NTNU
Norwegian University of
Science and Technology

Parameter Estimation in Extreme Value Models with Markov Chain Monte Carlo Methods

Eivind Blomholm Gausland

Master of Science in Physics and Mathematics
Submission date: June 2010
Supervisor: Arvid Næss, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The candidate should study Markov Chain Monte Carlo methods in extreme value models. In particular, the candidate should consider how to estimate model parameters with Markov Chain Monte Carlo methods, with synthetic data generated by spectral densities.

Assignment given: 20. January 2010
Supervisor: Arvid Næss, MATH

Preface

This Master's thesis completes my five year Master of Science program at the Norwegian University of Science and Technology (NTNU). The last 20 weeks I have studied how to estimate parameters in an extreme value distribution with stochastic simulation.

I would like to thank my supervisor Professor Arvid Næss for his help and good advice during this period. Professor Håkon Tjelmeland also deserves credit, for informative and useful discussions about stochastic simulation.

I have had much fun with my classmates during this period, and I would like to thank them all, especially Daniel Høyer Iversen and Morten Valberg for creating a good working environment at our office.

Eivind Blomholm Gausland

Trondheim, June 16, 2010

Abstract

In this thesis I have studied how to estimate parameters in an extreme value model with *Markov Chain Monte Carlo* (MCMC) given a data set. This is done with synthetic Gaussian time series generated by spectral densities, called spectrums, with a "box" shape. Three different spectrums have been used. In the acceptance probability in the MCMC algorithm, the likelihood have been built up by dividing the time series into blocks consisting of a constant number of points. In each block, only the maximum value, i.e. the extreme value, have been used. Each extreme value will then be interpreted as independent. Since the time series analysed are generated the way they are, there exists theoretical values for the parameters in the extreme value model. When the MCMC algorithm is tested to fit a model to the generated data, the true parameter values are already known. For the first and widest spectrum, the method is unable to find estimates matching the true values for the parameters in the extreme value model. For the two other spectrums, I obtained good estimates for some block lengths, others block lengths gave poor estimates compared to the true values. Finally, it looked like an increasing block length gave more accurate estimates as the spectrum became more narrow banded. A final simulation on a time series generated by a narrow banded spectrum, disproved this hypothesis.

Contents

Chapter 1

Introduction

Extreme value theory is not the oldest and most explored part of statistical theory. In 1928 the first methodology was proposed by Fisher and Tippet. Later, in 1943, Gnedenko contributed to the first extreme value theorem, the Fisher-Tippet-Gnedenko theorem. This is one of two approaches in extreme value theory. In the second approach, only data that crosses a given threshold is taken into account, while all data is used in the first approach.

Since extreme values rarely occur, it is difficult to find or estimate a distribution for the extreme values, which gives correct values for the extremes. This is because the extreme values belongs to the tail of the distribution. The values are often estimated with basis in former observed values, where one extrapolates the series of observations to estimate the return levels. The return levels are defined as how often a level is exceeded during a given time interval. An example is in analysis of wave data, where the decade wave or hundred-year wave may be estimated. For engineers it may be extremely useful to know the estimates for the return levels for different types of events, for instance how much wind a structure have to withstand.

To be able to extrapolate, one needs to assume that the data, or more precisely, the extreme events, are coming from a given statistical model. Then the statistical model have to be fitted to the data, so one can extrapolate to the given level. This means that some parameters in a given distribution have to be estimated. In this thesis, this is done with so called *Markov Chain Monte Carlo* (MCMC) methods. First time series are generated, where one knows the "true" values in a given distribution. One may then "pretend" that the values are unknown, and estimate them by MCMC methods. Since the "true" values are already known, one can control whether the algorithm gives a good or bad estimate for the parameters.

The thesis is organized as follows.

In the 2nd chapter one can first read about the classical extreme value theory, where one tries to find an appropriate distribution for a series of observations. It is assumed that the tail of the distribution can be fitted asymptotically. This leads to the generalized extreme value distribution, denoted GEV. Another model which often has been used is the generalized Pareto family of distributions. One assumes here that asymptotically, the excess values over a given level will follow the generalized Pareto model, as long as the underlying distribution belongs to the generalized extreme value family. Both these models assume that the observed data is independent.

In chapter 3 some theoretical properties of Markov chains are described. This is very central in stochastic simulation, where one will construct a Markov chain, whose equilibrium distribution will be the posterior distribution for the parameters to be estimated. This is described further in chapter 4, where the algorithm used, the Metropolis-Hastings algorithm is presented.

Chapter 5 shows how the time series are generated, and some of their properties. The data used in the analysis, and the distribution of the extreme events in the data are also presented.

Chapter 6 describes how the algorithm is implemented in Matlab, and some discussion about different MCMC schemes and their advantages and disadvantages. In chapter 7 all analysis on the different time series are done. A discussion about the results, the prior influence and different likelihood structures follows.

Chapter 2

Extreme Value Theory

The following deduction follows Stuart Coles book[1].

2.1 Classical extreme value theory

Start looking at a sequence of independent random variables, $X_1, ..., X_n \sim F_X(x)$. The extreme value is then described as

$$M_n = \max\{X_i | i = 1, ..., n\}.$$

Since all X_i's are independent, the M_n is distributed as

$$F_{M_n}(x) = P(M_n \leq x) = P(X_1 \leq x, ..., X_n \leq x) \overset{iid}{=} (F_X(x))^n$$

In practice, this does not work. F is unknown, but can be estimated from the data. However, this estimation can be inadequate. Since $F_X(x) < 1$, then $(F_X(x))^n \to 0$ as $n \to \infty$, which does not give any meaning. To solve this problem, $F_X(x)$ is rescaled by suitable constants a_n and b_n such that

$$M_n^* = \frac{M_n - b_n}{a_n}.$$

The constants are chosen to stabilize the location and scale of M_n^* as n increases. This is the basis for the extremal type theorem.

2.1.1 Generalized extreme value distribution

Theorem 1. *If there exists a sequence of constants* $\{a_n > 0\}$ *and* $\{b_n\}$ *such that*

$$P\Big(\frac{M_n - b_n}{a_n} \leq z\Big) \to G(z) \quad \text{as } n \to \infty, \tag{2.1}$$

where G is a non-degenerate distribution function, then G belongs to one of the following families:

$$
\begin{aligned}
I: G(z) &= \exp\left\{-\exp\{-\frac{z-b}{a}\}\right\}, \quad -\infty < z < \infty; \\
II: G(z) &= \begin{cases} 0, & z \leq b, \\ \exp\{-(\frac{z-b}{a})^{-\alpha}\}, & z > b; \end{cases} \\
III: G(z) &= \begin{cases} \exp\{-(-\frac{z-b}{a})^{\alpha}\}, & z < b, \\ 1, & z \geq b, \end{cases}
\end{aligned}
\tag{2.2}
$$

for parameters $a > 0$, b and, in the case of families II and III, $\alpha > 0$.

These distributions are known as Gumbel (I), Fréchet (II) and inverse Weibull (III). The theorem states that when M_n is rescaled, it must have one of the three distributions above as limiting distribution. The properties of these distributions are different, also for the tail behaviour, which is of special interest. Different tail behaviour gives totally different extreme value predictions. Considering the upper end point in the three limiting distributions, one observes that the upper end point in the Weibull distribution is finite, while it is ∞ in both Gumbel and Fréchet. However, the density for the Gumbel distribution decays exponentially, and for Fréchet it decays polynomial, so the Fréchet distribution has a long tail.

Further, all of the distributions in expression (2.2) can be expressed in one distribution, namely the generalized extreme value distribution

$$G(z) = \exp\left\{-[1 + \xi(\frac{z-\mu}{\sigma})]^{-\frac{1}{\xi}}\right\}.$$

To decide which model fits the data best, one can do inference on ξ. A Fréchet distribution corresponds with $\xi > 0$ and Weibull corresponds to $\xi < 0$. The limiting case when $\xi \to 0$ gives a Gumbel distribution, which is written as

$$G(z) = \exp\left\{-\exp\{-\frac{z-\mu}{\sigma}\}\right\}.$$

It is the data themselves who decide which model and tail behaviour that fits best, through inference on ξ. The parameters can be estimated with maximum likelihood estimation, profile likelihood estimation, or conventional MCMC methods.

A different interpretation is to block the data. If the observations are of a phenomenon with daily observations over several years, the data can be divided into

blocks, where each block corresponds to a year. By assuming (2.1) for large enough n,

$$\begin{aligned} P\Big(M_n \leq z\Big) &\approx G\Big(\frac{z-b_n}{a_n}\Big) \\ &= G^*(z), \end{aligned} \tag{2.3}$$

where $G^*(z)$ is another member of the GEV family. This means, that if M_n^* can be approximated by a distribution from the GEV family for large enough n, M_n can also be approximated by a different distribution from the GEV family.

2.1.2 Peaks over threshold

Another way to do this, is to look at the number of values above a given threshold η. Then one consider all values over η as the extreme values. If now, the Theorem above and (2.3) is fulfilled, then for large enough η, the distribution of $(X-\eta)$ given $X > \eta$ is

$$H(y) = 1 - (1 + \frac{\xi y}{\tilde{\sigma}})^{-\frac{1}{\xi}},$$

where $\{y = (X-\eta)|y > 0 \wedge (1+\frac{\xi y}{\tilde{\sigma}}) > 0\}$ and $\tilde{\sigma} = \sigma + \xi(\eta - \mu)$. This is called the generalized Pareto distribution. The constants corresponds to the constants in the generalized extreme value distribution of M_n. A distribution satisfying this, belongs to the generalized Pareto family of distributions.

The weakness in this family of distributions is if the observations occur in clusters, which is very usual. If one exceedance above threshold η is obtained, it is often followed by another exceedance, which makes the observations dependent. This can be solved by the method declustering. This method detects the clusters and filter out dependent observations only retaining the maximal observation in each cluster, so the remaining exceedances can be assumed independently. Then one may fit the generalized Pareto model to the maxima of each cluster.

Another difficulty is to determine a suitable threshold. If the threshold is too high, there will be few excesses and it will be difficult to find a limiting distribution. On the other hand, if the threshold is too low, there will be to many excesses, which also leads to difficulties in finding a proper distribution. As mentioned, the parameters in the different extreme value models can be estimated by many different methods. In this thesis, MCMC is used to simulate the posterior density

for each parameter, and estimate values for the parameters from this posterior density obtained with MCMC. In the next chapter, some background theory on Markov chains are described.

Chapter 3

Markov Chain Theory

Markov chain Monte Carlo is a term used to describe different types of algorithms, which all have the same goal. The goal is to be able to generate samples from a distribution $\pi(x)$. A formal definition from [2] is

Definition 1 (Markov chain Monte Carlo)**.** *A Markov chain Monte Carlo (MCMC) method for the simulation of a distribution π is any method producing an ergodic Markov chain where the stationary distribution is π.*

For many situations, it is not possible to sample with direct methods, often because it is impossible to sample from the desired distribution. If one is interested in deciding some properties of a distribution $\pi(x)$ e.g.

$$\mu = \mathrm{E}_\pi[f(x)] = \int f(x)\pi(x)\mathrm{d}x.$$

This sum may not be possible to compute in practice, but one can try to estimate

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} \mathrm{E}[f(x_i)],$$

where $x_1, ...x_N$ are samples from $\pi(x)$. Now assume that it is not possible to generate x_i from $\pi(x)$ directly.

The idea is to construct a Markov chain $\{X_i\}_{i=1}^{\infty}$ with $\pi(x)$ as the limiting distribution,

$$\lim_{i\to\infty} P(X_i = x) = \pi(x).$$

If the Markov chain is simulated for many iterations, then for large enough m, x_m, x_{m+1},... will essentially be samples from $\pi(x)$. This is now referred to as the

target distribution. One may then estimate μ by

$$\hat{\mu} = \frac{1}{n} \sum_{i=m}^{m+n} f(x_i).$$

To decide when m is large enough is a discussion of whether the chain has converged or not. To do this, and to make sure that the constructed Markov chain has an unique limiting distribution, some theory and properties of Markov chains have to be introduced.

3.1 Markov chain theory

Given a series of states $x_1, ..., x_n$, the Markov property is defined as the property of being dependent only on the previous state. This is the same as saying $\mathrm{P}(x_{n+1})$ conditioned on all previous states is the same as $P(x_{n+1}|x_n)$. Thus, the series of states is a Markov chain, which only depends on the previous state.

To define a Markov chain, first define a stochastic process. Let X_n be a random variable. A stochastic process is a collection of random variables,

$$\{X_n | n \in \mathbb{N}\},$$

where $\mathbb{N}$ is a countable set. The index n is often interpreted as time.

Now, a Markov chain is a stochastic process, where each state only depends on the previous state. This means that the future is independent of the past. This can be written as

$$P(X_{n+1} \in A | X_n = x_n, X_{n-1} = A_{n-1}, ..., X_1 = A_1) = P(X_{n+1} \in A | X_n = x_n), \quad \forall\, n$$

for all sets $A_1, ..., A_{n-1}, A \subset \mathcal{X}$. An example of a Markov chain is the auto regressive process of order 1, denoted AR(1), with standard normal errors[3]. This is expressed as

$$x_n = \phi x_{n-1}, \qquad \varepsilon \sim N(0, \sigma^2), \qquad |\phi| < 1$$

where index n represents time. This model can be seen as a random walk process, where the current state only depends on the previous state. A graphical model can be seen in figure 3.1. A more obvious expression is

$$x_n | x_1, ..., x_{n-1} \sim N(\phi x_{n-1}, \sigma^2).$$

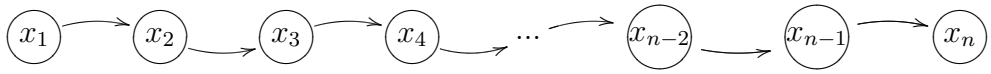


Figure 3.1: Bayesian model for the AR(1) model.

Here, x_n and x_s with $1 \leq s < n \leq N$ are conditionally independent given $\{x_{s+1}, ..., x_{n-1}\}$ if $n - s > 1$, which also implies the Markov property.

Given a Markov chain, the transition of the chain from state x_n to state x_{n+1} is given by the transition kernel, K, whose formal definition is taken from [4].

Definition 2 (Transition kernel)**.** *A transition kernel is a function K defined on $\mathcal{X} \times \mathcal{B}(\mathcal{X})$ such that*

$$\begin{aligned} &\textit{(i)} \; \forall \; x \in \mathcal{X}, K(x, \cdot) \textit{ is a probability measure;} \\ &\textit{(ii)} \; \forall \; A \in \mathcal{B}(\mathcal{X}), K(\cdot, A) \textit{ is measurable.} \end{aligned}$$

Here, the *measure* is the Lebesgue measure. This can be read more in detail in [5]. The chain must be homogeneous, that is, it must be independent of n. For a discrete $\mathcal{X}$, the transition kernel simplifies to a transition matrix P with elements

$$P_{xy} = P(X_n = y | X_{n-1} = x), \qquad x, y \in \mathcal{X}$$

from state x to state y. Note that P can only take values in $[0, 1]$, and must satisfy

$$\begin{aligned} P(x, y) \geq 0, \quad \forall \; x, y \in \mathcal{X}, \\ \sum_{y \in \mathcal{X}} P(x, y) = 1, \quad \forall \; x \in \mathcal{X}, \end{aligned}$$

to be called a transition probability.

The transition probability should allow the Markov chain to move freely all over the sample space, which is the definition of irreducibility:

Definition 3 (ϕ-irreducible)**.** *Given a measure ϕ, a Markov chain with transition kernel K defined on $\mathcal{X} \times \mathcal{B}(\mathcal{X}) \to [0, 1]$ is ϕ-irreducible if, for every $A \in \mathcal{B}(\mathcal{X})$ with $\phi(A) > 0$, there exists an n such that $K^n(x, A) > 0 \; \forall \; x \in \mathcal{X}$.*

This means that it is possible to visit every state in a finite number of transitions. This property does not give an equal probability of visiting every set in the sample

space, but the chain has opportunity to do so. After the chain has evolved for some time, some set will be visited more often than other "less important" sets. The introduction of recurrence will help to ensure that the more important sets will be visited often enough, hence increase the stability of the chain.

Definition 4 (Recurrence)**.** *A Markov chain is recurrent if, given a probability measure ϕ, for every*

$$A \in \mathcal{B}(\mathcal{X}) \text{ such that } \phi(A) > 0, \quad E[\eta_A] = \infty \; \forall \; x \in A,$$

where E is the expectation operator, and η_A is the number of passages of the chain in A. This means that a state $x \in A$ is recurrent if the Markov chain, starting in x, have probability 1 of returning to state x. However, if the state x have a non-negative probability less than 1 of returning, the state is said to be transient. The expected number of passages in A is then a finite number. Introduce the hitting time as the time between two visits of a state $x \in A$. If the mean of this hitting time is finite, the state x is said to be positive recurrent. Otherwise, the state is null recurrent. This is a very important result, as will be seen later. Another important concept is aperiodicity. This is defined as

$$d(x) = \text{g.c.d.}\{n \geq 1 | K^n(x, x) > 0\}, \qquad \forall x \in \mathcal{X},$$

where g.c.d. are the greatest common divisor. The interpretation of this, is that $d(x)$ is the period for a possible return to state x, already being in state x. If state x has period 1, the state is said to be aperiodic. This is also true for all states that communicate with x. In an irreducible chain, all states communicate, hence if a state is aperiodic, the chain is aperiodic. A chain which both have the properties of aperiodicity and positive recurrency, is said to be ergodic.

Theorem 2. *A Markov chain has a unique limiting distribution, $\pi(x)$, if the chain is irreducible, aperiodic and positive recurrent. If so, the limiting distribution*

$$\pi(x) = \lim_{i \to \infty} P(X_i = x)$$

is given by

$$\begin{aligned} \pi(y) &= \sum_{x \in \mathcal{X}} \pi(x) \cdot K(x, y) \quad \forall y \in \mathcal{X} \\ \sum_{x \in \mathcal{X}} \pi(x) &= 1. \end{aligned} \tag{3.1}$$

To prove this, introduce the detailed balance condition,

Definition 5 (Detailed balanse condition)**.** *A Markov chain with transition probability P satisfies the detailed balance condition if there exists a function π satisfying*

$$P(y,x)\pi(y) = P(x,y)\pi(x) \qquad \forall x, y \in \mathcal{X} \tag{3.2}$$

This is a sufficient condition for (3.1):

$$\begin{aligned}\sum_{x\in\mathcal{X}} \pi(x)P(x,y) &= \sum_{x\in\mathcal{X}} \pi(y)P(y,x) \\ &= \pi(y) \underbrace{\sum_{x\in\mathcal{X}} P(y,x)}_{=1} \\ &= \pi(y),\end{aligned}$$

which gives a time reversible Markov chain. This can be interpreted as the detailed balance condition, since when in equilibrium, the probability for going from state x to state y is the same as going from state y and back to state x.

