are i.i.d. random variables with density \( q(x) \). This is called importance sampling; it converges for the same reason that \( \bar{g}_n \) converges, no matter which \( q(x) \) is chosen (as long as the support of \( q \) is a superset of the support of \( f \) (Robert & Casella, 2004, p. 92)). The weight \( f(x)/q(x) \) is called the importance function (hence the name importance sampling).

It turns out (perhaps somewhat surprisingly) that the variance can be much lower when sampling from \( q(x) \) instead of directly from \( f(x) \) (see, e.g., Robert & Casella, 2004, Example 3.8). One example is when we are interested in tail events of a standard normal, say \( \Pr(Z > 4.5) \), which would need a lot of iterations if simulated directly from \( \mathcal{N}(0, 1) \); it could more accurately be simulated using importance sampling from a truncated exponential distribution (see Robert & Casella, 2004, Example 3.11 and Problem 3.16).

B.4 Markov Chains

This subsection will be a very brief introduction to discrete time (but continuous state-space) Markov chains, covering only the essentials needed for MCMC. For
A (discrete time) Markov chain is a random process $X(0), X(1), \ldots, X(n) = \{X(t)\}$ which possesses the Markov property, i.e., the probability distribution of $X(t)$ given $X(t-1), X(t-2), \ldots$ (called the transition kernel $K(\cdot, \cdot)$), depends only on $X(t-1)$,

$$X(t+1) \mid X(0), X(1), \ldots, X(t-1), X(t) \sim K(X(t), X(t+1)).$$  \hspace{1cm} (B.9)

The Markov chains used in MCMC often have—by construction—a stationary probability distribution $f$. This means that there exist a distribution $f$ such that if $X(t-1) \sim f$, then $X(t) \sim f$ (Robert & Casella, 2010, p. 169), that is, the stationary distribution and the kernel satisfies

$$\int_{\mathcal{X}} K(x, y) f(x) \, dx = f(y).$$  \hspace{1cm} (B.10)

One condition that must hold for stationarity is irreducibility, which means that there is a positive probability that any region in the state space is reached regardless of the starting value $X(0)$. One sufficient condition is that $K(x, \cdot) > 0$ everywhere (Robert & Casella, 2010, p. 169). The existence of a stationary distribution also means that most of the chains used in MCMC algorithms are recurrent (Robert & Casella, 2010, p. 169), meaning that they will return to any arbitrary non-negligible set an infinite number of times. When the stationary chain is recurrent, the stationary distribution is also the limiting distribution, meaning that the limiting distribution of $X(t)$ is $f$ (the stationary distribution) for almost any starting value $X(0)$ (Robert & Casella, 2010, p. 169). Ergodicity is the name of this property; meaning that an ergodic Markov chain with stationary distribution $f$—if run for long enough—will eventually generate simulations from $f$ (Robert & Casella, 2010, p. 169).

The Law of Large Numbers (LLN) can be applied in the same form as (B.2), which was used for Monte Carlo integration—even though the sample is no longer i.i.d. When applied for Markov Chains, the LLN is often called the Ergodic Theorem. The empirical average converges to the expectation

$$\frac{1}{n} \sum_{i=1}^{n} g(X(i)) \longrightarrow E_f [g(X)]$$  \hspace{1cm} (B.11)

where $g$ is an integrable function.

**B.5 The Metropolis-Hastings Algorithm**

The original Metropolis algorithm is due to Metropolis, Rosenbluth, Rosenbluth, Teller, and Teller (1953), and was extended by Hastings (1970) to, what is known as,
the Metropolis-Hastings algorithm. For an introduction, see Chib and Greenberg (1995) and Robert and Casella (2004, Chapter 7).

Markov Chain Monte Carlo methods can be described in the following way: We have a target density $f$ that we want to simulate from; a Kernel $K$ is then built with the target density $f$ as its stationary distribution, and a Markov chain $\{X^{(t)}\}$ is generated with this kernel so that the chain’s limiting distribution is $f$. We can then approximate integrals with (B.11) (Robert & Casella, 2010, p. 170).

The difficulty lies in constructing a kernel $K$ that works for an arbitrary target density $f$, but both the Metropolis-Hastings algorithm and Gibbs sampling (will be described in Subsection B.6) solves this problem.

Robert and Casella (2010, p. 170) gives the following intuition for the Metropolis-Hastings algorithm. Given a target density $f$, a conditional density $q(y|x)$ is chosen with the conditions that: (i) the ratio $f(y)/q(y|x)$ is known up to a constant independent of $x$, and (ii) the support of $q(\cdot|x)$ covers that of $f$ (so that all of $f$ can be explored when simulating from $q$). In practice we want to choose a $q$ that is easy to simulate from. As mentioned above, one feature of the Metropolis-Hastings algorithm is that, for any given $q$, it can be used to construct a Metropolis-Hastings kernel with $f$ as its stationary distribution.

