NTNU
Norwegian University of
Science and Technology

An empirical study of the maximum pseudo-likelihood for discrete Markov random fields.

Johannes Fauske

Master of Science in Physics and Mathematics
Submission date: June 2009
Supervisor: Håkon Tjelmeland, MATH

Norwegian University of Science and Technology
Department of Mathematical Sciences

Problem Description

The candidate will empirically study the performance of the maximum pseudo-likelihood and maximum general pseudo-likelihood estimators for discrete Markov random fields when the amount of data increases.

Assignment given: 21. January 2009
Supervisor: Håkon Tjelmeland, MATH

Preface

This paper is the result of my master thesis in the subject TMA4905. I would like to thank Steinar Åsebø, Jens Helge Larsen, Steinar Nerhus and Håkon Tjelmeland for all the help I have received.

Johannes Fauske

Trondheim, June 24. 2009

Abstract

In this text we will look at two parameter estimation methods for Markov random fields on a lattice. They are maximum pseudo-likelihood estimation and maximum general pseudo-likelihood estimation, which we abbreviate MPLE and MGPLE. The idea behind them is that by maximizing an approximation of the likelihood function, we avoid computing cumbersome normalising constants. In MPLE we maximize the product of the conditional distributions for each variable given all the other variables. In MGPLE we use a compromise between pseudo-likelihood and the likelihood function as the approximation. We evaluate and compare the performance of MPLE and MGPLE on three different spatial models, which we have generated observations of. We are specially interested to see what happens with the quality of the estimates when the number of observations increases. The models we use are the Ising model, the extended Ising model and the Sisim model. All the random variables in the models have two possible states, black or white. For the Ising and extended Ising model we have one and three parameters respectively. For Sisim we have 13 parameters. The quality of both methods get better when the number of observations grow, and MGPLE gives better results than MPLE. However certain parameter combinations of the extended Ising model give worse results.

Contents

1 Introduction

Spatial analysis is a branch of statistics were the random variables also have geometric or geographic properties. It has applications to diverse subjects as seismology, meteorology and image analysis. The last subject mentioned is related to the topic of this paper. The spatial models we consider are binary Markov random fields (Hurn, Husby & Rue 2003) on a lattice, each site or pixel can be associated with a random variable. The variables have two possible states, either black or white.

If we want to estimate the model parameters with maximum likelihood estimation (MLE), we need to compute the normalizing constant. It can be obtained by summing over all possible configurations of the field. However, the total number of combinations will increase exponentially with the field's size, so even with only two states the computation soon becomes infeasible. Several methods have been developed to circumvent this problem. A recursive scheme to find the constant is presented by Reeves & Pettitt (2002), a weakness with the method is that the field can not be too large. The maximum likelihood estimate can also be obtained by using algorithms which involve Markov chain Monte Carlo (MCMC) simulation (Geyer & Thompson 1992). Besag (1974) proposes maximum pseudo-likelihood estimatation (MPLE), where we ignore the constant by using an approximation to the likelihood function. As long as the interaction between the pixels are weak MPLE is a good replacement to MLE (Geyer & Thompson 1992). In cases of strong interaction, MPLE can be used as initial value in a MCMC method. Huang & Ogata (2002) propose maximum generalized pseudo-likelihood estimation (MGPLE). In this method they consider blocks of the field, instead of only one pixel at the time as we do in MPLE. The normalizing constant is computed for each block and the method can be considered as a compromise between MLE and MPLE. The recursive method by Reeves & Pettitt (2002) can be used to compute the constant.

The purpose of this text is to empirically study the performance of the maximum pseudo likelihood and maximum pseudo block likelihood estimators for discrete Markov random fields when the number of observations increases. We will consider several models, and see if the results differ much for the two methods. In chapter 2 we establish our notation and explain some of the theory behind Markov random fields, pseudo- and general pseudo-likelihood. In chapter 3 we explain the spatial models we will use. In chapter 4 we give more details about the implementation. In chapter 5 we present our results and discuss them. At last we give some conclusions in our closing remarks.