3.2 Acceptance probability

Now, the distribution $\pi(x)$ is given. The focus is to solve the equation in the detailed balance condition. This equation gives a large number of unknowns and equations. Solutions here will also be solution for (3.1). This system of equations is very large, and has many solutions, but it is sufficient to find at least one. From this a term called acceptance probability is derived. Introduce the acceptance probability $\alpha(x,y)$ as the probability for accepting a transition from state x to state y, where $\alpha(x,y) \in [0,1]$. Write

$$\begin{aligned}P(x,y) &= Q(x,y)\cdot\alpha(x,y), \quad y \neq x, \\ P(x,x) &= 1 - \sum_{x\neq y} Q(x,y)\cdot\alpha(x,y),\end{aligned}$$

where $Q(x,y)$ is a transition matrix (probability distribution) for some Markov chain. Now, look at the detailed balance condition in equation (3.2), and assume a choice for $Q(x,y)$ has been made. For $x \neq y$, (3.2) can be written as

$$\pi(x)Q(x,y)\alpha(x,y) = \pi(y)Q(y,x)\alpha(y,x),$$

where

$$\alpha(x,y) = r(x,y)\pi(y)Q(y,x) \leq 1$$

is a general solution. This indicates that $r(x,y) = \frac{1}{\pi(y)Q(y,x)} \quad \wedge \quad r(x,y) = r(y,x)$, and gives

$$r(x,y) \leq \frac{1}{\pi(x)Q(x,y)}.$$

Thus

$$\begin{aligned} r(x,y) &\leq \min\left\{\frac{1}{\pi(y)Q(y,x)}, \frac{1}{\pi(x)Q(x,y)}\right\} \\ &= \min\left\{\frac{1}{\pi(y)Q(y,x)}, \frac{1}{\pi(x)Q(x,y)}\right\}, \end{aligned}$$

where $r(x,y)$ is chosen as large as possible. This gives an expression for the acceptance probability

$$\begin{aligned} \alpha(x,y) &= \min\left\{\frac{1}{\pi(x)Q(x,y)}, \frac{1}{\pi(y)Q(y,x)}\right\} \cdot \pi(y) \cdot Q(y,x) \\ &= \min\left\{1, \frac{\pi(y)Q(y,x)}{\pi(x)Q(x,y)}\right\}. \end{aligned}$$

This is central in the algorithm in the next chapter, the Metropolis-Hastings algorithm.

3.3 Continuous case

When the sample space is continuous, the transition probability becomes a transition kernel (see 2. definition). This because for the continuous case, $K(x,\{y\})$ is always null, since $\{y\}$ is a set of measure zero. Therefore, there cannot exist a transition matrix for the continuous case. Instead, $K(x,y)$ can define a distribution such that

$$K(x,y) = P(X_n \leq y | X_{n-1} = x), \qquad \forall x, y \in \mathcal{X}.$$

This distribution should be absolute continuous[1] with respect to y, so the conditional density can be obtained as

$$p(x,y) = \frac{\partial K(x,y)}{\partial y}, \qquad \forall x, y \in \mathcal{X}. \tag{3.3}$$

[1]The reader may see [5] for a definition of absolute continuous functions.

The density $p(x, y)$ is used instead of $Q(x, y)$ in the continuous case. Most of the definitions in last section must also be redefined, such as the detailed balance condition in (3.2), which for the continuous case is

$$\pi(x)p(x, y) = \pi(y)p(y, x), \qquad x, y \in \mathcal{X}, \tag{3.4}$$

where $p(x, y)$ is the density obtained in (3.3). Integrating both sides in (3.4), yields the continuous version of (3.1), which is the stationary distribution $\pi(y)$.

$$\pi(y) = \int_{-\infty}^{\infty} \pi(x)p(x, y)\mathrm{d}x.$$

3.4 How to simulate from the Markov chain

Now, consider a Markov chain $\{X_i\}_{i=1}^{\infty}$ that fulfills all above criteria, so the chain has an unique limiting distribution, $\pi(x)$. Also a transition kernel $p(x, y)$ is defined. First, choose an initial state x_0. Then propose x_1 from the density given by $p(x_0, \cdot)$. This new value is either accepted or rejected. After "enough" iterations, say m, the proposed value from the density $p(x_{m-1}, \cdot)$, is a sample from the limiting distribution, namely $\pi(x)$. In the next chapter, the algorithm used in this thesis will be discussed, the Metropolis-Hastings algorithm, which is a frequently used algorithm in stochastic simulation.

Chapter 4

Metropolis-Hastings Algorithm

The MCMC algorithm used in this thesis for simulating the posterior density, is the Metropolis-Hastings algorithm. This algorithm first showed up in a paper by Nicholas Metropolis et. al. from 1953. In 1970, W. Keith Hastings[6] extended the algorithm from its original scheme, to a more general case.

As all MCMC methods, one needs a target distribution, $\pi(x)$. A Markov chain is then defined, with all the necessary conditions from the last chapter fulfilled. Also one must define a proposal distribution, which gives the transition kernel. This kernel must be made in such a way that $\pi(x)$ is the equilibrium distribution of the chain. There is no answer what is correct or not, but there are several "tricks" for an efficient proposal distribution. In this context, efficient means that the chain will converge after an acceptable amount of iterations, which will be discussed later. Last, all quantities are calculated and inserted into the acceptance probability,

$$\alpha(x, y) = \min\left\{1, \frac{\pi(y)}{\pi(x)}\frac{Q(y, x)}{Q(x, y)}\right\}.$$

The Metropolis-Hastings algorithm is as follows

Algorithm *Metropolis-Hastings*
($*$ Does n iterations of the Metropolis-Hastings algorithm $*$)
1. Start with an arbitrary initial value y_0
2. **for** $i = 1, ..., n$
3. Generate a proposal y^* from some proposal distribution
4. Calculate the acceptance probability, $\alpha(y, y^*)$
5. Draw $u \sim U[0, 1]$
6. **if** $u < \alpha(y, y^*)$
7. $y_i = y^*$
8. **else**
9. $y_i = y_{i-1}$
10. **end**
11. **end**

In the algorithm, y is a single parameter, but the algorithm may also update several parameters at the same time (y is then a parameter vector, often denoted $\boldsymbol{\theta}$). When a parameter vector is proposed and accepted/rejected, it is called a multiple update, while a single parameter proposal/update is called single update in this thesis.

When the chain has evolved for a given number of iterations, the analysis can begin. Exactly how many iterations needed is a big issue in MCMC, and is discussed later under the section Convergence diagnostics. One important element is to check the acceptance rate. In the algorithm, the acceptance probability decide whether the proposed move is accepted or not. If few proposals are accepted, the chain may move very slowly through the parameter space, hence it will use a long time to reach convergence. The acceptance rate is then very low. On the other hand, a very high acceptance rate indicates that the proposed steps are too small, so the chain uses a long time to explore the parameter space. How fast the chain explores the parameter space is called mixing properties of the chain. A chain with bad mixing can use an enormous amount of CPU time to converge, while a chain with good mixing properties can use seconds. A theoretical rule of thumb says that the acceptance rate ideally should be close to 0.234, but between 0.15 and 0.50 is OK[7]. The acceptance rate can be changed by changing the so called tuning parameter, which is the variance parameter in the transition density.

Another way to propose new values in the transition, is by independent proposal. Here the proposed value does not depend on the current value. Convergence of

a chain with independent proposal is either very good or very bad, usually it is very bad. This is because the tails in the proposal distribution should ideally be as heavy as in the target distribution, which is very hard to get[7].

4.1 Acceptance probability

Once again, look at the acceptance probability. As described before, Q is the proposal distribution, which is a density for the continuous case and a probability for the discrete case. π is also a density for the continuous case and a probability for the discrete case. Given a parameter vector, $\boldsymbol{\theta}$, and a data set $\boldsymbol{x}$, Bayes rule gives

$$\begin{aligned}\pi(\boldsymbol{\theta}|\boldsymbol{x}) &= \frac{\pi(\boldsymbol{x}|\boldsymbol{\theta}) \cdot \pi(\boldsymbol{\theta})}{\pi(\boldsymbol{x})} \\ &\propto \pi(\boldsymbol{x}|\boldsymbol{\theta}) \cdot \pi(\boldsymbol{\theta}).\end{aligned}$$

This is inserted into the acceptance probability,

$$\alpha(\boldsymbol{\theta}^*|\boldsymbol{\theta}) = \min\left\{1, \frac{\pi(\boldsymbol{x}|\boldsymbol{\theta}^*) \cdot \pi(\boldsymbol{\theta}^*)}{\pi(\boldsymbol{x}|\boldsymbol{\theta}) \cdot \pi(\boldsymbol{\theta})} \cdot \frac{Q(\boldsymbol{\theta}^*, \boldsymbol{\theta})}{Q(\boldsymbol{\theta}, \boldsymbol{\theta}^*)}\right\}, \tag{4.1}$$

where $\pi(\boldsymbol{\theta})$ is the prior distribution for the parameters, and $\pi(\boldsymbol{x}|\boldsymbol{\theta})$ is the likelihood.

Prior distribution The prior distribution may be chosen freely. Here, one can assume that the parameters comes from a given distribution. All apriori knowledge is put into this prior distribution, for a contribution to the acceptance probability. When doing simulation on a data set, one often have knowledge about the parameters from a different simulation or estimation of the same phenomenon. Say, one is to estimate some parameters in a statistical model given a data set of wind measurements or wave measurements by simulation. Then one can look at some old simulations of other wind data or wave data and use this information for the parameters in the prior distributions.

Likelihood The only place where the data set is taken into account, is through the likelihood. The likelihood is a measure of how well the statistical model fits the given data set.

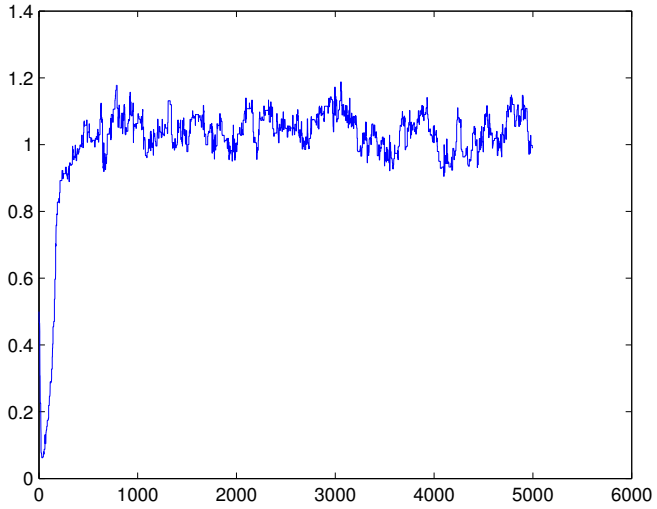


Figure 4.1: Trace plot.

Transition kernel The transition kernel is chosen such that the acceptance rate is at an acceptable level, as discussed above.

4.2 Convergence diagnostics

One essential topic is when the Markov chain has converged, i.e. how many iterations are needed to reach equilibrium. There are several theoretical results, but these are in general too weak to be of any practical value. Usually output analysis is used to determine if the chain has reached its equilibrium distribution. The acceptance rate is itself not enough to determine convergence, but it is a good tool to reach convergence faster, since the variance in the transition density can be controlled. This is called a tuning parameter, since it controls how big or small steps which are proposed. Besides checking the acceptance rate for each parameter, one may look at trace plots to see if the values are stable. Stable means in this context that the value, after an initial or burn-in period, does not change radically. It is a plot of how the parameter evolved during the simulation. An example can be seen in figure 4.1. Here one may observe that after approximately 1000 iterations, the value for the parameter stabilizes around 1. Based on this, one can say that all realisations from the initial value to approximately number 1000 is considered the burn-in period. After this, the chain is unable to remember where it started, and the realisations are essentially samples from the target distribution. In all

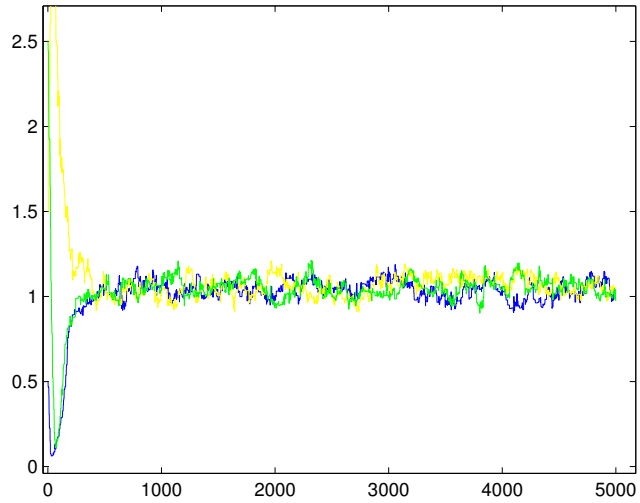
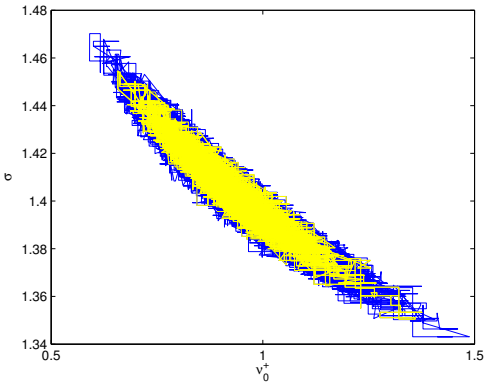


Figure 4.2: Trace plot with three independent runs from different initial values.

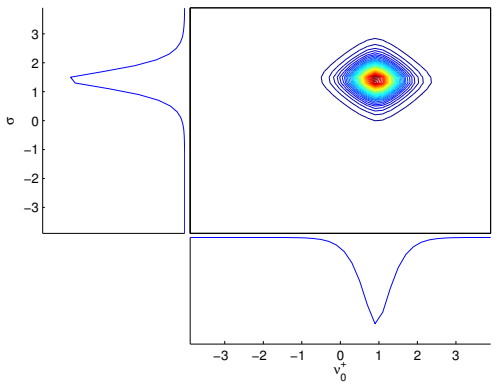
calculations and estimation of the posterior density, the burn-in period must be discarded.

Another tool is to do several runs with different initial states. One may say that convergence is reached if all parameters converge to the same value, from different initial states. This can be seen in figure 4.2. The idea is that the chain is independent of the initial state, so all chains from different initial states shall coincide if the chain converges. This happens in figure 4.2, where all three chains starts to mix before 1000 iterations. If there are several parameters to estimate, one may also plot the parameters against each other, to see how the parameter space and joint distribution behaves. This is shown in figure 4.3, where two parameters are plotted against each other. Here, the parameter in figure 4.1 is plotted against another parameter, whose burn-in period is equal. A total of $800,000$ iterations have been done, and the parameter space does not grow especially from $100,000$ iterations (yellow area) to $800,000$ iterations (blue area). Both exact values are inside the space (the simulations here is from case I in section 7.1.1, so the exact values are known). If the chain has bad mixing properties, one may see that the parameter space will continue to grow uncontrolled when the number of iterations increases, which indicates that the chain has not reached convergence. The parameter space will grow slightly for an increasing number of iterations, since the tail will be visited more often. However, using the mean value of the posterior distribution cannot be done without checking the parameter space. If the parameter space curves, one may risk that the mean will lie outside of the parameter

space, hence the mean does not give any meaning. Looking at the joint posterior density, the posterior density for each parameter is shown. The mode is almost in the exact values, which is a good indication that the parameter estimation is successful.



(a) The parameter space of two parameters. The blue space is after 800,000 iterations while the yellow space is after 100,000 iterations.



(b) The joint density for two parameters.

Figure 4.3: The parameter space (a) and the joint density (b) for two parameters.

4.3 Other algorithms

Metropolis-Hastings is the algorithm used in this thesis, but this is only one of many different algorithms and strategies used in MCMC simulation. There are also several different versions of the Metropolis-Hastings algorithm. In the following, two other common algorithms will be briefly presented.

4.3.1 Metropolis algorithm

A special case of the Metropolis-Hastings is when the proposal distributions are symmetric, i.e. $Q(x, y) = Q(y, x)$. This is the case if the proposal distribution is a normal distribution. As an example, look at the ratio in the following equations

when the proposal density is a normal distribution, centered at the current state.

$$\begin{aligned}\frac{Q(y,x)}{Q(x,y)} &= \frac{\frac{1}{\sqrt{2\pi}}\cdot\frac{1}{\sigma}\cdot\exp\{\frac{1}{2\sigma^2}(x-y)^2\}}{\frac{1}{\sqrt{2\pi}}\cdot\frac{1}{\sigma}\cdot\exp\{\frac{1}{2\sigma^2}(y-x)^2\}} \\ &= 1,\end{aligned} \tag{4.2}$$

so the proposal distributions cancel, and the acceptance probability is independent of the proposal distribution. This is called a Metropolis-proposal, and can save some CPU time, since the acceptance probability is easier to calculate.

4.3.2 The Gibbs sampler

Another sampling algorithm is the Gibbs sampler. The advantage of this algorithm is that one can sample directly from the full conditionals, since they are known. This plays an important role in the acceptance probability, which for Gibbs sampling is

$$\begin{aligned}\alpha(x,y) &= \min\left\{1, \frac{\pi(y)}{\pi(x)}\cdot\frac{Q(y,x)}{Q(x,y)}\right\} \\ &= \min\left\{1, \frac{\pi(y)}{\pi(x)}\cdot\frac{\frac{\pi(x^i,y^{-i})}{\pi(y^{-i})}}{\frac{\pi(y^i,x^{-i})}{\pi(x^{-i})}}\right\} \\ &= \min\left\{1, \frac{\pi(y)}{\pi(x)}\cdot\frac{\pi(x^i,x^{-i})}{\pi(y^i,y^{-i})}\right\} \\ &= 1.\end{aligned}$$

Since the acceptance probability is always 1, this means that every proposal is accepted. There are both pros and cons with this method, but the requirement that one needs the full conditionals reduces the ability to use Gibbs sampling, since one very often is unable to find the full conditionals. Full conditionals and conjugate distributions are described more in appendix A. $\pi(y^i, x^{-i})$ is easy to sample from if one have used conditional conjugate prior distributions. Another advantage is that there is no tuning parameter, since the acceptance probability is always equal to 1.

Chapter 5

The Time Series

In this thesis, all analysis on time series have been done on synthetic Gaussian time series. The advantage of synthetic time series is that one know some theoretical results from the generation process, which can be used later when the different MCMC algorithms tries to adapt a model to the data.