Algorithm B.4 describes the Metropolis-Hastings algorithm.


Given $x^{(t)}$,

1. Generate $Y_t \sim q(y|x^{(t)})$

2. Take

$$X^{(t+1)} = \begin{cases} Y_t & \text{with probability } \alpha(x^{(t)}, Y_t) \\ x^{(t)} & \text{with probability } 1 - \alpha(x^{(t)}, Y_t) \end{cases}$$

where $\alpha(x^{(t)}, Y_t)$ is the Metropolis-Hastings acceptance probability

$$\alpha(x^{(t)}, Y_t) = \min \left( \frac{f(y)}{f(x)} \frac{q(x|y)}{q(y|x)} , 1 \right)$$

The Metropolis-Hastings algorithm assures that $f$ is the stationary distribution of the chain $\{X^{(t)}\}$, since the detailed balance condition

$$f(x)K(y|x) = f(y)K(x|y) \quad \text{(B.12)}$$

is satisfied (Robert & Casella, 2010, p. 172).

An example of a Metropolis-Hastings algorithm follows. Say that the target density is $B(2.7, 6.3)$ and our candidate $q$ is the uniform density over $[0, 1]$ (cf., Figure B.2, for the previous example with the same target and candidate densities
R implementation. The top of Figure B.4 shows all 5000 iterations of the chain, 
while the bottom zooms in on 300 iterations near the end (4500–4800). Figure B.5 
shows histograms with overlaid densities for the Metropolis-Hastings sample and an 
i.i.d. sample drawn directly from the beta distribution using the rbeta command. 
As can be seen the fit is quite close.

Algorithm B.5 R code for Metropolis-Hastings algorithm – beta example (from Robert & 

```r
nsim = 5000
X = rep(runif(1), nsim) # initialize the chain
for (i in 2:nsim) {
  Y = runif(1)
  alpha = dbeta(Y, 2.7, 6.3) / dbeta(X[i - 1], 2.7, 6.3)
  X[i] = X[i - 1] + (Y - X[i - 1]) * (runif(1) < alpha)
}
```

Figure B.4 A sample of 5000 draws from Metropolis-Hastings algorithm for the target density 
\( B(2.7, 6.3) \) with a \( U_{[0,1]} \) proposal (Algorithm B.5). (Left) All iterations. (Right) Zooming in on 
iterations 4500 to 4800.

B.6 Gibbs Sampling

The Gibbs sampler is formally a special case of the Metropolis-Hastings algorithm 
with an acceptance probability equal to 1 (Robert & Casella, 2004, Therorem 10.13), 
but it has different historical motivation and methodology.

Introductions to Gibbs sampling can be found in Casella and George (1992), 
Robert and Casella (2004, Chapters 8–10, but earlier chapter are also useful, 
especially 6 and 7) and Robert and Casella (2010, Chapter 7). Gibbs sampling was 
introduced by Geman and Geman (1984)—for image-processing models, and it was 
popularized in statistics by Gelfand and Smith (1990). Robert and Casella (2010, 
p. 200) also mentions earlier contributions to statistics by Tanner and Wong (1987) 
and Besag and Clifford (1989).
B.6.1 The two-stage Gibbs sampler

Robert and Casella (2010, p. 200) describes a basic bivariate Gibbs sampler by the following steps. For two random variables $X$ and $Y$, with the joint density $f(x, y)$ and conditional densities $f_{Y|X}$ and $f_{X|Y}$, a Markov chain $(X_t, Y_t)$, is generated by:

Take $X_0 = x_0$; for $t = 1, 2, \ldots$, generate:

1. $Y_t \sim f_{Y|X}(:,|x_{t-1})$;
2. $X_t \sim f_{X|Y}(:,|y_t)$.

Since both Step 1 and Step 2 simulate from the true conditionals, if $(X_t, Y_t) \sim f$, then $(X_{t+1}, Y_{t+1}) \sim f$ as well.