2 Theory

2.1 Markov random fields

In this text our random fields are random variables on lattices with $N = M \times M$ pixels. The position of the sites is given by a set of indexes $\mathcal{I} = \{1, ..., N\}$. As shown in figure 1, they are ordered in each row from left to right, and the rows are ordered from top to bottom. Our random field is a set of random variables denoted as $\boldsymbol{x} = \{x_i : i \in \mathcal{I}\}$,

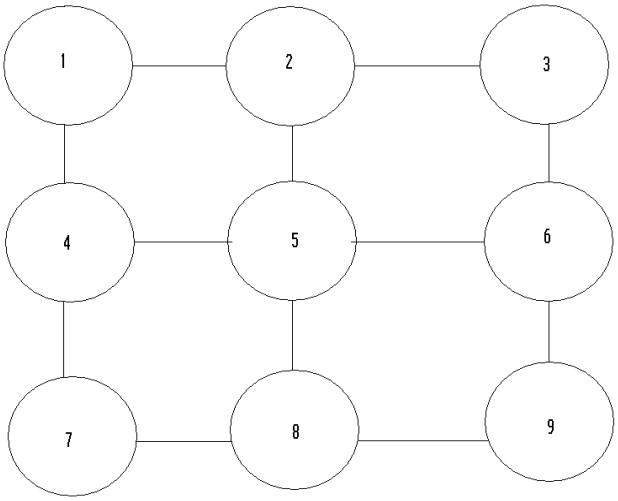


Figure 1: The ordering of indexes for a 3×3 lattice

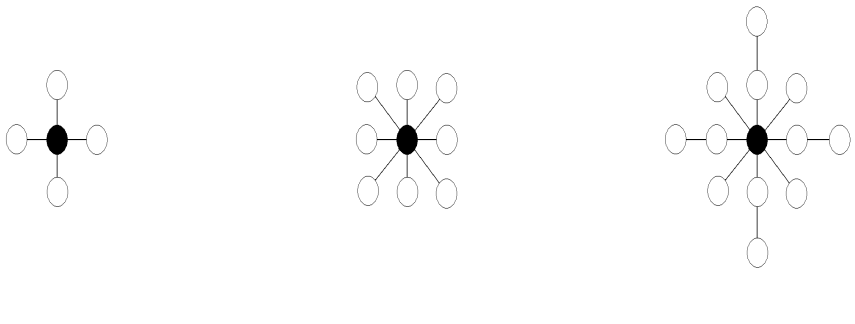
each x_i takes value in a finite set $\mathcal{X}$. The variables given by the subset $A \subseteq \mathcal{I}$ and its complement A^c are denoted as $\boldsymbol{x}_A = \{x_i : i \in A\}$ and $\boldsymbol{x}_{-A} = \{x_i : i \in \mathcal{I} \backslash A\}$, respectively.

We now define two important concepts, neighbourhoods and cliques. We say that each pixel i has a number of adjacent pixels which we call its neighbourhood, we denote this by $\partial_i \subseteq \mathcal{I} \backslash \{i\}$. A pixel may not be in its own neighbourhood, so the largest possible neighbourhood of i is given by the set of all indexes excluded itself. Which pixels we define to be in the neighbourhood differ, figure 2 shows three typical neighbourhood structures. If $j \in \partial_i$ we denote it as $i \sim j$. We require the neighbourhood relations to be symmetric, so $j \in \partial_i \Leftrightarrow i \in \partial_j$. The next concept ,the clique, is defined to be any subset of sites where every pair of these sites are neighbours. The shape and size of the cliques depend on the neighbourhood structure we choose. In figure 3(a) we have a lattice with neighbourhood structure from figure 2(a). With this structure only sites that are horizontally or vertically adjacent to each other can be neighbours. Thus the largest possible clique consists of two sites. If we choose the structure from figure 2(b),then sites that are diagonally adjacent also become neighbours. The largest possible clique in this case consist of four sites.

Our field is called a Markov random field (MRF) if the conditional distribution of a pixel given the rest, only depends on its neighbours,

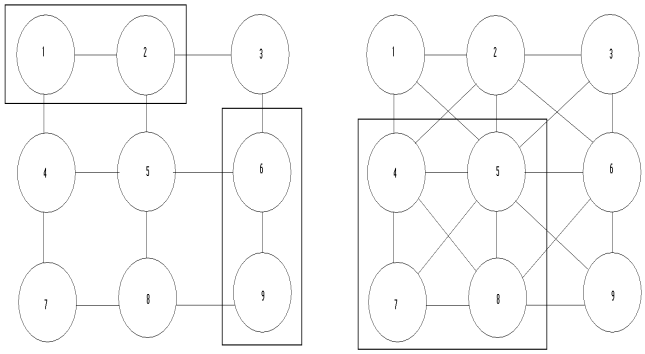
$$\pi(x_i|\boldsymbol{x}_{-i}) = \pi(x_i|\boldsymbol{x}_{\partial_i}), \forall i. \tag{1}$$