5.1 Generation of time series

The time series used are generated as Gaussian time series, from a given spectral density. All spectral densities used, denoted $S(\omega)$, are so-called "box-spectrum's", because of their shape. The spectral density is often just referred to as spectrum. All spectral densities used are multiplied with a constant, to control the variance. The variance is given as

$$\sigma^2 = \int_0^\infty S(\omega)\, \mathrm{d}\omega. \tag{5.1}$$

First, introduce the characteristic function of a set E, defined by

$$\chi_E(x) = \begin{cases} 1 & \text{if } x \in E, \\ 0 & \text{if } x \notin E. \end{cases}$$

As an example, one spectral density is given by the function

$$S(\omega) = 4 \cdot \chi_{[2,6]}(\omega).$$

A graph of this function is found in figure 5.1. As one may see, the shape of the spectrum is like a box. Once the spectrum is in place, the time series are

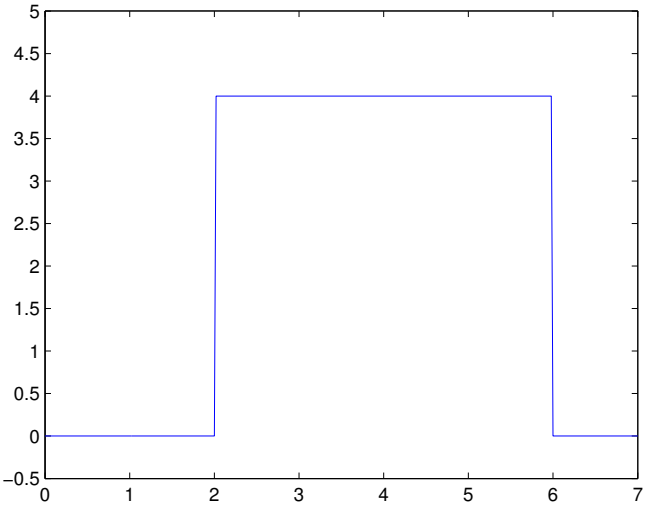


Figure 5.1: $S(\omega) = 4 \cdot \chi_{[2,6]}(\omega)$.

generated from the spectrum by the following formulas, given in [8].

$$X(t) = \sum_{j=1}^{n} \sqrt{S(\omega_j) \cdot \Delta\omega_j} \cdot \Big(a_i \cos(\omega_i t) + b_i \sin(\omega_j t) \Big), \tag{5.2}$$

where a_i and b_i are independent zero-mean Gaussians with variance 1. This expression can also be written as

$$X(t) = \mathcal{R} \left\{ \sum_{j=1}^{n} \sqrt{S(\omega_j) \cdot \Delta\omega_j} \cdot (a_j + i b_j) \exp\{i\omega_j t\} \right\}, \tag{5.3}$$

where $\mathcal{R}$ denotes the real part of the expression, and i is the complex number defined by $i = \sqrt{-1}$. This last expression is used when generating the time series, since one then can apply the discrete Fourier transform, to prevent the series from repeating itself after a long period of time. The effective period of (5.2) is $\frac{2\pi}{\Delta t}$. With the discrete Fourier transform, one have better control over the period, since $\Delta\omega$ can be made smaller without an increasing computing time. The sine and cosine are very computer intensive to find compared to the discrete Fourier transform, so with Fourier transform one can make very long time series and still avoid periodicity in the series.

When generating the time series, one must specify how dense the time series should be sampled. It must contain enough points to catch the peaks of the series. This is done by the factor Δt in the following way. The factor Δt decides how dense

the sampling is done, i.e. how many observations there are per time unit (which is the definition of frequency). The maximal frequency of ω is then

$$\omega_{\max} = \frac{2\pi}{\Delta t}.$$

Then the discretization of ω between 0 and $\omega_{\max}$ will be such that each point is equidistant. Call this discretized vector $\boldsymbol{\omega}$, where ω_j is the j'th element in $\boldsymbol{\omega}$ where $j = 1, ..., n+1$ and n is the length of the series, such that

$$0 < \omega_1 < ... < \omega_{n-1} < \omega_n = \omega_{\max}.$$

$\Delta\omega$ is then the difference between two neighbors in $\boldsymbol{\omega}$, which is constant for all j. Then, using formula (5.3) and the *fft* function in Matlab, one has generated a Gaussian time series with a given density of points. The following algorithm is used to generate the time series in Matlab.

Algorithm *Time series*
($*$ Generates a time series of length n $*$)
1. chose Δt
2. $\omega_{\max} = \frac{2\pi}{\Delta t}$
3. discretisize $\omega_{\max}$ into $n+1$ equidistant points
4. chose $S(\omega)$
5. **for** $j = 1, ..., n$
6. draw a_j, $b_j \sim \mathrm{N}(0, 1)$
7. $f = \sqrt{S(\omega_j) \cdot \Delta\omega_j} \cdot (a_j + ib_j)$
8. **end**
9. $x =$ **real**(**fft**(f))
10. **return** x

The resulting vector $\boldsymbol{x}$ is then a Gaussian time series of length n with mean equal to zero, and variance determined by the formula in (5.1). This algorithm is used m times to generate m independent time series.

5.2 Theoretical properties

From [8], the theoretical mean zero-upcrossing rate is defined as

$$\nu_X^+(0) = \frac{\sigma_{\dot{X}}}{2\pi\sigma_X} = \frac{1}{2\pi}\sqrt{\frac{m_2}{m_0}}, \tag{5.4}$$

where m_j is the spectral moments given by

$$m_j = \int_0^\infty \omega^j S(\omega) \mathrm{d}\omega, \qquad j = 0, 1, ... \tag{5.5}$$

The mean upcrossing rate is defined as how many observations that are above 0, given that the previous observation is below 0. With the algorithm above, it is easy to generate time series, where the mean upcrossing rate is known. This is done to test the stochastic simulation scheme. The idea is to generate data, and try to estimate the parameters in the extreme value distribution of the generated time series. Since the parameters are already known, this becomes a test to check the consistency in the MCMC scheme.

5.3 Extreme value distribution

From the algorithm in the previous section, time series are generated from different spectrums. This is done with the aim of testing the robustness in estimating the parameters in the statistical model with MCMC. The extremes in the series has the following approximate distribution,

$$\eta_{\text{extreme}} \sim \exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta - b)^2}{2\sigma^2}\}\right\}, \tag{5.6}$$

which is asymptotic Gumbel. ν_0^+ is the mean upcrossing rate, T is the length of the time series in seconds, η is the observation, b is the mean of the time series while σ^2 is the variance. All these parameters are known from the generation process, so to test the Metropolis-Hastings algorithm, one will first pretend that ν_0^+ and σ^2 is unknown, b is equal to zero and T is the length of each series. This will give a MCMC simulation with two parameters that have to be decided by simulating the posterior distribution for each parameter. Then one will assume that b is unknown, so one have to estimate 3 parameters. Finally, the $(\eta - b)^2$ term in (5.6) is replaced by $(\eta - b)^c$, so 4 parameters have to be estimated. The parameter c is a shape parameter. In extreme value analysis with a different method called Average Conditional Exceedance Rate (ACER-method [9]), one have discovered problems in some cases with estimation of this constant c. If the algorithm in this thesis is successful, this can be a great contribution to the ACER method.

5.4 The data

Three different spectral densities are used,

$$\begin{aligned} \text{I} \quad & S(\omega)_{\text{I}} && = \tfrac{1}{4} \cdot \chi_{[2,10]}, \\ \text{II} \quad & S(\omega)_{\text{II}} && = 4 \cdot \chi_{[2,6]}, \\ \text{III} \quad & S(\omega)_{\text{III}} && = 3 \cdot \chi_{[3,6]}. \end{aligned} \tag{5.7}$$

The constant in front of the characteristic function is to control the variance. Using the formulas in equation (5.4) and (5.5), one can easily see that the mean upcrossing rate is independent of this constant. The first spectrum is very wide, while the next one and third becomes more narrow banded. The narrower the spectrum is, the time series consists of less sine and cosine components with high energy. The energy is dependent of the area of the spectrums. The bigger area, the more energy. A practical interpretation of this can be wave heights. Say, at $t = 0$ the wind speed increase to a given constant speed. It will then take some time before the waves will be affected by this. First waves with low amplitude are formed, with a small amount of energy and short frequency. This corresponds to a spectrum with small area. When the time evolves, the amplitude will grow, i.e. the waves will be higher which means more energy and bigger spectrum area [10].

From equation (5.4) and (5.5), the exact mean upcrossing rate and variance can be estimated for all three cases. For the first case, the spectral moments are

$$\begin{aligned} m_0 = \int_0^\infty S(\omega)\mathrm{d}\omega &= \frac{1}{4}\int_0^\infty \chi_{[2,10]}\mathrm{d}\omega \\ &= \frac{1}{4}\int_2^{10} \mathrm{d}\omega \\ &= \frac{1}{4} \cdot (10-2) = 2 \end{aligned}$$

and

$$\begin{aligned} m_2 = \int_0^\infty \omega^2 S(\omega)\mathrm{d}\omega &= \frac{1}{4}\int_2^{10} \omega^2 \mathrm{d}\omega \\ &= \frac{1}{4} \cdot \frac{1}{3}(10^3 - 2^3) \\ &= \frac{1}{12} \cdot 992. \end{aligned}$$

This is inserted into equation (5.4),

$$\nu_0^+ = \frac{1}{2\pi}\sqrt{\frac{m_2}{m_0}} = \frac{1}{2\pi}\sqrt{\frac{1}{12}\cdot 992\cdot\frac{1}{2}}$$
$$\approx 1.02.$$

From these calculations, one obtains the true values for the standard deviation and the mean upcrossing rate for the time series generated by the spectral density $S(\omega)_{\mathrm{I}}$ in (5.7). The corresponding values for the two other cases are found in the same way, and all values are shown in table 5.1 below.

Table 5.1: Analytic results for the standard deviation and mean upcrossing rate from the densities in equation (5.7).

Spectrum	Standard deviation	Mean upcrossing rate
$S(\omega)_{\mathrm{I}}$	$\sqrt{2}$	1.02
$S(\omega)_{\mathrm{II}}$	4	0.66
$S(\omega)_{\mathrm{III}}$	3	0.73

Chapter 6

Implementation

6.1 Building up the MCMC algorithm

For the first simulation where only two parameters are to be estimated, three different schemes are tested, which are all based on the Metropolis-Hastings algorithm. In the first two schemes, it is assumed that the parameters have a normal distribution as prior distribution, with a restriction for negative values for both parameters, i.e. all negative proposals are rejected. This is because negative values for these parameters does not give any meaning. Looking at the distribution in equation (5.6), it is obvious that the mean upcrossing rate and the standard deviation cannot be negative. The normal distribution has the following density.

$$f(x) = \frac{1}{\sqrt{2\pi}\sigma} \exp\{-\tfrac{1}{2\sigma^2}(x-\mu)^2\},$$

where μ is the mean and σ^2 is the variance. The two first algorithms updates first one parameter, then the next and so on, such that each parameter is independent of each other in the sense that a new proposed parameter does not influence whether another parameter is accepted or rejected. This means that the parameters are apriori independent. The third algorithm updates both parameters at the same time. In the third algorithm, the mean upcrossing rate is assumed to be log-normally distributed, while the standard deviation still is assumed to be normally distributed. To be log-normally distributed means that the natural logarithm of the random variable is normally distributed. The log-normal distribution has support from $[0, \infty]$, hence it cannot propose negative values. The density for the

log-normal distribution is

$$f(x) = \frac{1}{x\sqrt{2\pi}\sigma} \exp\{-\tfrac{1}{2\sigma^2}(\ln x - \mu)^2\}, \quad \wedge \quad \begin{cases} \mu & = \ln\left(\mathrm{E}[x]\right) - \frac{1}{2}\ln\left(1 + \frac{\mathrm{Var}[x]}{\mathrm{E}[x^2]}\right) \\ \sigma^2 & = \ln\left(1 + \frac{\mathrm{Var}[x]}{\mathrm{E}[x^2]}\right) \end{cases}, \tag{6.1}$$

where μ and σ^2 are the mean and variance of the variable's natural logarithm. At first, one will simulate the posterior density for the unknown parameters. The hypothesis is that the mean of the posterior density for each parameter, will be approximately equal to the theoretical value found in table 5.1. There are several other possibilities for the prior distributions, but this is one of the issues about MCMC, there is no answer to what the "correct" prior distribution should be. Recall the acceptance probability in equation (4.1). In the following, each of the three terms is described in detail.

Prior information Both parameters have been given a prior distribution. This is common, since one usually knows something about the parameters. In extreme value statistics, the prior can be decided due to earlier or similar parameter estimation, but only on data different from the data which is to be used in the simulation. All prior knowledge shall be included in the prior information, since this can simplify or "help" the algorithm to find the "correct" posterior distribution. At first it is assumed that the desired prior distribution have a mean equal to the true value, with standard deviation 0.5. Later, a predetermined value unequal to the true value have been set in the prior distribution, to see how the posterior distribution depends on the prior choice.

The Likelihood The likelihood term in the acceptance probability is built up the following way. The analysis is done on one single, long time series consisting of $2,000,000$ points. The whole series is divided into n batches. In each batch, only the maximum observation is used. The length of the batch will be constant through the simulation, but in the different analysis, the length will vary between the different simulations. The maximum value, i.e. the extremes, is assumed to be independent. The maximum in each block is put into the density function in equation (5.6), and multiplied together. This is the only term where the data will be considered. This can be expressed in the following way. The CDF (cumulative

distribution function) for an event being larger than a given threshold η, is

$$F(\eta) = \exp\left\{-\nu_0^+ T \exp\{-\frac{\eta^2}{2\sigma^2}\}\right\}. \tag{6.2}$$

To obtain the PDF (probability density function), one simply takes the derivative of $F(\eta)$ with respect to η.

$$\begin{aligned} f(\eta) = \frac{\partial F}{\partial \eta} &= \exp\left\{-\nu_0^+ T \exp\{-\frac{\eta^2}{2\sigma^2}\}\right\} \cdot -\nu_0^+ T \exp\{-\frac{\eta^2}{2\sigma^2}\} \cdot -\frac{\eta}{\sigma^2} \\ &= \exp\left\{-\nu_0^+ T \exp\{-\frac{\eta^2}{2\sigma^2}\}\right\} \cdot \nu_0^+ T \exp\{-\frac{\eta^2}{2\sigma^2}\} \cdot \frac{\eta}{\sigma^2}. \end{aligned}$$

The likelihood is expressed as the ratio between the proposed new parameter vector, and the old one,

$$\begin{aligned} \frac{f(\boldsymbol{\eta}|\tilde{\boldsymbol{\theta}})}{f(\boldsymbol{\eta}|\boldsymbol{\theta})} &= \prod_{i=1}^{r} \left(\frac{f(\eta_i|\tilde{\boldsymbol{\theta}})}{f(\eta_i|\boldsymbol{\theta})} \right) \\ &= \prod_{i=1}^{r} \left(\frac{\exp\left\{-\tilde{\nu}_0^+ T \exp\{-\frac{\eta_i^2}{2\tilde{\sigma}^2}\}\right\} \cdot \tilde{\nu}_0^+ T \exp\{-\frac{\eta_i^2}{2\tilde{\sigma}^2}\} \cdot \frac{\eta_i}{\tilde{\sigma}^2}}{\exp\left\{-\nu_0^+ T \exp\{-\frac{\eta_i^2}{2\sigma^2}\}\right\} \cdot \nu_0^+ T \exp\{-\frac{\eta_i^2}{2\sigma^2}\} \cdot \frac{\eta_i}{\sigma^2}} \right), \end{aligned}$$

where the tilde indicates the proposed new value. On log-level this yields

$$\begin{aligned} \ln\left(\frac{f(\boldsymbol{\eta}|\tilde{\boldsymbol{\theta}})}{f(\boldsymbol{\eta}|\boldsymbol{\theta})}\right) &= \sum_{i=1}^{r} \ln\left(\frac{f(\eta_i|\tilde{\boldsymbol{\theta}})}{f(\eta_i|\boldsymbol{\theta})}\right) \\ &= \sum_{i=1}^{r} \Big[\ln\left(\frac{\tilde{\nu}_0^+}{\nu_0^+}\right) - \eta_i^2 (\tfrac{1}{2\tilde{\sigma}^2} - \tfrac{1}{2\sigma^2}) + \ln\left(\frac{\sigma^2}{\tilde{\sigma}^2}\right) \\ &\quad - T\left(\tilde{\nu}_0^+ \exp\{-\tfrac{\eta_i^2}{2\tilde{\sigma}^2}\} - \nu_0^+ \exp\{-\tfrac{\eta_i^2}{2\sigma^2}\}\right) \Big] \end{aligned}$$

where η_i is the maximum observation in block i, r is the total number of blocks and $\boldsymbol{\theta}$ is the parameter vector, which in this case is

$$\boldsymbol{\theta} = \begin{bmatrix} \nu_0^+ \\ \sigma \end{bmatrix}.$$

The transition kernel As transition kernel, the proposed parameter is drawn from the desired distribution, with the current state as the mean value, with a tuning parameter as the variance so the acceptance rate is at an acceptable

level. All transition probabilities concerning the normal distribution will cancel out(see equation (4.2)). For the log-normal distribution, the transition probability becomes

$$\frac{Q(y,x)}{Q(x,y)} = \frac{\frac{1}{x\sqrt{2\pi}}\frac{1}{\sigma_x}\exp\{-\frac{1}{2\sigma_x^2}(\ln x - \mu_x)^2\}}{\frac{1}{y\sqrt{2\pi}}\frac{1}{\sigma_y}\exp\{-\frac{1}{2\sigma_y^2}(\ln y - \mu_y)^2\}}, \tag{6.3}$$

where μ and σ^2 is given as in (6.1).

6.2 Single update versus multiple update

There are both advantages and disadvantages with single parameter update versus a multiple update.

Single update Here, each parameter is updated separately, so there will be an acceptance probability for each parameter that has to be calculated for each proposal. The advantage is that the acceptance rate can be higher than multiple update, since each update is independent of the other parameters.

Multiple update Now a parameter vector is proposed, with values for all parameters at the same time. This can save CPU time, since the acceptance probability only have to be calculated once, but the acceptance rate is often lower than for a single update. This is because if all parameters are good proposals except one, this one proposal may lead to the whole vector being rejected. Hence, no parameter is updated, even though many proposals would be good enough considering each parameter separately.

When simulating the three cases, three different schemes were used, both single and multiple update for both parameters distributed as Gaussians, and a single update when the mean upcrossing rate was assumed to be from the log-normal distribution. After some runs, the multiple updated scheme was discarded due to the low acceptance rate. Even though the tuning parameter (variance in the transition density) was set very low, the algorithm used a very long time to explore the parameter space.