A simple example with a bivariate normal distribution follows (Robert & Casella,
2010, Example 7.1).\textsuperscript{17}

\[
\begin{bmatrix}
X \\
Y
\end{bmatrix} \sim \mathcal{N}(\begin{bmatrix}
0 \\
0 \\
1
\end{bmatrix}, \begin{bmatrix}
1 & \rho \\
\rho & 1
\end{bmatrix})
\]

(B.13)

For the Gibbs sampler we then generate for $t = 1, 2, \ldots$

1. $Y_{t+1} | x_t \sim \mathcal{N}(\rho x_t, 1 - \rho^2)$
2. $X_{t+1} | y_{t+1} \sim \mathcal{N}(\rho y_{t+1}, 1 - \rho^2)$

According to Robert and Casella the subchain $(X_t)_t$ is then

\[X_{t+1} | X_t = x_t \sim \mathcal{N}(\rho^2 x_t, 1 - \rho^4)\]

which can be rewritten using recursion to

\[X_t | X_0 = x_0 \sim \mathcal{N}(\rho^2 t x_0, 1 - \rho^{4t})\]

and letting $t \to \infty$, this converges to a standard normal $\mathcal{N}(0, 1)$. Figure B.6 shows 50 draws (no burn-in) from this example, with $\rho = 0.5$, using the command \texttt{rbiNormGibbs} in the \texttt{R} package \texttt{bayesm}.

In the above example, the Markov chain generated by the Gibbs sampler converges to the joint distribution $f$, leading both subchains $(X_t)_t$ and $(Y_t)_t$ to reach their marginal distributions as well.

\subsection*{B.6.2 The multistage Gibbs sampler}

The Gibbs sampler can be extended to a multivariate setting for a random variable $X \in \mathcal{X}$, $X = (X_1, X_2, \ldots, X_p)$, where the $X_i$’s themselves could also be of one

\textsuperscript{17}The joint density is then given by

\[
f(x, y) = \exp\left(\frac{-2 \rho xy + x^2 + y^2}{2(\rho^2 - 1)}\right)
\]

The marginal densities are standard normals,

\[
f(x) = \frac{e^{-x^2/2}}{\sqrt{2\pi}}, \quad f(y) = \frac{e^{-y^2/2}}{\sqrt{2\pi}}
\]

and thus the conditionals are given by

\[
f(X|Y) = \frac{(x-\rho y)^2}{\sqrt{2\pi} \sqrt{1 - \rho^2}}, \quad f(Y|X) = \frac{(y-\rho x)^2}{\sqrt{2\pi} \sqrt{1 - \rho^2}}
\]
or higher dimensions. Using $x_{i:j}$ to denote $x$ excluding $x_i$, and suppose we can simulate from the conditional densities $f_1, f_2, \ldots, f_p$, i.e., that we can simulate

$$X_i | x_{\backslash i} \sim f_i(x_i | x_{\backslash i})$$

for $i = 1, 2, \ldots, p$. The **Gibbs sampler** is then given by (Robert & Casella, 2010, Algorithm 8)

**Algorithm B.6** The Gibbs sampler.

At iteration $t = 1, 2, \ldots$, given $x^{(t)} = (x_1^{(t)}, x_2^{(t)}, \ldots, x_p^{(t)})$, generate

1. $X_1^{(t+1)} \sim f_1(x_1 | x_{\backslash 1}^{(t)})$;
2. $X_2^{(t+1)} \sim f_2(x_2 | x_{\backslash 2}^{(t)})$;
   
   \vdots
3. $X_p^{(t+1)} \sim f_p(x_p | x_{\backslash p}^{(t)})$.

One cycle of Step 1 to Step $p$ (from $t$ to $t+1$) is called a scan or a sweep. Under regularly conditions, the draws from the Gibbs sampler converges to draws from the target density at a geometric rate (Kim et al., 1998).

The advantage of the Gibbs sampler compared to Metropolis-Hastings and other algorithms is that instead of simulating from a $p$ dimensional distribution
(which may be very high-dimensional in some applications), we can simulate from
$p$ univariate distributions (Robert & Casella, 2010).
C Probability Distributions

This chapter is a summary of a few probability distributions used in this thesis; the density and first two moments are given. The main reference used is Robert and Casella (2004).

C.1 Normal

If the random variable \( X \) is normally distributed, this is written as \( X \sim \mathcal{N}(\mu, \sigma^2) \), where \( \mu \in \mathbb{R} \) is the expectation and \( \sigma^2 > 0 \) is the variance. The density is given by

\[
f(x|\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp \left( -\frac{1}{2\sigma^2}(x - \mu)^2 \right). \tag{C.1}\]

The expectation and variance are given by

\[
E[X] = \mu \quad \text{Var}(X) = \sigma^2.
\]

Note that the alternative parameterization of using the precision \( \tau = \sigma^{-2} > 0 \) (i.e., the reciprocal of the variance) instead of the variance is common in Bayesian modeling, i.e., the density is given by

\[
f(x|\mu, \tau) = \left( \frac{\tau}{2\pi} \right)^{1/2} \exp \left( -\frac{\tau}{2}(x - \mu)^2 \right). \tag{C.2}\]