We also require that the field fulfil the positivity condition (Besag (1974)) $\pi(\boldsymbol{x}) > 0$, which means that all the $\mathcal{X}^N$ possible configurations for $\boldsymbol{x}$ may occur. Even if this condition and equation (1) are satisfied, it does not mean that it will be easy to use equation (1) to construct a joint density for $\pi(\boldsymbol{x})$. To solve this problem we use the Hammersley-Clifford



(a) (b) (c)

Figure 2: Three examples of common neighbourhood structures



(a) (b)

Figure 3: Examples of cliques with neighbourhood structure 2(a) and 2(b) respectively

theorem (Hurn et al. 2003). The theorem states that a distribution satisfying $\pi(\boldsymbol{x}) > 0$ for all configurations in $\mathcal{X}^{|\mathcal{I}|}$ is a Markov random field if and only if it has a joint density on the form

$$\pi(\boldsymbol{x}) = \frac{1}{Z}\exp\left\{-\sum_{C\in\mathcal{C}}\Phi_C(\boldsymbol{x}_C)\right\}. \tag{2}$$

The function $\Phi_C(\boldsymbol{x}_C)$, which is often called a potential function, describes the interaction between the random variables, and $\mathcal{C}$ is the set of all cliques. Z is the normalising constant

$$Z = \sum_{\boldsymbol{x}}\exp\left\{-\sum_{C\in\mathcal{C}}\Phi_C(\boldsymbol{x}_C)\right\}. \tag{3}$$

Essential for the random fields are their model parameters, the interaction between sites depend on them. The paramters can either be scalars or vectors. For the parameter estimation we will do in this paper, the conditional distribution in equation (1) is of great importance. If we use the joint density in equation (2), it can be expressed as

$$\pi(x_i|\boldsymbol{x}_{\partial_i}) = \frac{\pi(\boldsymbol{x})}{\pi(\boldsymbol{x}_{\partial_i})} \propto \exp\left\{-\sum_{C\in\mathcal{C}:i\in C}\Phi_C(\boldsymbol{x}_C)\right\}. \tag{4}$$

The only cliques we consider are the ones who contain x_i.

To illustrate Markov random fields we will use the Ising model (Grimmet (1987)) as an example. It was first presented as model for ferromagnetism where each variable has two possible states, either spin up or spin down, in our case the states are black or white. We define that a site has four neighbours, as in figure 2(a), and the cliques are the sets of horizontal and vertical neighbour pairs. The dependency between variables x_i and x_j is given by a parameter β. If it has a positive value, the variables tend to have the same colour as the majority of its neighbours, the opposite occurs if β is negative. The dependency between variables increases with the size of $|\beta|$. We use $\Phi_C(\boldsymbol{x}_C) = -\beta I(x_i = x_j)$ as potential function, then (2) can be written as

$$\pi(\boldsymbol{x}) = \frac{1}{Z(\beta)}\exp\left\{\beta\sum_{i\sim j} I(x_i = x_j)\right\}, \tag{5}$$

where the sum consists of all possible neighbour pairs, and each pair contributes to the sum only once. If β is unknown and we have an observation $\boldsymbol{x}$ of the Ising model, a maximum likelihood estimate for the parameter can be found by maximizing the log likelihood function,

$$l(\beta;\boldsymbol{x}) = \beta\sum_{i\sim j} I(x_i = x_j) - \log Z(\beta). \tag{6}$$

The normalising constant

$$Z(\beta) = \sum_{\boldsymbol{x}}\exp\left\{\beta\sum_{i\sim j} I(x_i = x_j)\right\} \tag{7}$$

is obtained by summing over all possible configurations for $\boldsymbol{x}$. With an $M \times M$ lattice and two possible states the total number of configurations are $2^{M \times M}$. Even with a small 4×4 lattice, the sum consists of 65536 terms. In this paper we use 64×64 grids, so its obvious that a brute force approach is not feasible.