6.3 Implementation in Matlab

The implementation of the algorithm is done in Matlab. All calculation of the acceptance probability is done on log-level, otherwise it can collapse in many situations. Matlab is not very comfortable with for example exponential of large numbers, which can occur in the acceptance probability. The acceptance probability now becomes

$$\ln\left(\alpha(x,y)\right) = \min\left\{0, \ln\left(\frac{\pi(\boldsymbol{x}|\tilde{\boldsymbol{\theta}})}{\pi(\boldsymbol{x}|\boldsymbol{\theta})}\right) + \ln\left(\frac{\pi(\tilde{\boldsymbol{\theta}})}{\pi(\boldsymbol{\theta})}\right) + \ln\left(\frac{Q(\tilde{\boldsymbol{\theta}},\boldsymbol{\theta})}{Q(\boldsymbol{\theta},\tilde{\boldsymbol{\theta}})}\right)\right\}.$$

As an example, the transition kernel in equation (6.3) becomes

$$\begin{aligned}\ln\left(\frac{Q(y,x)}{Q(x,y)}\right) &= \ln\left(\frac{y}{x}\right) + (-\tfrac{1}{2\sigma_x^2}(\ln x - \mu_x)^2) - (-\tfrac{1}{2\sigma_y^2}(\ln y - \mu_y^2)^2) \\ &= \ln\left(\frac{y}{x}\right) + \tfrac{1}{2\sigma_y^2}(\ln y - \mu_y)^2 - \tfrac{1}{2\sigma_x^2}(\ln x - \mu_x)^2 + \ln\left(\frac{\sigma_y}{\sigma_x}\right),\end{aligned}$$

where

$$\begin{aligned}\mu_y &= \ln\left(\mathrm{E}[y]\right) - \tfrac{1}{2}\ln\left(\frac{\mathrm{Var}[y]}{\mathrm{E}[y]^2}\right), \\ \sigma_y^2 &= \ln\left(1 + \tfrac{\mathrm{Var}[y]}{\mathrm{E}[y]^2}\right),\end{aligned}$$

and corresponding for μ_x and σ_x^2. For this transition probability, the expected value of y, $\mathrm{E}[y] = x$, which is the current value and vice versa for $\mathrm{E}[x]$, which has y as expected value, hence

$$\ln\left(\frac{Q(y,x)}{Q(x,y)}\right) = \ln\left(\frac{y}{x}\right) + \ln\left(\tfrac{\sigma_y}{\sigma_x}\right) + \tfrac{1}{2\sigma_y^2}[\ln\{\tfrac{y}{x}\} + \tfrac{1}{2}\sigma_y^2] - \tfrac{1}{2\sigma_x^2}[\ln\{\tfrac{x}{y}\} + \tfrac{1}{2}\sigma_x^2].$$

Chapter 7

Results

Now, all theory and apriori knowledge have been presented and the implementation briefly explained. The first simulation is with 2 unknown parameters.

7.1 First simulation

With all assumptions made, the distribution in equation (5.6) now simplifies to

$$\eta_{\text{extremes}} \sim \exp\left\{-\nu_0^+ T \exp\{-\frac{\eta^2}{2\sigma^2}\}\right\}, \tag{7.1}$$

where T is the length of each block in seconds, and ν_0^+ and σ remain as the unknown parameters to be simulated with MCMC.

7.1.1 Case I

Now, $S(\omega)_{\text{I}}$ from equation (5.7) is used to generate the time series. This is a very wide spectrum, which means that the time series consist of many different components with different energy. This will give a very irregular time series, in the sense that there is no indication of dependence between each peak. In figure 7.1 one can see the first 2000 points of the generated time series. In table 7.1 one can see the result of some simulations with different block lengths in the likelihood, for the given spectral density. In all simulations, the prior information is given as

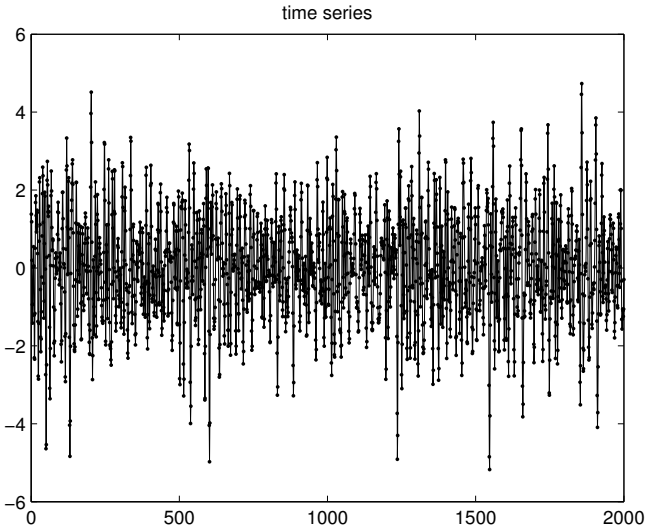


Figure 7.1: First 2000 points of a time series generated by $S(\omega)_{\mathrm{I}} = \frac{1}{4} \cdot \chi_{[2,10]}(\omega)$.

follows

$$\begin{aligned} \nu_0^+ &\sim \begin{cases} \mathcal{N}(\mu_{\nu_0^+}, 0.5^2) \\ \text{Log-}\mathcal{N}(\mu_{\nu_0^+}, 0.5^2) \end{cases} \\ \sigma &\sim \mathcal{N}(\mu_\sigma, 0.5^2), \end{aligned} \tag{7.2}$$

where μ_* is the true value for the given parameter. All values in the table is the mean of the posterior density, with an appropriate burn-in period discarded. The total number of iterations are $100,000$. The plots in figure 4.1 and 4.2 are

Table 7.1: Estimation of the parameters from three different MCMC schemes, with different block sizes. The data is generated by $S(\omega)_{\mathrm{I}}$ and the acceptance rate is shown in parenthesis.

Block length	Normal update for ν_0^+	Log-normal update for ν_0^+	Multi update
400	$\nu_0^+ \approx 1.02(19\%)$	$\nu_0^+ \approx 1.02(18\%)$	$\nu_0^+ \approx 1.03(6\%)$
	$\sigma \approx 1.39(12\%)$	$\sigma \approx 1.39(12\%)$	$\sigma \approx 1.38(6\%)$
600	$\nu_0^+ \approx 1.01(20\%)$	$\nu_0^+ \approx 1.01(20\%)$	$\nu_0^+ \approx 0.91(7\%)$
	$\sigma \approx 1.40(14\%)$	$\sigma \approx 1.40(12\%)$	$\sigma \approx 1.39(7\%)$
2000	$\nu_0^+ \approx 0.93(33\%)$	$\nu_0^+ \approx 0.87(34\%)$	$\nu_0^+ \approx 0.93(7\%)$
	$\sigma \approx 1.40(18\%)$	$\sigma \approx 1.41(19\%)$	$\sigma \approx 1.37(7\%)$

the trace plots from the simulation where ν_0^+ was log-normally distributed with a

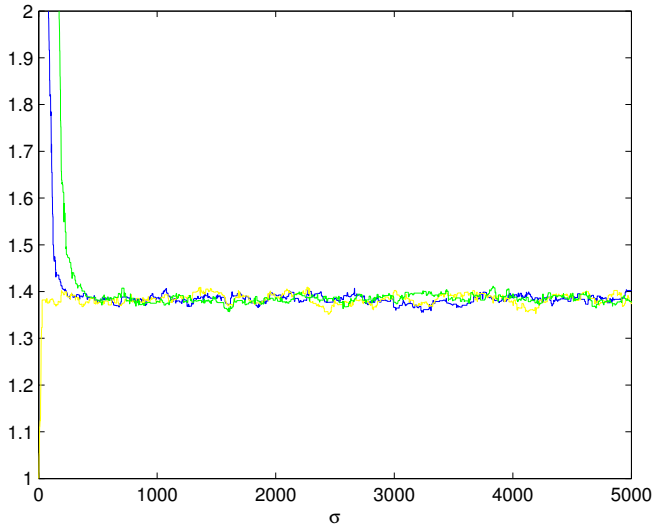
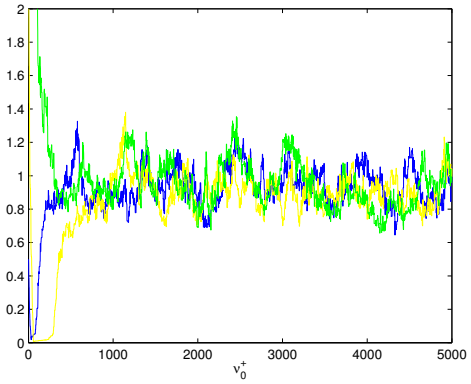
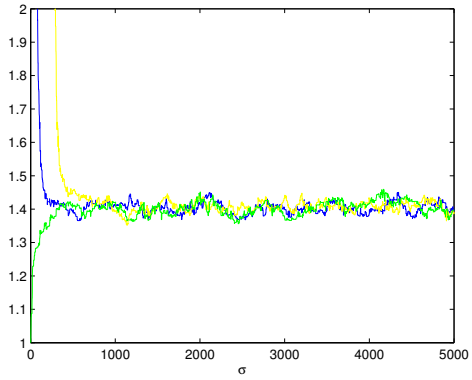


Figure 7.2: Trace plot of the standard deviation with 3 independent runs starting from different initial states from a simulation with block length 600. Blue: $\sigma_0 = 3$, Yellow: $\sigma_0 = 1$, Green: $\sigma_0 = 4$. The data is generated by $S(\omega)_{\mathrm{I}}$.

block length of 600 points. In figure 7.2 the trace plot of the standard deviation is shown for the same case, with three independent runs from different initial states. For both parameters, it looks like the chain have reached convergence after at least 1000 iterations, so this is considered as the burn-in period, which is discarded in all estimation of the posterior distributions. For the simulation with block length equal to 2000, the trace plots for both parameters are shown in figure 7.3, and their parameter space and joint posterior density in figure 7.4. To be able to estimate the parameters by taking the mean of the posterior distribution, it is useful to check the parameter space, and how it evolves when the number of iterations increases. From the parameter space in 7.4a, one may observe that the true values lie within the density, which is an indication of a well functioning simulation algorithm for this case, hence for this given spectral density, it looks like the MCMC output is close enough to the true values. Observe that the posterior density for each parameter is plotted to the left and under the posterior joint density in 7.4b, and that the mode is very near the true value for the given spectrum. Comparing the results for the different block lengths, one observes that the larger block lengths, the more the estimation differs from the true values.

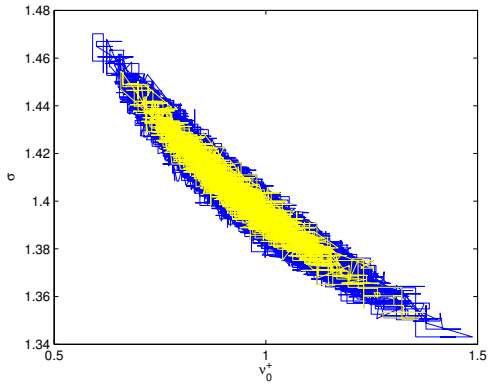


(a) Traceplot of the mean upcrossing rate with 3 independent runs with different initial states. Blue: $\nu_0^+ = 0.5$, Yellow: $\nu_0^+ = 2.5$, Green: $\nu_0^+ = 3$.

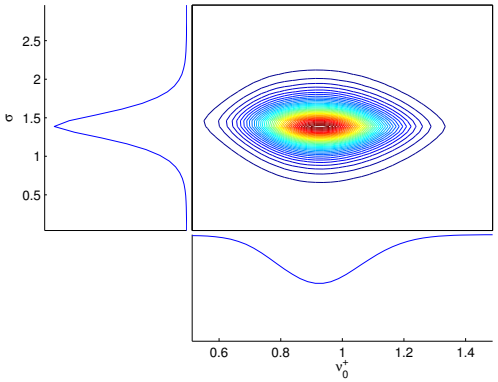


(b) Traceplot of the standard deviation with 3 independent runs with different initial states. Blue: $\sigma_0 = 3$, Yellow: $\sigma_0 = 5$, Green: $\sigma_0 = 1$.

Figure 7.3: Trace plots of the mean upcrossing rate (a) and the standard deviation (b) with 3 independent runs starting from different initial states from a simulation with a block length of 2000. The data is generated by $S(\omega)_{\mathrm{I}}$.



(a) The parameter space for ν_0^+ and σ. The blue space is after $1,000,000$ iterations, while the yellow space is after $100,000$ iterations.



(b) The joint posterior density between ν_0^+ and σ.

Figure 7.4: The parameter space (a) and the joint posterior density (b) for a simulation with block length 2000 from data generated by $S(\omega)_{\mathrm{I}}$.

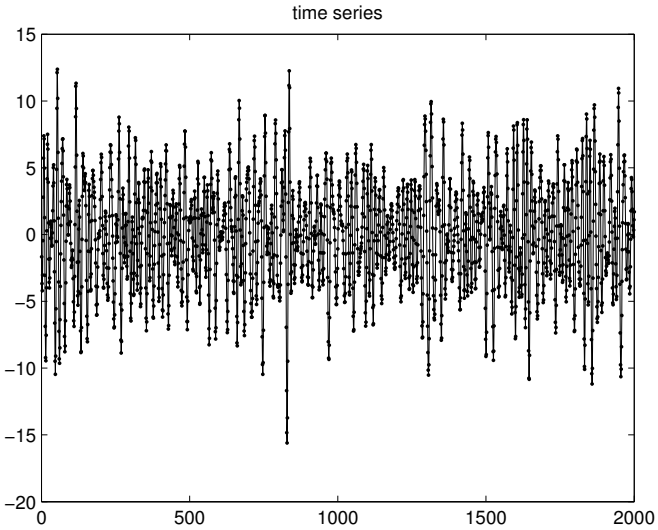


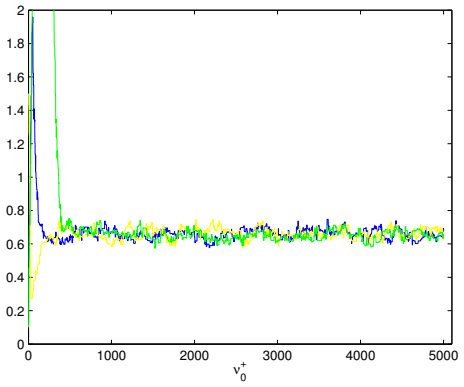
Figure 7.5: First 2000 points of a time series generated by $S(\omega)_{\text{II}} = 4 \cdot \chi_{[2,6]}(\omega)$.

7.1.2 Case II

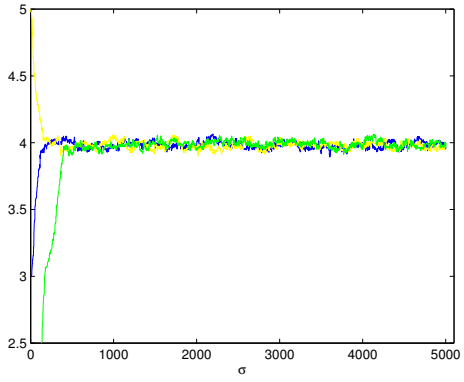
In this case the spectrum $S(\omega)_{\text{II}}$ in (5.7) is used. This spectrum can be seen in figure 5.1. In figure 7.5, the first 2000 points are shown. The spectrum is not as wide as in case I. In table 7.2 the simulation output is shown for the same block lengths as case I, with the prior distributions as in expression (7.2). The total number of iterations performed is $100,000$. The acceptance rate is given in parentheses. Comparing this to the true values, the mean of the simulated posterior distributions for both the mean upcrossing rate and the standard deviation approximately coincide for block lengths of 400 and 600. In figure 7.6 the trace plot with block length 600 and 3 independent runs from different initial values are shown for both parameters. Visual inspection suggests a burn-in period of approximately 600 iterations, which is used in the estimation in table 7.2. For a block length of 2000, the estimates are not quite as good as for block lengths of 400 and 600. Especially the multi update algorithm has a large deviation, together with a low acceptance rate. This may indicate that the algorithm uses a long time to explore the parameter space, or that the algorithm is unable to find the true values at all. There is no guarantee that MCMC gives the "correct" result, there exist many situations where MCMC methods will go wrong.

Table 7.2: Estimation of the parameters from three different MCMC schemes, with different block sizes. The data is generated by $S(\omega)_{\text{II}}$ and the acceptance rate is shown in parenthesis.

Block length	Normal update for ν_0^+	Log-normal update for ν_0^+	Multi update
400	$\nu_0^+ \approx 0.66(14\%)$	$\nu_0^+ \approx 0.67(15\%)$	$\nu_0^+ \approx 0.59(6\%)$
	$\sigma \approx 3.97(36\%)$	$\sigma \approx 3.96(35\%)$	$\sigma \approx 3.98(6\%)$
600	$\nu_0^+ \approx 0.66(15\%)$	$\nu_0^+ \approx 0.65(18\%)$	$\nu_0^+ \approx 0.60(7\%)$
	$\sigma \approx 3.97(37\%)$	$\sigma \approx 3.98(36\%)$	$\sigma \approx 3.97(7\%)$
2000	$\nu_0^+ \approx 0.61(23\%)$	$\nu_0^+ \approx 0.58(28\%)$	$\nu_0^+ \approx 0.55(6\%)$
	$\sigma \approx 4.02(47\%)$	$\sigma \approx 4.04(48\%)$	$\sigma \approx 3.89(6\%)$



(a) Trace plot of the mean upcrossing rate with 3 independent runs with different initial states. Blue: $\nu_{0\,0}^+ = 0.5$, Yellow: $\nu_{0\,0}^+ = 1.5$, Green: $\nu_{0\,0}^+ = 0.1$



(b) Trace plot of the standard deviation with 3 independent runs with different initial states. Blue: $\sigma_0 = 3$, Yellow: $\sigma_0 = 5$, Green: $\sigma_0 = 1$.

Figure 7.6: Trace plots of the mean upcrossing rate (a) and standard deviation (b) with 3 independent runs starting from different initial states from a simulation with a block length of 600. The data is generated by $S(\omega)_{\text{II}}$.

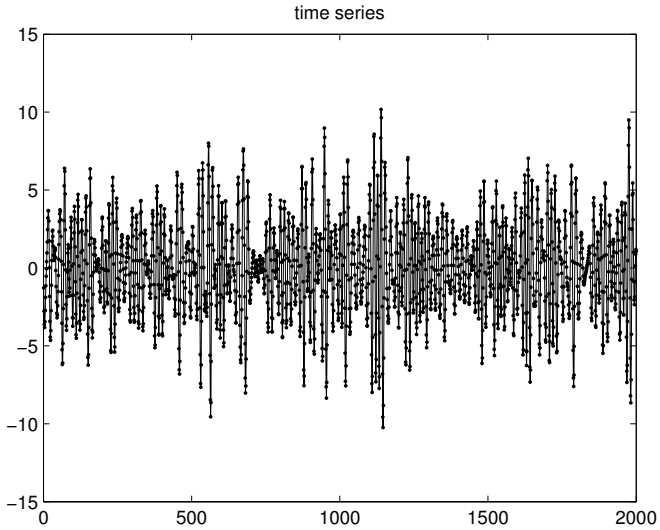


Figure 7.7: First 2000 points of a time series generated by $S(\omega)_{\text{III}} = 3 \cdot \chi_{[3,6]}(\omega)$.