If a \( k \) random vector \( \mathbf{X} \) is normally distributed, this is denoted by \( \mathbf{X} \sim \mathcal{N}(\mathbf{\theta}, \mathbf{\Sigma}) \), where \( \mathbf{\theta} \in \mathbb{R}^k \) and \( \mathbf{\Sigma} \) is a symmetric positive definite \( k \times k \) matrix. Its density is given by

\[
f(\mathbf{x}|\mathbf{\theta}, \mathbf{\Sigma}) = (\det \mathbf{\Sigma})^{-1/2}(2\pi)^{-k/2} \exp \left( -\frac{1}{2}(\mathbf{x} - \mathbf{\theta})^\top \mathbf{\Sigma}^{-1}(\mathbf{x} - \mathbf{\theta}) \right) \tag{C.3}\]

C.2 Gamma

If a random variable \( X \) is gamma distributed, this is denoted by \( X \sim \mathcal{G}(\alpha, \beta) \), where \( \alpha > 0 \) and \( \beta > 0 \). Its density is given by

\[
f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} x^{\alpha - 1} e^{-\beta x} \mathbb{1}_{[0,\infty)}(x) \tag{C.4}\]

where \( \Gamma(\cdot) \) denotes the gamma function (for some useful theorems and properties see, e.g., Rudin, 1976, pp. 192–195)

\[
\Gamma(\alpha) = \int_0^\infty t^{\alpha - 1} e^{-t} \, dt \tag{C.5}\]

\footnote{Note that the alternative notation \( X \sim \mathcal{G}(\alpha, \lambda) \) with \( \lambda = 1/\beta \) (and the density defined accordingly) is also common.}
and $\mathbb{1}_{[0,\infty)}(x)$ is the indicator function which takes the value 1 if $x \in [0, \infty)$ and 0 otherwise.

The gamma distribution nests some other important distributions, e.g., the chi-squared distribution $\chi^2_\nu$ with $\mathcal{G}(\nu/2, 1/2)$, the exponential distribution $\mathcal{E}(\beta)$ with $\mathcal{G}(1, \beta)$ and the Erlang distribution with $\mathcal{G}(\alpha, 1)$.

The expectation and variance are given by

$$E[X] = \frac{\alpha}{\beta} \quad \text{Var}(X) = \frac{\alpha}{\beta^2}.$$  

### C.3 Inverse Gamma

If a random variable $Y$ follows a gamma distribution $Y \sim \mathcal{G}(\alpha, \beta)$, then—as suggested by the name—$X = Y^{-1}$ follows an inverse gamma distribution, which is denoted by $X \sim \mathcal{IG}(\alpha, \beta)$, where $\alpha > 0$ and $\beta > 0$. Its density is given by

$$f(x|\alpha, \beta) = \frac{\beta^\alpha}{\Gamma(\alpha)} \frac{e^{-\beta/x}}{x^{\alpha+1}} \mathbb{1}_{[0,\infty)}(x)$$  

where $\Gamma(\cdot)$ denotes the gamma function (see (C.5)) and $\mathbb{1}_{[0,\infty)}(x)$ is the indicator function which takes the value 1 if $x \in [0, \infty)$ and 0 otherwise.

The expectation and variance are given by

$$E[X] = \frac{\beta}{\alpha - 1} \quad \text{Var}(X) = \frac{\beta^2}{(\alpha - 1)^2(\alpha - 2)}$$  

where $\alpha > 1$ and $\alpha > 2$ are needed for the expectation and variance, respectively, to exist.

### C.4 Beta

If a random variable $X$ is beta distributed, this is denoted by $X \sim \mathcal{B}(\alpha, \beta)$, where $\alpha > 0$ and $\beta > 0$. Its density is given by

$$f(x|\alpha, \beta) = \frac{x^{\alpha-1}(1-x)^{\beta-1}}{B(\alpha, \beta)} \mathbb{1}_{[0,1]}(x)$$  

where the beta function $B(\cdot, \cdot)$ is defined as

$$B(\alpha, \beta) = \int_0^1 t^{\alpha-1}(1-t)^{\beta-1} \, dt = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha + \beta)}$$  

53
in the last equality expressed in terms of gamma functions $\Gamma(\cdot)$ (see (C.5)) and $I_{[0,1]}(x)$ is the indicator function which takes the value 1 if $x \in [0,1]$ and 0 otherwise.

The expectation and variance are given by

$$E[X] = \frac{\alpha}{\alpha + \beta} \quad \text{Var}(X) = \frac{\alpha \beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}.$$ 

A beta distributed random variable $X \sim \mathcal{B}(\alpha, \beta)$ can be written as a transformation of two gamma distributed random variables $Y_1 \sim \mathcal{G}(\alpha, 1)$ and $Y_2 \sim \mathcal{G}(\beta, 1)$, $X = Y_1/(Y_1 + Y_2)$.