2.2 Likelihood approximation

2.2.1 Pseudo-likelihood

In this paper we have stated several times that estimating a parameter from the likelihood function is often a dead end because of the normalising constant. A different approach to the problem is needed. Instead of the likelihood function, we use approximations were we avoid computationally expensive constants. Besag (1974) proposes the pseudo-likelihood, which is the product of conditional distributions for each variable given all the other variables. Since we have a Markov random field a variable only depends on its neighbours. If we have an observation $\boldsymbol{x}$ and parameter θ the log pseudo likelihood is

$$l_p(\beta; \boldsymbol{x}) = \sum_{i=1}^{N} \log \pi(x_i | \boldsymbol{x}_{\partial_i}, \beta). \tag{8}$$

By maximizing the expression above with respect to θ we get the maximum pseudo likelihood estimate. This is done by numerical optimization, the method will be explained in chapter 4.1.

For the Ising model each site has four neighbours and the conditional distribution on the form of equation (4) is

$$\pi(x_i | \boldsymbol{x}_{\partial_i}) \propto \exp\left\{ \beta \sum_{i \sim j} I(x_i = x_j) \right\}. \tag{9}$$

The expression above is easy to normalize, x_i has only two possible states so it becomes

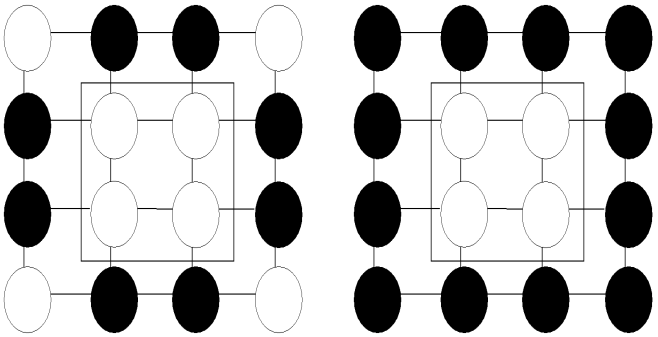
$$\pi(x_i | \boldsymbol{x}_{\partial_i}) = \frac{\exp\left\{\beta \sum_{i \sim j} I(x_i = x_j)\right\}}{\exp\left\{\beta \sum_{i \sim j} I(x_i = x_j)\right\} + \exp\left\{\beta \sum_{i \sim j} I(x_i \neq x_j)\right\}}. \tag{10}$$

Then equation (8) can be written as

$$l_p(\beta; \boldsymbol{x}) = \sum_{i=1}^{N} \log \frac{\exp\left\{\beta \sum_{i \sim j} I(x_i = x_j)\right\}}{\exp\left\{\beta \sum_{i \sim j} I(x_i = x_j)\right\} + \exp\left\{\beta \sum_{i \sim j} I(x_i \neq x_j)\right\}}. \tag{11}$$

2.2.2 Generalized pseudo-likelihood

With the pseudo-likelihood we considered one pixel at the time and normalized its conditional distribution. Huang & Ogata (2002) proposes a natural extension to this, the generalized pseudo-likelihood. Instead of considering one site at a time we look at larger sets



(a) (b)

Figure 4: The sites $\partial_{g(i)}$ are colored black, in 4(a) we have used neighbour structure from figure 2(a), and in 4(b) we have used the structure from 2(b)

or blocks of sites. For each site i we define a group of adjacent sites denoted as $g(i)$. The random variables given by the site $g(i)$ and its complement, are $\boldsymbol{x}_{g(i)} = \{x_k : k \in g(i)\}$ and $\boldsymbol{x}_{-g(i)} = \{x_k : k \notin g(i)\}$. For $g(i)$ we may choose any size or shape we want, and they do not have to be the same for all sites, but we have to consider the cost of finding the normalising constant and the simplicity of the implementation. In this paper we have chosen the blocks to be quadratic for all sites i. As in pseudo-likelihood we take advantage of that we are dealing with Markov random fields, we define a set

$$\partial_{g(i)} = \bigcup_k \partial_k \backslash g(i), \tag{12}$$

where $k \in g(i)$. The set consists of the sites adjacent to $g(i)$. Which sites we consider adjacent are decided by the neighbourhood structure we choose. In figure 4 we have shown $\partial_{g(i)}$ for two different neighbourhood structures. Then the approximation of interest, the product of conditional distributions for each $\boldsymbol{x}_{g(i)}$ given the rest, becomes

$$L_g(\theta; \boldsymbol{x}) = \prod_{i=1}^{N} \pi(\boldsymbol{x}_{g(i)} | \boldsymbol{x}_{\partial_{g_i}}, \theta). \tag{13}$$

If we maximize it with $g(i) = \{i\}$ we get the maximum pseudo-likelihood estimate, if we use that $g(i)$ consists of all sites in $\mathcal{I}$ we get the maximum likelihood estimate.