7.1.3 Case III

Finally spectrum III in (5.7) is used to generate the time series. Figure 7.7 shows the same plot as in the previous cases. In table 7.3 the same estimates as in the

Table 7.3: Estimation of the parameters from three different MCMC schemes, with different block sizes. The data is generated by $S(\omega)_{\text{III}}$ and the acceptance rate is shown in parenthesis.

Block length	Normal update for ν_0^+	Log-normal update for ν_0^+	Multi update
400	$\nu_0^+ \approx 0.71(14\%)$	$\nu_0^+ \approx 0.71(15\%)$	$\nu_0^+ \approx 0.69(6\%)$
	$\sigma \approx 2.96(36\%)$	$\sigma \approx 2.97(35\%)$	$\sigma \approx 2.88(6\%)$
600	$\nu_0^+ \approx 0.70(15\%)$	$\nu_0^+ \approx 0.70(18\%)$	$\nu_0^+ \approx 0.66(7\%)$
	$\sigma \approx 2.97(37\%)$	$\sigma \approx 2.98(36\%)$	$\sigma \approx 2.97(7\%)$
2000	$\nu_0^+ \approx 0.61(23\%)$	$\nu_0^+ \approx 0.68(28\%)$	$\nu_0^+ \approx 0.59(6\%)$
	$\sigma \approx 3.04(47\%)$	$\sigma \approx 3.02(48\%)$	$\sigma \approx 2.89(6\%)$

previous sections are shown. The results here does not coincide with the true values at all. It looks like the more narrow banded the spectrum is, the more difficult it is to obtain good estimates from the posterior distribution for each parameter. Due to the low acceptance rate and bad estimates in the multi update algorithm, this algorithm is discarded in the further analysis.

7.2 Expanding the model

Now, one will expand the model in (7.1) by including the parameter b. This is now the same as (5.6). This parameter is an expression for the mean value of the data points in the time series. When generating the time series, one already know that the value of b is 0. The CDF is now given as

$$F(\eta) = \exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta - b)^2}{2\sigma^2}\}\right\}.$$

The PDF is found in the same way as above, by taking the derivative of $F(\eta)$, which yields

$$f(\eta) = \exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta - b)^2}{2\sigma^2}\}\right\} \cdot \nu_0^+ T \exp\{-\frac{(\eta - b)^2}{2\sigma^2}\} \cdot \frac{(\eta - b)}{\sigma^2}.$$

The new likelihood model then becomes

$$\begin{aligned} f(\boldsymbol{\eta}|\boldsymbol{\theta}) &= \sum_{i=1}^{r} f(\eta_i|\boldsymbol{\theta}) \\ &= \prod_{i=1}^{r} \left(\exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta_i - b)^2}{2\sigma^2}\}\right\} \cdot \nu_0^+ T \exp\{-\frac{(\eta - b)^2}{2\sigma^2}\} \cdot \frac{(\eta - b)}{\sigma^2} \right), \end{aligned}$$

where η_i is the maximum observation in block i, and r is the total number of blocks. This gives the following ratio

$$\begin{aligned} \frac{f(\boldsymbol{\eta}|\tilde{\boldsymbol{\theta}})}{f(\boldsymbol{\eta}|\boldsymbol{\theta})} &= \prod_{i=1}^{r} \left(\frac{f(\eta_i|\tilde{\boldsymbol{\theta}})}{f(\eta_i|\boldsymbol{\theta})} \right) \\ &= \prod_{i=1}^{r} \left(\frac{\exp\left\{-\tilde{\nu}_0^+ T \exp\{-\frac{(\eta_i - \tilde{b})^2}{2\tilde{\sigma}^2}\}\right\}}{\exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta_i - b)^2}{2\sigma^2}\}\right\}} \cdot \frac{\tilde{\nu}_0^+ \exp\{-\frac{(\eta_i - \tilde{b})^2}{2\tilde{\sigma}^2}\} \cdot \frac{(\eta_i - \tilde{b})}{\tilde{\sigma}^2}}{\nu_0^+ \exp\{-\frac{(\eta_i - b)^2}{2\sigma^2}\} \cdot \frac{(\eta_i - b)}{\sigma^2}} \right), \end{aligned}$$

which on log-level simplifies to

$$\begin{aligned} \ln\left(\frac{f(\boldsymbol{\eta}|\tilde{\boldsymbol{\theta}})}{f(\boldsymbol{\eta}|\boldsymbol{\theta})}\right) &= \sum_{i=1}^{r} \ln\left(\frac{f(\eta_i|\tilde{\boldsymbol{\theta}})}{f(\eta_i|\boldsymbol{\theta})}\right) \\ &= \sum_{i=1}^{r} \Big[\ln\left(\frac{\tilde{\nu}_0^+}{\nu_0^+}\right) - T\left\{\tilde{\nu}_0^+ \exp\{-\frac{(\eta_i - \tilde{b})^2}{2\tilde{\sigma}^2}\} - \nu_0^+ \exp\{-\frac{(\eta_i - b)^2}{2\sigma^2}\}\right\} \\ &\quad + \ln\left(\frac{\sigma^2}{\tilde{\sigma}^2}\right) + \ln\left(\frac{(\eta_i - \tilde{b})}{(\eta_i - b)}\right) - \tfrac{1}{2\tilde{\sigma}^2}(\eta_i - \tilde{b})^2 + \tfrac{1}{2\sigma^2}(n_i - b)^2 \Big]. \end{aligned}$$

Now, the parameter vector $\boldsymbol{\theta}$ is given as

$$\boldsymbol{\theta} = \begin{bmatrix} \nu_0^+ \\ b \\ \sigma \end{bmatrix}.$$

The new parameter b must also be given a prior distribution. As before, ν_0^+ will be normally distributed in one MCMC scheme and log-normally distributed in the second MCMC scheme. σ is normally distributed for both schemes and one will assume that b is also normally distributed, with a given mean and standard deviation. The mean in the prior distribution will be the same as the true value, namely 0 while the standard deviation is equal to 0.5. This gives the following prior distributions.

$$\begin{aligned} \nu_0^+ &\sim \begin{cases} \mathcal{N}(\mu_{\nu_0^+}, 0.5^2) \\ \text{Log-}\mathcal{N}(\mu_{\nu_0^+}, 0.5^2) \end{cases} \\ b &\sim \mathcal{N}(\mu_b, 0.5^2), \\ \sigma &\sim \mathcal{N}(\mu_\sigma, 0.5^2), \end{aligned} \tag{7.3}$$

where μ_* is the true value for the given parameter. In the following, the same three spectral densities as in equation (5.7) are used to generate time series, and the posterior distributions for the unknown parameters in (5.6) are simulated with MCMC.

7.2.1 Case I

Again spectrum I from (5.7) is used, now trying to estimate 3 parameters, in which the true values are known. It appears that it is more difficult to estimate the parameters from the simulated posterior distribution. Unlike the situation when only 2 parameters were to be estimated, the results in this case differ very much from the true values. The results is shown in table 7.4 below. Looking at the table, and comparing it to the true values, most of the estimated parameter values does not make any sense. The estimates varies a lot, even between the two different simulating schemes.

Table 7.4: Estimation of the parameters from two different MCMC schemes, with different block sizes. The data is generated by $S(\omega)_{\mathrm{I}}$ and the acceptance rate is shown in parenthesis.

Block length	Normal update for ν_0^+	Log-normal update for ν_0^+
600	$\nu_0^+ \approx 0.84(29\%)$	$\nu_0^+ \approx 1.14(22\%)$
	$b \approx 0.08(20\%)$	$b \approx -0.11(13\%)$
	$\sigma \approx 1.40(19\%)$	$\sigma \approx 1.42(13\%)$
2000	$\nu_0^+ \approx 0.94(41\%)$	$\nu_0^+ \approx 0.77(40\%)$
	$b \approx 0.11(25\%)$	$b \approx 0.24(25\%)$
	$\sigma \approx 1.39(23\%)$	$\sigma \approx 1.37(23\%)$
6000	$\nu_0^+ \approx 1.12(41\%)$	$\nu_0^+ \approx 0.85(54\%)$
	$b \approx 0.04(31\%)$	$b \approx 0.23(32\%)$
	$\sigma \approx 1.39(27\%)$	$\sigma \approx 1.36(28\%)$

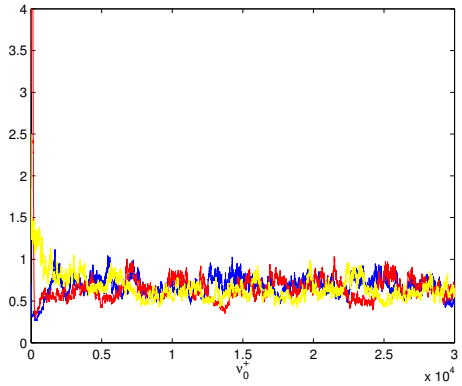
7.2.2 Case II

In this part, spectrum II in (5.7) is used. The parameter estimates are shown in table 7.5. The results for the two different schemes are now more consistent,

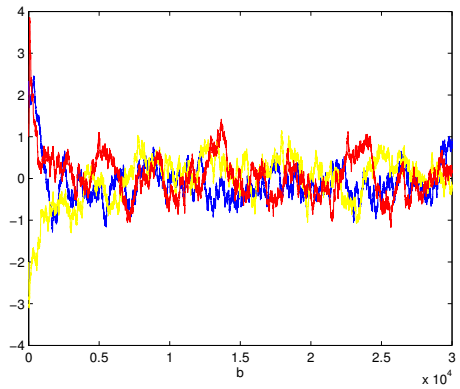
Table 7.5: Estimation of the parameters from two different MCMC schemes, with different block sizes. The time series is generated by $S(\omega)_{\text{II}}$. The acceptance rate is shown in parenthesis.

Block length	Normal update for ν_0^+	Log-normal update for ν_0^+
600	$\nu_0^+ \approx 0.78(29\%)$	$\nu_0^+ \approx 0.73(18\%)$
	$b \approx -0.24(20\%)$	$b \approx -0.20(33\%)$
	$\sigma \approx 4.03(19\%)$	$\sigma \approx 4.00(36\%)$
2000	$\nu_0^+ \approx 0.63(24\%)$	$\nu_0^+ \approx 0.58(28\%)$
	$b \approx -0.02(48\%)$	$b \approx 0.02(48\%)$
	$\sigma \approx 4.02(47\%)$	$\sigma \approx 4.01(48\%)$
4000	$\nu_0^+ \approx 0.53(28\%)$	$\nu_0^+ \approx 0.50(35\%)$
	$b \approx -0.03(57\%)$	$b \approx 0.06(58\%)$
	$\sigma \approx 4.08(55\%)$	$\sigma \approx 4.08(56\%)$

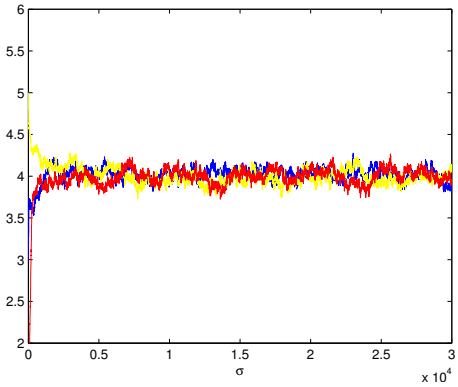
the two methods gives essentially the same estimates. It looks like the true values should be obtainable for a block size between 600 and 2000. In the following, a more detailed analysis is done on a simulation with block size equal to 1000. Three independent simulations, each with $1,000,000$ iterations are done with ν_0^+ log-normally distributed. Figure 7.8 shows the trace plot for all three parameters, starting from three different initial values. For the third parameter, σ, the trace plot is more stable around the mean compared to ν_0^+ and b. After approximately 5000 iterations, one may assume that the samples are from the posterior distribution of the parameters. Figure 7.9 shows the development of the parameter space, the blue area is the space for $1,000,000$ iterations, while the yellow area is the parameter space after $100,000$ iterations. Note that the true values lies well within the parameter spaces. Figure 7.10 shows the joint density for all parameters plotted against each other. Observe that the density for each parameter has the mode around the true values. In table 7.6 below, the mean of the posterior distribution for each parameter is used to get an estimate of the parameters. These estimates are very close to the true values.



(a) Trace plot of the mean upcrossing rate with 3 independent runs with different initial states. Blue: $\nu^+_{0\,0} = 0.5$, Yellow: $\nu^+_{0\,0} = 2.5$, Red: $\nu^+_{0\,0} = 3$

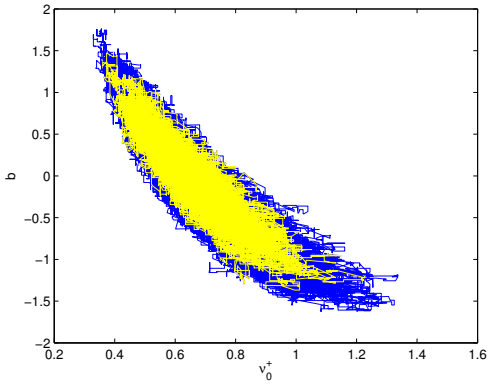


(b) Trace plot of the mean with 3 independent runs with different initial states. Blue: $b_0 = 2$, Yellow: $b_0 = -2$, Red: $b_0 = 1$.

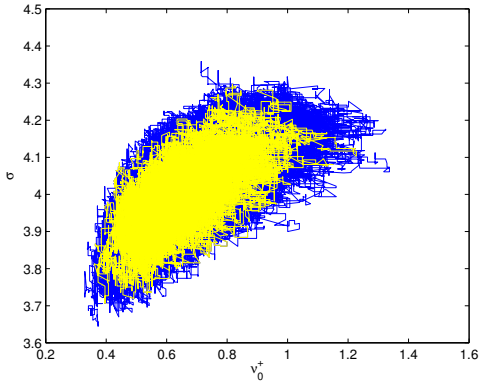


(c) Trace plot of the standard deviation with 3 independent runs with different initial states. Blue: $\sigma_0 = 3$, Yellow: $\sigma_0 = 5$, Red: $\sigma_0 = 1$.

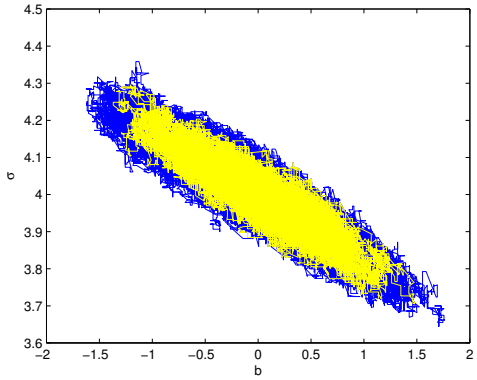
Figure 7.8: Trace plots of the mean upcrossing rate (a), the mean of the time series (b) and standard deviation (c) with 3 independent runs starting from different initial states from a simulation with a block length of 1000. The data is generated by $S(\omega)_{\text{II}}$.



(a) The parameter space for ν_0^+ and b.



(b) The parameter space for ν_0^+ and σ.

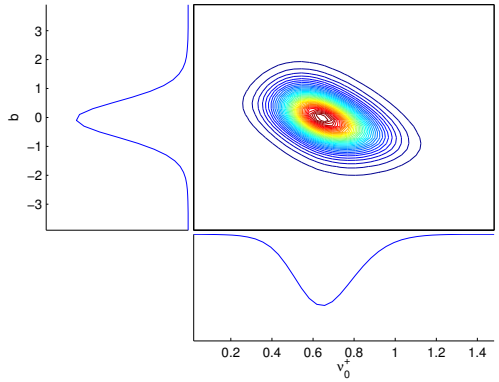


(c) The parameter space for b and σ.

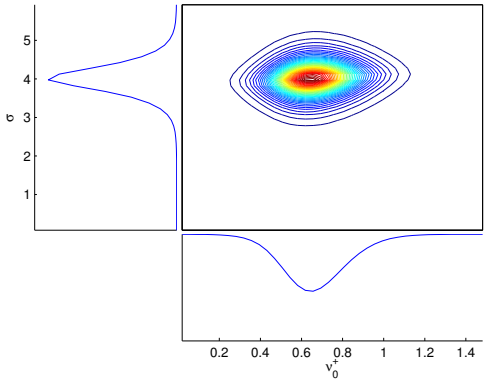
Figure 7.9: The parameter space for the three parameters plotted against each other. The blue space is after $1,000,000$ iterations, while the yellow space is after $100,000$ iterations. A burn-in period of the 5000 first iterations is discarded. The data is generated by $S(\omega)_{\text{II}}$.

Table 7.6: Estimation of the parameters from three independent simulations with ν_0^+ log-normally distributed, with a block size of 1000. The time series is generated by $S(\omega)_{\text{II}}$. All simulations have different initial values. The acceptance rate is shown in parenthesis.

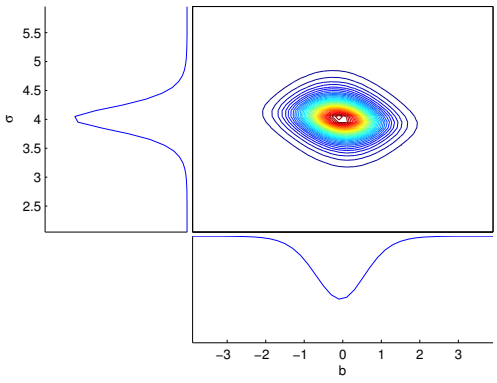
Block length	First run	Second run	Third run	True Value
ν_0^+	0.65(22%)	0.66(22%)	0.66(22%)	0.66
b	-0.07(39%)	-0.03(39%)	-0.05(39%)	0
σ	4.01(41%)	4.00(41%)	4.01(41%)	4



(a) The joint posterior density between ν_0^+ and b.



(b) The joint posterior density between ν_0^+ and σ.



(c) The joint posterior density between b and σ.

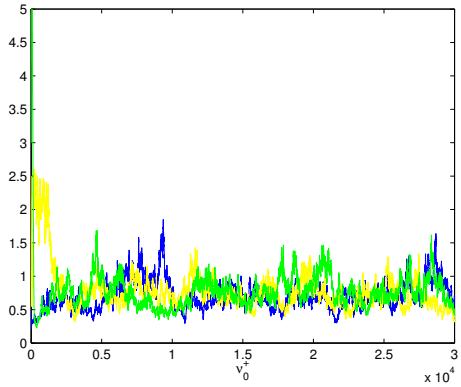
Figure 7.10: Joint posterior density between the three parameters. The data is generated by $S(\omega)_{\text{II}}$.

7.2.3 Case III

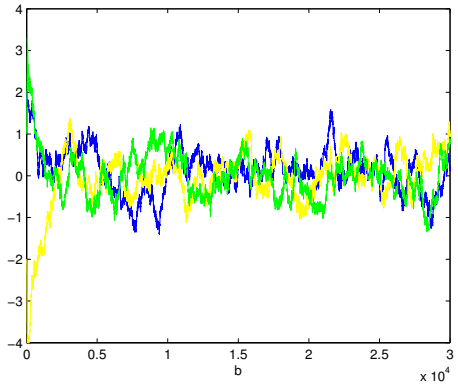
$S(\omega)_{\text{III}}$ is the most narrow banded spectrum in (5.7). For the simulation with block length equal to 600, the results differ a lot from the true values, but with block length equal to 2000, the mean of the posterior density for each parameter is almost exactly the true value. The further analysis is done with a block length of 2000. The number of iterations used is still $1,000,000$, and the trace plots are shown in figure 7.11. Visual inspection of the trace plots may suggest a burn-in period of 5000 iterations. After 5000 iteration, all three runs seems to mix well, and the chain has forgotten where it started. The development of parameter space is shown in 7.12. Looking at the plots, one may observe that the true values for both parameters plotted is well inside the parameter space, almost in the middle of the space. The yellow space is after $100,000$ iterations, while the blue space is after $1,000,000$ iterations. This because one may then see how the parameter space evolves as the number of iterations increase. If the parameter space for $1,000,000$ iterations had been very much larger then for $100,000$ iterations, this could have indicated that the chain was mixing very slow, i.e. it uses long time to explore the parameter space, hence the convergence would be problematic. In figure 7.13 the joint posterior density between the three parameters are shown. In table 7.7 the parameter estimates for the different posterior distributions with a block length of 2000 is found. The estimates are very close to the true values.

Table 7.7: Estimation of the parameters from three independent simulations with ν_0^+ log-normally distributed, with a block size of 2000. The time series is generated by $S(\omega)_{\text{III}}$. All simulations have different initial values. The acceptance rate is shown in paranthesis.

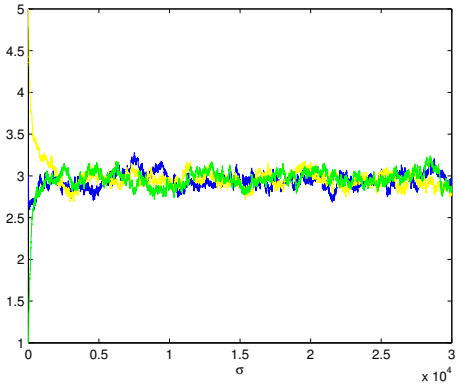
Block length	First run	Second run	Third run	True Value
ν_0^+	0.72(31%)	0.72(31%)	0.73(31%)	0.73
b	0.05(38%)	0.04(38%)	0.04(38%)	0
σ	2.95(37%)	2.95(37%)	2.95(37%)	3



(a) Trace plot of the mean upcrossing rate with 3 independent runs with different initial states. Blue: $\nu_{0\ 0}^{+} = 0.5$, Yellow: $\nu_{0\ 0}^{+} = 2.5$, Green: $\nu_{0\ 0}^{+} = 3$

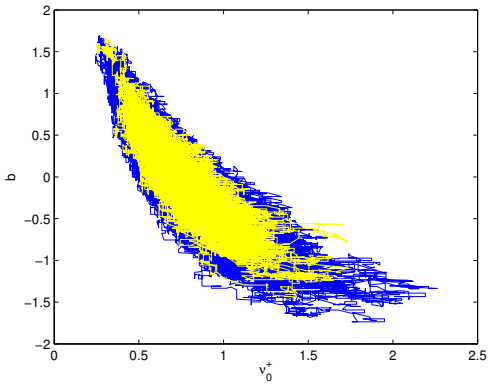


(b) Trace plot of the mean with 3 independent runs with different initial states. Blue: $b_0 = 2$, Yellow: $b_0 = -2$, Green: $b_0 = 1$.

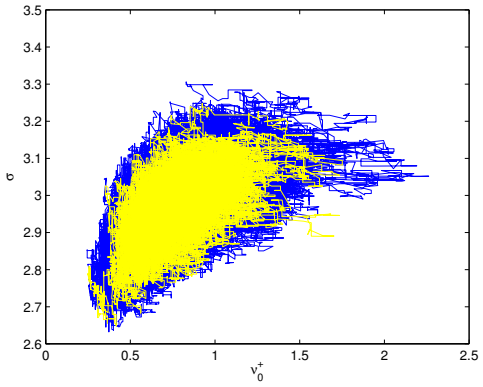


(c) Trace plot of the standard deviation with 3 independent runs with different initial states. Blue: $\sigma_0 = 3$, Yellow: $\sigma_0 = 5$, Green: $\sigma_0 = 1$.

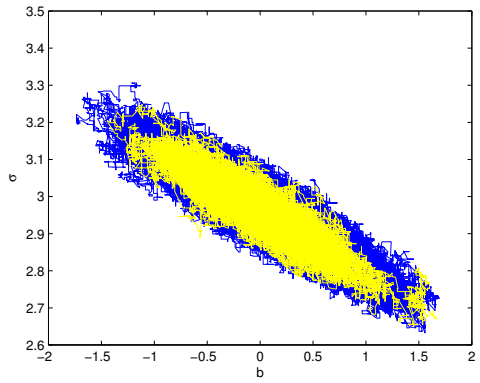
Figure 7.11: Trace plots of the mean upcrossing rate (a), the mean of the time series (b) and standard deviation (c) with 3 independent runs starting from different initial states from a simulation with a block length of 2000. The data is generated by $S(\omega)_{\text{III}}$.



(a) The parameter space for ν_0^+ and b.

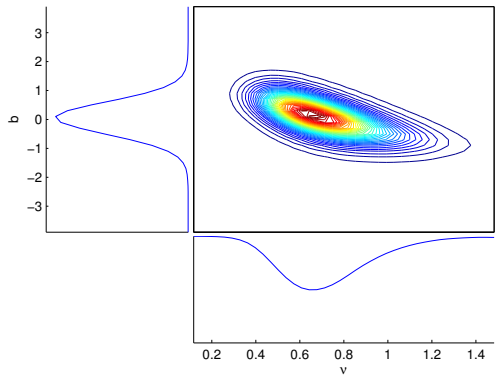


(b) The parameter space for ν_0^+ and σ.

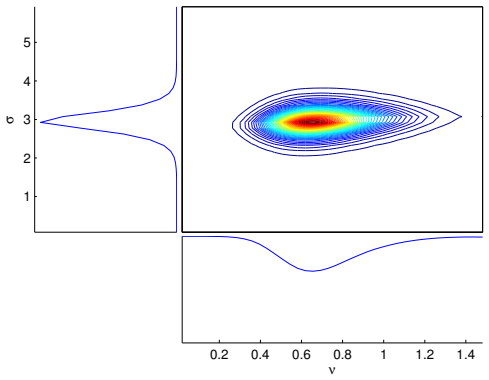


(c) The parameter space for b and σ.

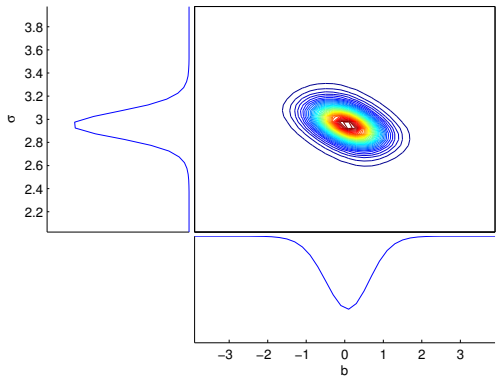
Figure 7.12: The parameter space for the three parameters to be estimated plotted against each other. The blue space is after $1,000,000$ iterations, while the yellow space is after $100,000$ iterations. The data is generated by $S(\omega)_{\text{III}}$.



(a) The joint posterior density between ν_0^+ and b.



(b) The joint posterior density between ν_0^+ and σ.



(c) The joint posterior density between b and σ.

Figure 7.13: Joint posterior density between the three parameters. The data is generated by $S(\omega)_{\text{III}}$.

7.3 Final extension of the model

Now the model in (5.6) has changed to

$$F(\eta) = \exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta-b)^c}{2\sigma^2}\}\right\}, \tag{7.4}$$

where a constant c has been added as an unknown. This is a shape parameter. As before, the PDF is found by taking the derivative of $F(\eta)$,

$$f(\eta) = \exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta-b)^c}{2\sigma^2}\}\right\} \cdot \nu_0^+ T \exp\{-\frac{(\eta-b)^c}{2\sigma^2}\} \cdot \frac{c(\eta-b)^{c-1}}{2\sigma^2}.$$

The new likelihood model then becomes

$$\begin{aligned} f(\boldsymbol{\eta}|\boldsymbol{\theta}) &= \sum_{i=1}^{r} f(\eta_i|\boldsymbol{\theta}) \\ &= \prod_{i=1}^{r} \left(\exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta-b)^c}{2\sigma^2}\}\right\} \cdot \nu_0^+ T \exp\{-\frac{(\eta-b)^c}{2\sigma^2}\} \cdot \frac{c(\eta-b)^{c-1}}{2\sigma^2} \right). \end{aligned}$$

This gives the following ratio

$$\begin{aligned} \frac{f(\boldsymbol{\eta}|\tilde{\boldsymbol{\theta}})}{f(\boldsymbol{\eta}|\boldsymbol{\theta})} &= \prod_{i=1}^{r} \left(\frac{f(\eta_i|\tilde{\boldsymbol{\theta}})}{f(\eta_i|\boldsymbol{\theta})} \right) \\ &= \prod_{i=1}^{r} \left(\frac{\exp\left\{-\tilde{\nu}_0^+ T \exp\{-\frac{(\eta_i-\tilde{b})^{\tilde{c}}}{2\tilde{\sigma}^2}\}\right\}}{\exp\left\{-\nu_0^+ T \exp\{-\frac{(\eta_i-b)^c}{2\sigma^2}\}\right\}} \cdot \frac{\tilde{\nu}_0^+ \exp\{-\frac{(\eta_i-\tilde{b})^{\tilde{c}}}{2\tilde{\sigma}^2}\} \cdot \frac{\tilde{c}(\eta_i-\tilde{b})^{\tilde{c}-1}}{2\tilde{\sigma}^2}}{\nu_0^+ \exp\{-\frac{(\eta_i-b)^c}{2\sigma^2}\} \cdot \frac{c(\eta_i-b)^{c-1}}{2\sigma^2}} \right), \end{aligned}$$

which on log-level simplifies to

$$\begin{aligned} \ln\left(\frac{f(\boldsymbol{\eta}|\tilde{\boldsymbol{\theta}})}{f(\boldsymbol{\eta}|\boldsymbol{\theta})}\right) &= \sum_{i=1}^{r} \ln\left(\frac{f(\eta_i|\tilde{\boldsymbol{\theta}})}{f(\eta_i|\boldsymbol{\theta})}\right) \\ &= \sum_{i=1}^{r} \Big[\ln\left(\frac{\tilde{\nu}_0^+}{\nu_0^+}\right) - T\left\{\tilde{\nu}_0^+ \exp\{-\frac{(\eta_i-\tilde{b})^{\tilde{c}}}{2\tilde{\sigma}^2}\} - \nu_0^+ \exp\{-\frac{(\eta_i-b)^c}{2\sigma^2}\}\right\} \\ &\quad + \ln\left(\frac{\sigma^2}{\tilde{\sigma}^2}\right) + (\tilde{c}-1)\ln\Big((\eta_i-\tilde{b})\Big) - (c-1)\ln\Big((\eta_i-b)\Big) + \ln\left(\frac{\tilde{c}}{c}\right) \\ &\quad - \tfrac{1}{2\tilde{\sigma}^2}(\eta_i-\tilde{b})^{\tilde{c}} + \tfrac{1}{2\sigma^2}(n_i-b)^c \Big]. \end{aligned}$$

The new extended parameter vector $\boldsymbol{\theta}$ is now

$$\boldsymbol{\theta} = \begin{bmatrix} \nu_0^+ \\ b \\ \sigma \\ c \end{bmatrix}.$$

The prior distributions for the parameters are the same as the previous case, and the new parameter c is given the same prior as the mean upcrossing rate, namely a log-normal distribution, with the true value as the mean value. As for the mean upcrossing rate and the standard deviation, the parameter c cannot take negative values. Also, only the algorithm using a log-normal proposal for the mean upcrossing rate has been used. The prior distributions are as follows.

$$\begin{aligned} \nu_0^+ &\sim \text{Log-}\mathcal{N}(\mu_{\nu_0^+}, 0.5^2), \\ b &\sim \mathcal{N}(\mu_b, 0.5^2), \\ \sigma &\sim \mathcal{N}(\mu_\sigma, 0.5^2), \\ c &\sim \text{Log-}\mathcal{N}(\mu_c, 0.5^2), \end{aligned} \tag{7.5}$$

where μ_* is the true value for the given parameter.

7.3.1 Case I

This case gave very accurate results when the unknown parameter vector only consisted of two parameters. When this was extended to three parameters, the two different simulation schemes gave very different results. In table 7.8, the estimates for the four unknown parameters are presented. These results do not coincide with the true values. From the table, one may say that one will obtain a good estimate when the block size is around 2000 points, maybe a bit less. All 4 parameters are roughly around the true values. The block length is hard to know in practice using real data sets, where the true values are unknown.

Table 7.8: Estimation of the parameters with different block sizes. The time series is generated by $S(\omega)_{\text{I}}$. The acceptance rate is shown in parenthesis.

Parameter	1000	2000	4000	6000	True value
ν_0^+	1.25(26%)	0.95(34%)	0.84(42%)	0.83(48%)	1.02
b	0.06(14%)	−0.01(19%)	0.11(25%)	0.24(30%)	0
σ	1.31(13%)	1.46(19%)	1.47(24%)	1.27(25%)	$\sqrt{2}$
c	1.91(15%)	2.02(18%)	2.04(22%)	1.90(26%)	2

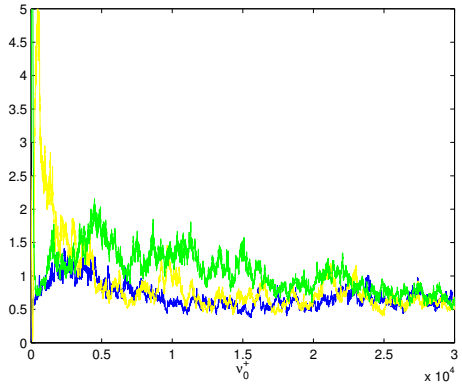
7.3.2 Case II

For the third time, spectrum $S(\omega)_{\text{II}}$ is used in the process of generating the time series. In table 7.9 the results of the parameter estimation is shown. As in section 7.2.2, a block length of 1000 gives the best result compared to the true values, even though the estimate of the standard deviation is more correct when the block length is 2000. Figure 7.14 shows the trace plots for the simulation with

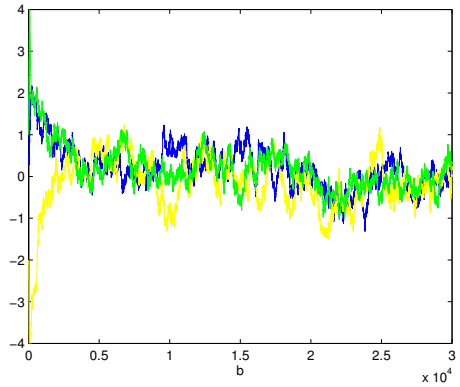
Table 7.9: Estimation of the parameters with different block sizes. The time series is generated by $S(\omega)_{\text{II}}$. The acceptance rate is shown in parenthesis.

Parameter	1000	2000	4000	6000	True value
ν_0^+	0.68(22%)	0.59(29%)	0.46(34%)	0.59(43%)	0.66
b	0.02(39%)	0.13(48%)	0.23(58%)	0.13(61%)	0
σ	3.87(39%)	4.00(45%)	4.21(57%)	4.16(59%)	4
c	1.98(10%)	1.99(12%)	2.02(15%)	2.02(16%)	2

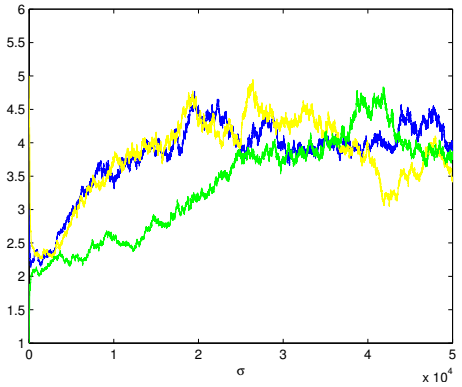
block length equal to 1000. The parameters does not have the same burn in period, ν_0^+ seems to have converged after 25000 iterations while b only uses 5000 iterations. The standard deviation σ uses approximately 40000 while the shape parameter c uses 30000 iterations. In figure 7.15 the parameter spaces are shown for all parameters plotted against each other. In the figure, (f) seems to have a strong positive correlation. This is the plot between the standard deviation σ and the shape parameter c. It may also look like the plot in (c) between the mean upcrossing rate ν_0^+ and the shape parameter c indicates a small negative correlation. Figure 7.16 shows the joint density between all parameters.



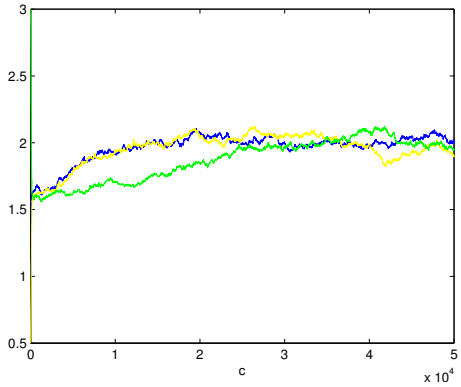
(a) Trace plot of the mean upcrossing rate with 3 independent runs with different initial states. Blue: $\nu_{0\,0}^{+} = 0.5$, Yellow: $\nu_{0\,0}^{+} = 2.5$, Green: $\nu_{0\,0}^{+} = 3$



(b) Trace plot of the mean with 3 independent runs with different initial states. Blue: $b_0 = 2$, Yellow: $b_0 = -2$, Green: $b_0 = 1$.

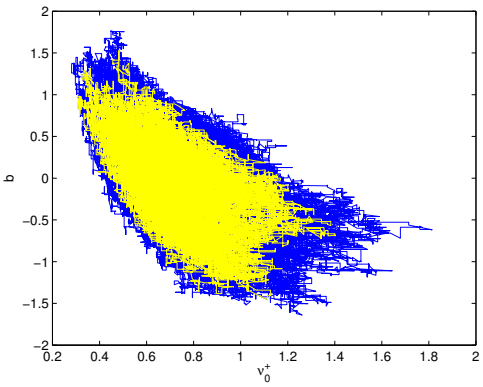


(c) Trace plot of the standard deviation with 3 independent runs with different initial states. Blue: $\sigma_0 = 5$, Yellow: $\sigma_0 = 2$, Green: $\sigma_0 = 1$.

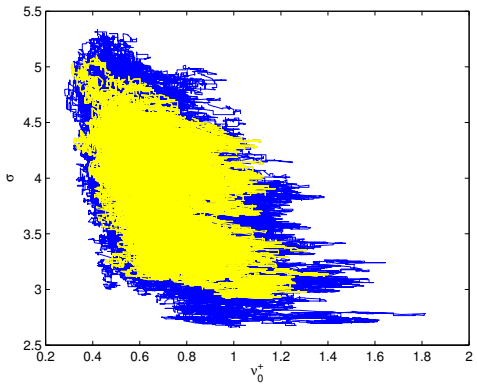


(d) Trace plot of the shape parameter with 3 independent runs with different initial states. Blue: $c_0 = 1$, Yellow: $c_0 = 0.5$, Green: $c_0 = 3$.

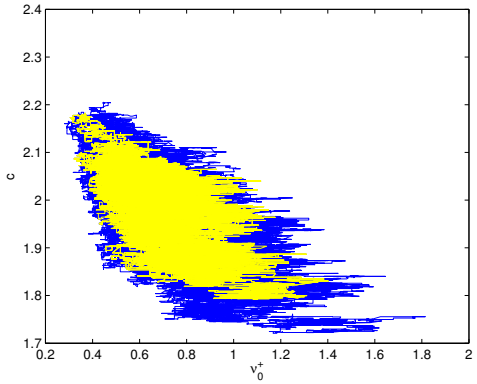
Figure 7.14: Trace plots of the mean upcrossing rate (a), the mean of the time series (b), the standard deviation (c) and the shape parameter (d) with 3 independent runs starting from different initial states from a simulation with a block length of 4000.



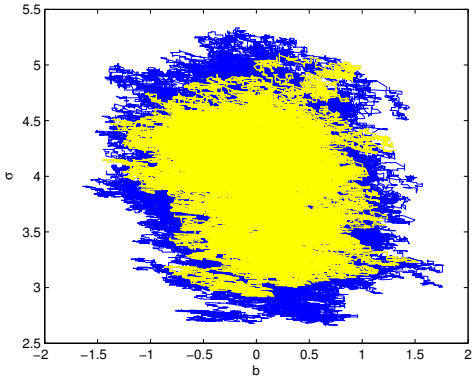
(a) The parameter space for ν_0^+ and b.



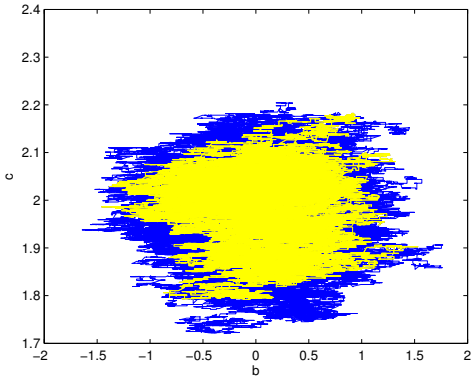
(b) The parameter space for ν_0^+ and σ.



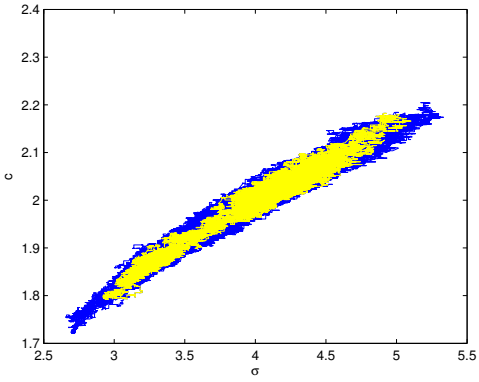
(c) The parameter space for ν_0^+ and c.



(d) The parameter space for b and σ.

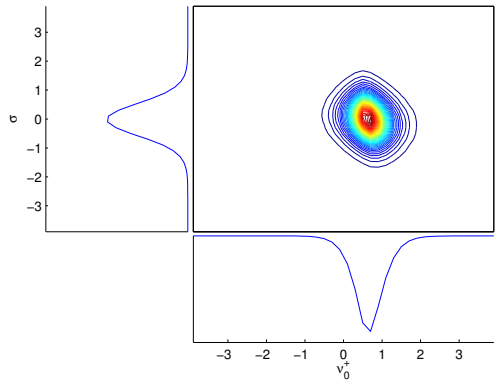


(e) The parameter space for b and c.

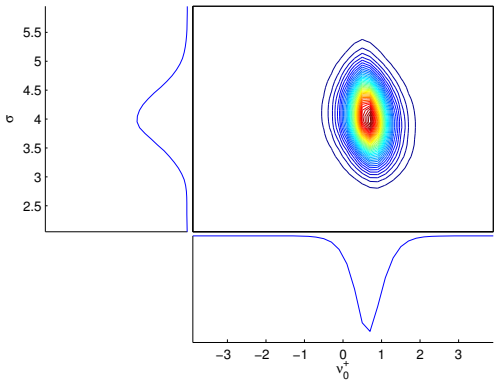


(f) The parameter space for σ and c.

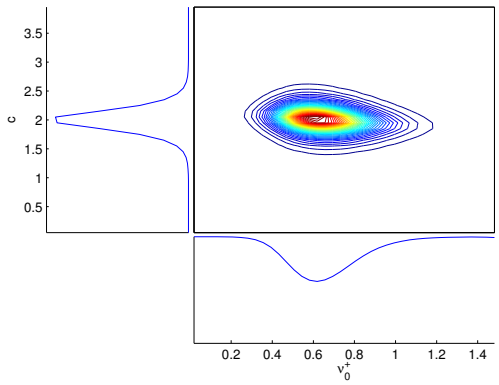
Figure 7.15: The parameter space for the four parameters plotted against each other. The blue space is after $1,000,000$ iterations, while the yellow space is after $200,000$ iterations. A burn-in period of the 40000 first iterations is discarded. The data is generated by $S(\omega)_{\text{II}}$.



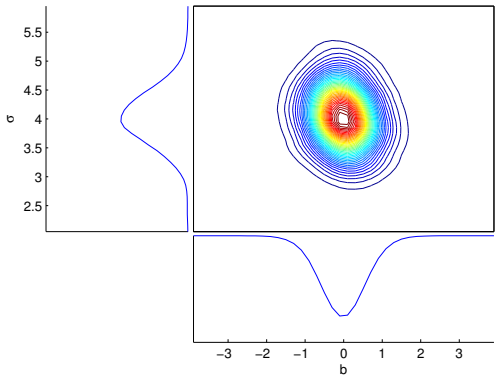
(a) The joint posterior density between ν_0^+ and b.



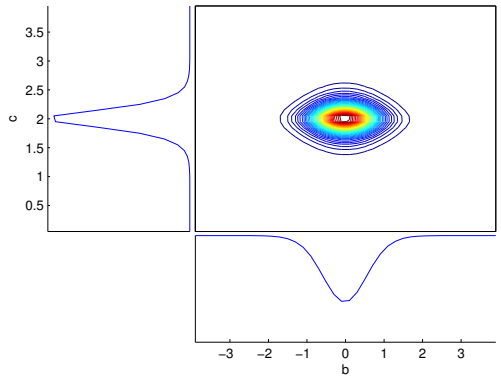
(b) The joint posterior density between ν_0^+ and σ.



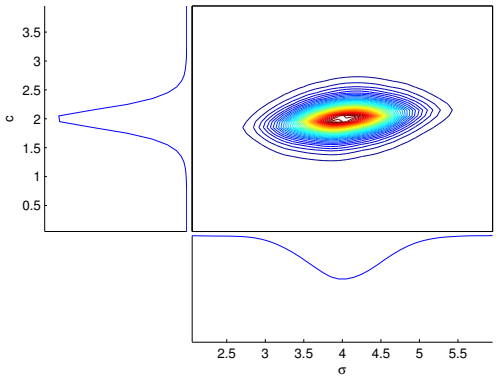
(c) The joint posterior density between ν_0^+ and c.



(d) The joint posterior density between b and σ.



(e) The joint posterior density between b and c.



(f) The joint posterior density between σ and c.

Figure 7.16: The joint posterior density between the four parameters plotted against each other. The data is generated by $S(\omega)_{\text{II}}$.

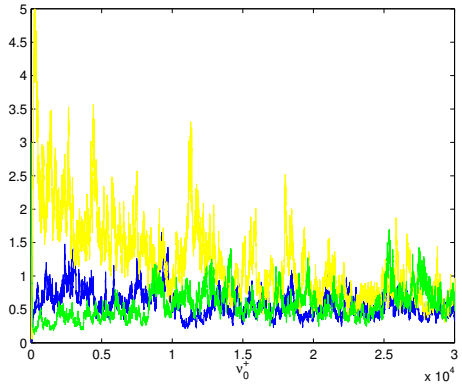
7.3.3 Case III

The result for the analysis of time series generated by $S(\omega)_{\text{III}}$ is found in table 7.10. Here, the simulations with block length 4000 and 6000 gives the best estimates. Trace plots for three independent simulations with block length 4000 is found in figure 7.17. The chain now uses more iterations to reach convergence than before. For the parameter σ and c, the chain seems to have reached convergence after $50,000$ iterations. This is not surprising, since there are 4 parameters to be estimated, and the uncertainty in each parameter should be bigger. In figure 7.18

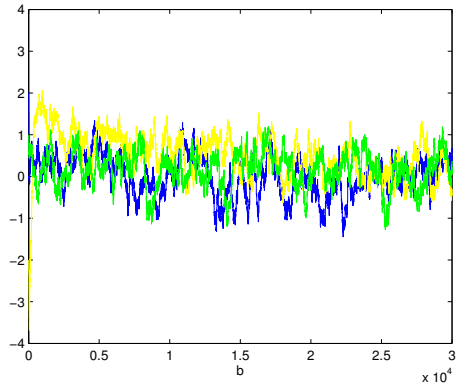
Table 7.10: Estimation of the parameters with different block sizes. The time series is generated by $S(\omega)_{\text{III}}$. The acceptance rate is shown in parenthesis.

Parameter	2000	4000	6000	True value
ν_0^+	0.87(33%)	0.69(39%)	0.76(47%)	0.73
b	0.23(38%)	0.09(47%)	0.14(51%)	0
σ	2.55(33%)	2.99(44%)	2.95(47%)	3
c	1.89(13%)	1.99(16%)	1.99(17%)	2

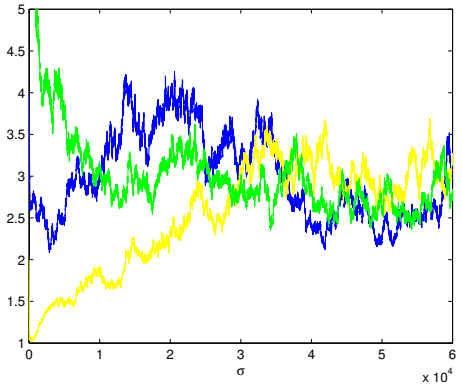
the parameter space for all parameters plotted against each other are shown. As in case II, there is a strong positive correlation between the standard deviation σ and the shape parameter c. Some of the other parameter spaces, (a), (b) and (c) may also have a little correlation between the parameters. The joint density between all parameters is shown in figure 7.19.



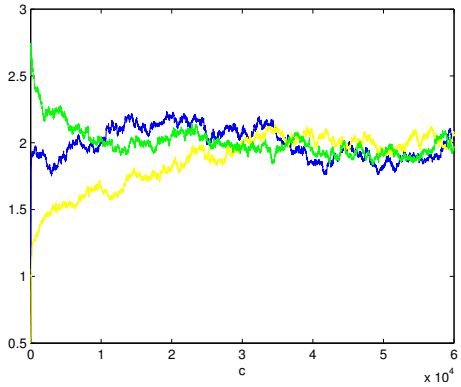
(a) Trace plot of the mean upcrossing rate with 3 independent runs with different initial states. Blue: $\nu_{0\,0}^{+} = 0.5$, Yellow: $\nu_{0\,0}^{+} = 2.5$, Green: $\nu_{0\,0}^{+} = 3$



(b) Trace plot of the mean with 3 independent runs with different initial states. Blue: $b_0 = 2$, Yellow: $b_0 = -2$, Green: $b_0 = 1$.

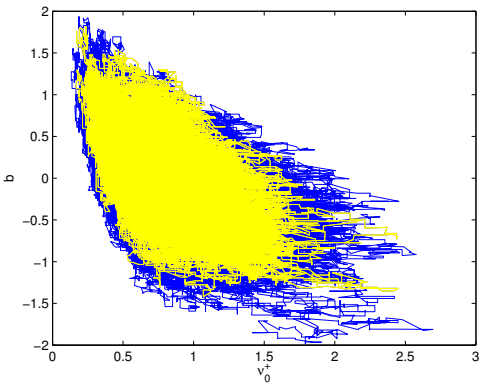


(c) Trace plot of the standard deviation with 3 independent runs with different initial states. Blue: $\sigma_0 = 4$, Yellow: $\sigma_0 = 2$, Green: $\sigma_0 = 6$.

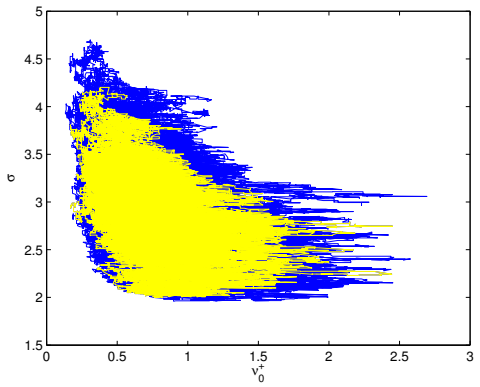


(d) Trace plot of the standard deviation with 3 independent runs with different initial states. Blue: $c_0 = 1$, Yellow: $c_0 = 0.5$, Green: $c_0 = 2.5$.

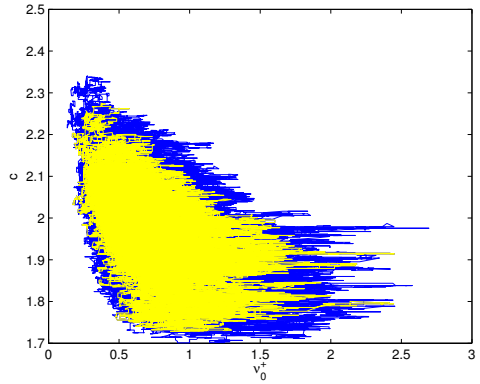
Figure 7.17: Trace plots of the mean upcrossing rate (a), the mean of the time series (b), the standard deviation (c) and the shape parameter (d) with 3 independent runs starting from different initial states from a simulation with a block length of 4000. The data is generated by $S(\omega)_{\mathrm{III}}$.



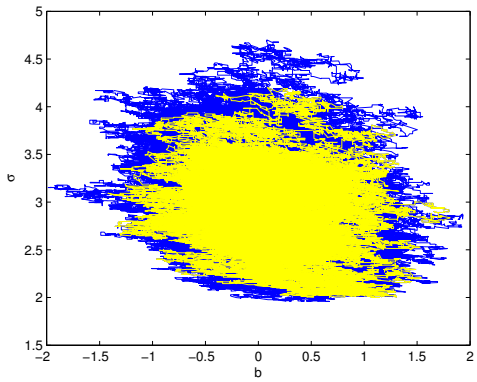
(a) The parameter space for ν_0^+ and b.



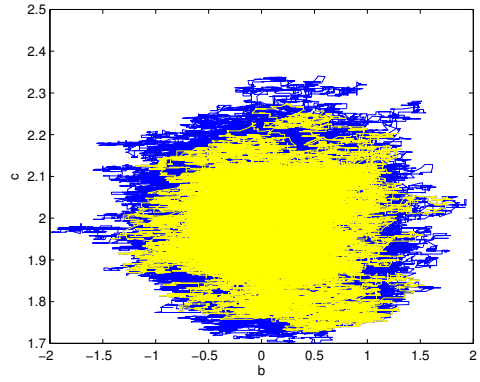
(b) The parameter space for ν_0^+ and σ.



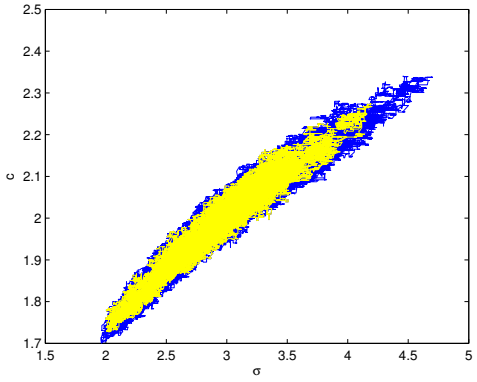
(c) The parameter space for ν_0^+ and c.



(d) The parameter space for b and σ.

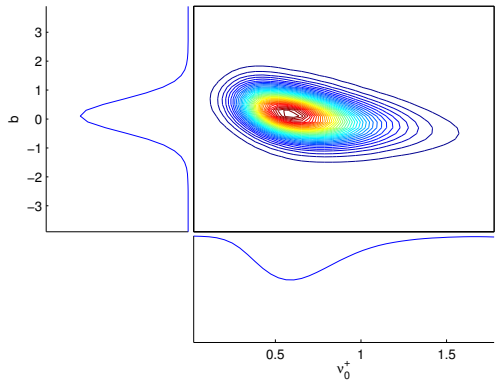


(e) The parameter space for b and c.

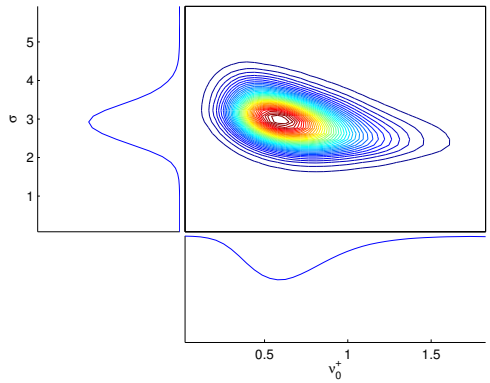


(f) The parameter space for σ and c.

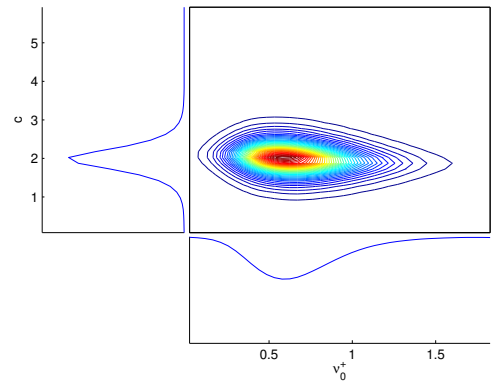
Figure 7.18: The parameter space for the four parameters plotted against each other. The blue space is after $1,000,000$ iterations, while the yellow space is after $200,000$ iterations. A burn-in period of the 40000 first iterations is discarded. The data is generated by $S(\omega)_{\text{III}}$.



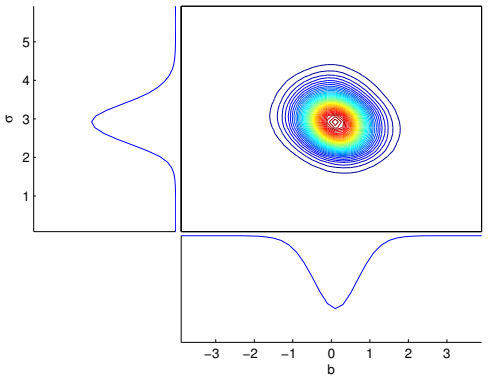
(a) The joint posterior density between ν_0^+ and b.



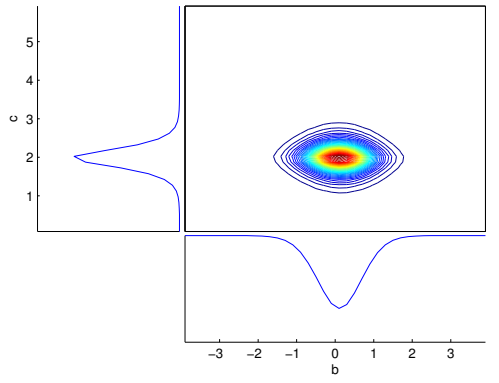
(b) The joint posterior density between ν_0^+ and σ.



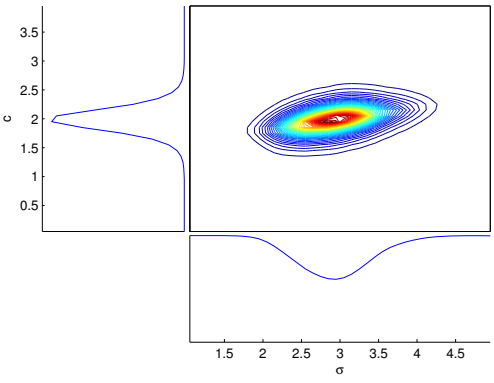
(c) The joint posterior density between ν_0^+ and c.



(d) The joint posterior density between b and σ.



(e) The joint posterior density between b and c.



(f) The joint posterior density between σ and c.

Figure 7.19: The joint posterior density between the four parameters plotted against each other. The data is generated by $S(\omega)_{\mathrm{III}}$.

7.4 Prior influence

All simulations so far have been done with a prior for each parameter, which have had the true value as the expected value. This is not the situation for real data sets, since the true values are unknown. In this section, some simulations have been repeated, but with different prior parameters.

7.4.1 2 unknown parameters

In table 7.11, a comparison is done for both spectrums analysed where 2 parameters were to be estimated. In this example, the prior does not have much influence on the result, the estimates are almost equal in both simulations. The prior distributions for the parameters were

$$\nu_0^+ \sim \text{Log-}\mathcal{N}(0.8, 0.5^2),$$
$$\sigma \sim \mathcal{N}(2.5, 0.5^2).$$

Table 7.11: Estimation of the parameters with different prior information for a block length of 600. The mean value in the prior different from the true values is 0.8 for ν_0^+ and 2.5 for σ. Both priors have the same standard deviation, which is 0.5. The data is generated by $S(\omega)_{\text{I}}$.

Spectrum	True values as prior	Prior different from true values	True value
$S(\omega)_{\text{I}}$	$\nu_0^+ \approx 1.01$	$\nu_0^+ \approx 1.00$	1.02
	$\sigma \approx 1.40$	$\sigma \approx 1.38$	$\sqrt{2}$
$S(\omega)_{\text{II}}$	$\nu_0^+ \approx 0.65$	$\nu_0^+ \approx 0.66$	0.66
	$\sigma \approx 3.97$	$\sigma \approx 3.97$	4

7.4.2 3 unknown parameters

In section 7.2.2, a block length of 1000 gave the best results from the MCMC simulation. In table 7.12 below, a new simulation has been done, but with priors different from the ones used in the previous simulations, which used the true values

in the prior for each parameter. Now, the following prior distributions have been used where the parameters is the mean and the variance.

$$\begin{aligned} \nu_0^+ &\sim \text{Log-}\mathcal{N}(1, 0.5^2), \\ b &\sim \mathcal{N}(0.5, 0.5^2), \\ \sigma &\sim \mathcal{N}(2, 0.5^2). \end{aligned}$$

Unlike table 7.11, the prior seems to be of interests in table 7.12, since the esti-

Table 7.12: Estimation of the parameters with different prior information with a block size of 1000. The mean value in the prior different from the true values is $\nu_0^+ = 1$, $b = 0.5$ and $\sigma = 2$. All priors have the same standard deviation, which is 0.5. The data is generated by $S(\omega)_{\text{II}}$.

Parameters	True values as prior	Prior different from true values	True value
ν_0^+	0.66	0.55	0.66
b	-0.03	0.49	0
σ	4.00	3.90	4

mation from the simulated posterior distributions gives more inaccurate responses compared to the results obtained when the prior distributions were as in equation (7.3).

7.4.3 4 unknown parameters

In section 7.3.3, a simulation with block length 4000 gave the best results. The simulation has been done again, but now with different prior information. The prior distribution is as follows, with the mean and the variance written as

$$\begin{aligned} \nu_0^+ &\sim \text{Log-}\mathcal{N}(1, 0.5^2), \\ b &\sim \mathcal{N}(0.5, 0.5^2), \\ \sigma &\sim \mathcal{N}(4, 0.5^2), \\ c &\sim \text{Log-}\mathcal{N}(1.5, 0.5^2). \end{aligned}$$

In table 7.13 the results of how the prior influence the estimation can be seen. The data is generated by $S(\omega)_{\text{III}}$. The estimates differ more from the true values in comparison to the priors in (7.5). Thus the prior have an influence on the results for this simulation, compared to the results when the priors were as in equation (7.5).

Table 7.13: Estimation of the parameters with different prior information with a block size of 4000. The mean value in the prior different from the true values is $\nu_0^+ = 1$, $b = 0.5$, $\sigma = 4$ and $c = 1.5$. All priors have the same standard deviation, which is 0.5. The data is generated by $S(\omega)_{\text{III}}$.

Parameters	True values as prior	Prior different from true values	True value
ν_0^+	0.69	0.55	0.73
b	0.09	0.10	0
σ	2.99	3.36	3
c	1.99	2.08	2

7.5 Difficulties determining the block length

In all analysis, it is a problem to determine how long the blocks shall be, when simulating the posterior density for each parameter. When the answer is known, as in this analysis, it is easy to choose the block length that gives the estimates which are closest to the true values. However, if the method is used on real data, one does not know the true values, and there does not exist a method for determining a suitable block length. What is the best length varies also between each case. When 2 parameters are to be estimated, a block length of 400 and 600 gives the best results for $S(\omega)_{\text{I}}$ and $S(\omega)_{\text{II}}$, while when 3 parameters are to be estimated, a block length of 1000 gives the best result for $S(\omega)_{\text{II}}$ and 2000 gives the best results for $S(\omega)_{\text{III}}$. With 4 unknown parameters, a block length of 1000 gives the most accurate estimates for $S(\omega)_{\text{II}}$, while for $S(\omega)_{\text{III}}$ between 4000 and 6000 gives estimates closest to the true values. It looks like the more narrow banded the spectrum is, the bigger block lengths have to be used. This is tested in the next section.

7.6 A narrow banded spectrum

Introduce a new spectrum,

$$S(\omega)_{\text{IV}} = 4 \cdot \chi_{[4,5]}.$$

Compared to the spectrums in expression (5.7), this is a very narrow banded spectrum. Looking at figure 7.20, which is the first 2000 points of the generated

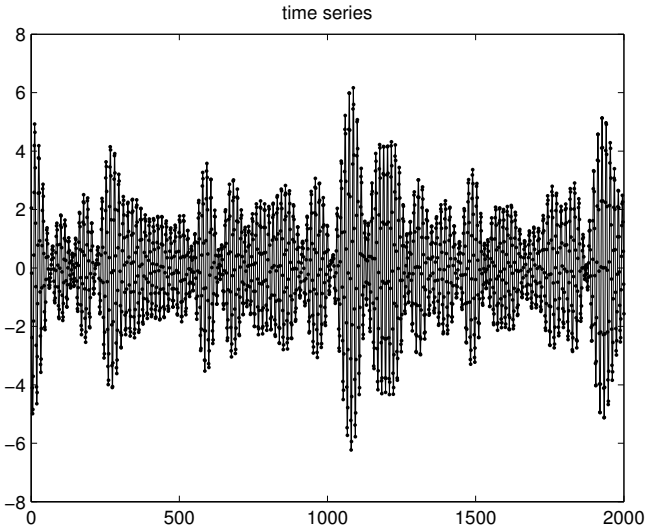


Figure 7.20: First 2000 points of a time series generated by $S(\omega)_{\text{IV}} = 4 \cdot \chi_{[4,5]}(\omega)$.

time series, and comparing it to the corresponding plots for case I, II and III, one may observe that the time series is more smooth as the spectrum becomes more narrow banded. The peaks seems to have a little correlation compared to the time series in I, II and III, where there is no sign of dependency between the peaks. The true values for the mean upcrossing rate and the standard deviation for this spectrum is given by the formulas in expression (5.4) and (5.5). In the extreme value model in (7.4), the true parameter values are

$$\begin{aligned} \nu_0^+ &\approx 0.72, \\ b &= 0, \\ \sigma &= 2, \\ c &= 2. \end{aligned}$$

The prior distribution is as follows

$$\begin{aligned} \nu_0^+ &\sim \text{Log-}\mathcal{N}(\mu_{\nu_0^+}, 0.5^2), \\ b &\sim \mathcal{N}(\mu_b, 0.5^2), \\ \sigma &\sim \mathcal{N}(\mu_\sigma, 0.5^2), \\ c &\sim \text{Log-}\mathcal{N}(\mu_c, 0.5^2), \end{aligned}$$

where μ_* is the true value for the given parameter. In table 7.14 the results of the parameter estimation is shown. The estimates are found by computing the

mean value of the simulated posterior distribution. The number of iterations are $1,000,000$. The estimation of ν_0^+ differs from the true value for all different block lengths. Among the other parameters, some seems to be close to the true values, but the estimates are not consistent in any way.

Table 7.14: Estimation of the parameters with different block sizes. The time series is generated by $S(\omega)_{\text{IV}}$. The acceptance rate is shown in parenthesis.

Parameter	500	6000	10000	14000	True value
ν_0^+	0.37(12%)	0.31(33%)	0.42(42%)	0.47(51%)	0.72
b	0.16(20%)	0.10(44%)	0.03(45%)	0.11(52%)	0
σ	1.45(17%)	1.81(38%)	1.91(40%)	1.79(43%)	2
c	1.68(16%)	1.87(26%)	1.94(28%)	1.91(30%)	2

Chapter 8

Closing Remarks

After the different analysis done in chapter 7, there is one big problem. Although the MCMC method is able to find very accurate estimates for the parameters in the extreme value model, different block lengths give different parameter estimates. The "correct" block length also varies among the different spectrums used in the generation of the data.

When spectrum $S(\omega)_{\mathrm{I}}$, which is a large spectrum, is analysed, the algorithm does not obtain good estimates for the situation with 4 unknown parameters. Spectrum $S(\omega)_{\mathrm{II}}$ is more narrow banded, and good estimates are obtained for a block size of 1000. $S(\omega)_{\mathrm{III}}$ is again more narrow banded, and for a block length of 4000 the algorithm obtains very good estimates for the true values. Even though it looked like a big block length should give a more precise approximation to the true value for a narrow banded spectrum, this was not the case. For $S(\omega)_{\mathrm{IV}}$ the algorithm was unable to find a suitable estimate for the parameters for all block lengths tested. Some parameters were close, but overall the algorithm was not consistent.

In stochastic simulation, many different choices have to be made. Some choices can be smart in a way such that the simulation will reach convergence faster. Other choices may be unwise, and result in a CPU intensive algorithm. Convergence is not guaranteed at all in MCMC simulation. In the simulations done in this thesis, many choices can be made different, for instance the choice of prior distributions and the likelihood. The likelihood is perhaps the largest element of uncertainty in the algorithm, because of the block lengths that have to be chosen, and there does not exist any formulas to determine a suitable block length.

In analysis of synthetic data, the extremes are evenly distributed. This may not be the case in real data sets. For instance, the extremes in wave data may not be evenly distributed in the time series. There can be bigger waves in one sea-

son, compared to another season. This gives a problem in determining the block length, since many extremes may be discarded, and many values that should not be regarded as extremes will contribute in the likelihood. One way to solve this problem can be to let the block length vary, such that the length is shorter for the season where there are most extremes, and long block lengths where there are few values that can be regarded as extremes.

Another likelihood can be to use every single observation, and introduce a threshold. All values above the threshold is put into the PDF of the given model, while for all points under the threshold, the CDF is used to express the probability for being under the given threshold. This method will most likely be more computer intensive, since more points is taken into account. Another issue is how to determine a suitable threshold. However, this method can also result in better estimates, since there are more data taken into account.

Bibliography

[1] S. Coles. *An introduction to statistical modeling of extreme values*. Springer Verlag, 2001.

[2] C.P. Robert and G. Casella. *Monte Carlo statistical methods*. Springer Verlag, 1999.

[3] H. Rue and L. Held. *Gaussian Markov random fields: theory and applications*. Chapman & Hall, 2005.

[4] D. Gamerman and H.F. Lopes. *Markov chain Monte Carlo: stochastic simulation for Bayesian inference*. Chapman & Hall/CRC, 2006.

[5] E.M. Stein and R. Shakarchi. *Real analysis: measure theory, integration, and Hilbert spaces*. Princeton University Press, 2005.

[6] W. K. Hastings. Monte carlo sampling methods using markov chains and their applications. *Biometrika*, 57(1):97–109, 1970.

[7] H. Tjelmeland. Lecture notes in subject tma4300, 2009.

[8] Naess A. Svigninger og dynamisk respons. compendium in subject 37076. k-2-94, 1998.

[9] A. Naess and O. Gaidai. Estimation of extreme values from sampled time series. *Structural Safety*, 31(4):325–334, 2009.

[10] Myrhaug D. Marin dynamikk - uregelmessig sjø. compendium in subject tmr4180, 2007.

Appendix A

Full conditionals

A.1 Conjugate distributions

A conjugate distribution is when a distribution given a prior distribution results in a posterior distribution of the same kind as the prior distribution. An example taken from [7] follows.

Example 1. *Let*

$$\pi(x|p) \sim \binom{n}{x} p^x (1-p)^{n-x},$$
$$\pi(p) \sim \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1}.$$

Then the posterior distribution is

$$\pi(p|x) \sim \frac{\Gamma(\tilde{\alpha}+\tilde{\beta})}{\Gamma(\tilde{\alpha})\Gamma(\tilde{\beta})} p^{\tilde{\alpha}-1}(1-p)^{\tilde{\beta}-1},$$

where

$$\tilde{\alpha} = \alpha + x,$$
$$\tilde{\beta} = \beta + n - x.$$

This is the same distribution as $\pi(p)$, namely a beta distribution, but with different parameters. One may say that the family of beta distributions is conjugate to the binomial distribution.

A.2 Conditional conjugacy

The previous section can be extended to conditional conjugacy, where one may be unable to find $\pi(\mu, \sigma | x_1, ..., x_n)$, but one may obtain the full conditionals given apriori independence between μ and σ^2.

Example 2. *Let*

$$x_1, .., x_n \overset{iid}{\sim} \mathcal{N}(\mu, \sigma^2),$$

where μ and σ^2 are independent apriori,

$$\pi(\mu, \sigma^2) = \pi(\mu) \cdot \pi(\sigma^2) \tag{A.1}$$

and assume

$$\begin{aligned} \mu &\sim \mathcal{N}(\mu_0, \tau^2) \quad \textit{(Conjugate prior distribution if } \sigma^2 \textit{ is known)}, \\ \sigma^2 &\sim \mathcal{IG}(\alpha, \beta) \quad \textit{(Conjugate prior distribution if } \mu \textit{ is known)}. \end{aligned}$$

Then the posterior distribution becomes

$$\pi(\mu, \sigma^2 | x_1, ..., x_n) \propto \exp\{-\tfrac{1}{2\tau^2}(\mu - \mu_0)^2\} \cdot \frac{\exp\{\frac{1}{\beta\sigma^2}\}}{(\sigma^2)^{\alpha-1}} \cdot \prod_{i=1}^{n} \left(\tfrac{1}{\sigma} \exp\{-\tfrac{1}{2\sigma^2}(x_i - \mu_0)^2\} \right).$$

This distribution can not be written as a product of a normal density for μ and inverse gamma for σ^2. However, because of (A.1)*,*

$$\begin{aligned} \pi(\mu | x_1, ..., x_n, \sigma^2) &= \frac{\pi(\mu, \sigma^2, x_1, ..., x_n)}{\pi(\sigma^2 | x_1, ..., x_n)} \\ &\propto \pi(\mu, \sigma^2, x_1, ..., x_n) \\ &\propto \exp\left\{-\tfrac{1}{2\tau^2}(\mu - \mu_0)^2 - \tfrac{1}{2\sigma^2}(x_i - \mu_0)^2\right\} \\ &\Downarrow \\ \pi(\mu | x_1, ..., x_n, \sigma^2) &\sim \mathcal{N}\left(\frac{\frac{\mu_0}{\tau^2} + \frac{n}{\sigma^2}\bar{x}}{\frac{1}{\tau^2} + \frac{n}{\sigma^2}}, \frac{1}{\frac{1}{\tau^2} + \frac{n}{\sigma^2}} \right) \end{aligned}$$

and

$$\begin{aligned} \pi(\sigma^2 | x_1, ..., x_n, \mu) &= \frac{\pi(\sigma^2, \mu, x_1, ..., x_n)}{\pi(\mu | x_1, ..., x_n)} \\ &\propto \pi(\sigma^2, \mu, x_1, ..., x_n) \\ &\Downarrow \\ \pi(\sigma^2 | x_1, ..., x_n, \mu) &\sim \mathcal{IG}\left(\alpha + \tfrac{n}{2}, \frac{1}{\frac{1}{\beta} + \frac{1}{2}\sum_{i=1}^{n}(x_i - \mu)^2} \right), \end{aligned}$$

which is the full conditionals.