Again we will use the Ising model as an example. For the pseudo-likelihood we only had to consider four neighbour pairs for each pixel i, now we have to look at all pairs in $g(i) \cup \partial_{g(i)}$. For $\boldsymbol{x}_{g(i)} = \{x_k : k \in g(i)\}$ and $\boldsymbol{x}_{\partial_{g(i)}} = \left\{x_l : l \in \partial_{g(i)}\right\}$ we define the statistic

$$G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g}) = \sum_{k \sim j} I(x_k = x_j) + \sum_{k \sim l} I(x_k = x_l), \tag{14}$$

where $j \in g(i)$ and $l \in \partial_{g(i)}$. The neighbour pairs enters the sum only once. For a set of sites $g(i)$ we get the conditional distribution

$$\pi(\boldsymbol{x}_{g(i)}|\boldsymbol{x}_{\partial_{g(i)}}) = \frac{\exp\left\{\beta G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g})\right\}}{\sum_{\boldsymbol{x}_g} \exp\left\{\beta G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g})\right\}}. \tag{15}$$

In the denominator we have summed all possible configurations of x_g to get the normalising constant. The log GPL is then given by

$$l_g(\beta; \boldsymbol{x}) = \sum_{i=1}^{N} \log \frac{\exp\left\{\beta G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g})\right\}}{\sum_{\boldsymbol{x}_g} \exp\left\{\beta G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g})\right\}}. \tag{16}$$

Even if our blocks are small compared to the full lattice, a straightforward computation of the denominator will be both inefficient, and most often infeasible. In the next chapter we will look at an more effective way to compute the normalising constant.

3 Spatial Models

In the introduction we promised we would look at parameter estimation for several models, one of them is of course the Ising model. Since we have used it as an example on several occasions, we concentrate on the other models in this chapter. The models we consider are binary Markov random fields were the variables have two states, black or white. We also use toroidal boundary conditions, which means that for an Ising model with $N = M \times M$ sites, the neighbourhood of site M is $\partial_M = \{M-1, 1, 2M, N\}$. It is the same principle for the other models, but the neighbourhood structure may vary.

3.1 Extended Ising model

This model is similar to the Ising model in the way that it uses the same neighbourhood structure. The difference is that we use three parameters instead of just one. We have a new abundance parameter α and we split the β parameter into β_h and β_v which influences the horizontal and vertical neighbour relations, respectively. The joint density for the model can be written as

$$\pi(\boldsymbol{x}) = \frac{1}{Z(\beta)} \exp\left\{\sum_{i=1}^{N} \alpha x_i + \beta_h \sum_{i\sim j} I(x_i = x_j) + \beta_v \sum_{i\sim l} I(x_i = x_l)\right\}, \tag{17}$$

where $i \sim j$ and $i \sim l$ in the equation above are all the horizontal and vertical neighbour pairs, respectively. The normalising constant is

$$Z(\beta) = \sum_{\boldsymbol{x}} \exp\left\{\sum_{i=1}^{N} \alpha x_i + \beta_h \sum_{i\sim j} I(x_i = x_j) + \beta_v \sum_{i\sim l} I(x_i = x_l)\right\}. \tag{18}$$

If we want the pseudo-likelihood we need the normalized conditional distribution on the form of (4),

$$\pi(x_i|\boldsymbol{x}_{\partial_i}) = \frac{\exp\left\{\alpha x_i + \beta_h \sum_{i\sim j} I(x_i = x_j) + \beta_v \sum_{i\sim l} I(x_i = x_l)\right\}}{\sum_y \exp\left\{\alpha y + \beta_h \sum_{i\sim j} I(y = x_j) + \beta_v \sum_{i\sim l} I(y = x_l)\right\}}, \tag{19}$$

where the sums over $i \sim j$ and $i \sim l$ each consist of two terms. In the denominator we sum over the two possible states for x_i. If we use vector notation $\phi = (\alpha, \beta_h, \beta_v)^T$, the log pseudo-likelihood becomes

$$l_p(\phi; \boldsymbol{x}) = \sum_{i=1}^{N} \log \frac{\exp\left\{\alpha x_i + \beta_h \sum_{i\sim j} I(x_i = x_j) + \beta_v \sum_{i\sim l} I(x_i = x_l)\right\}}{\sum_y \exp\left\{\alpha y + \beta_h \sum_{i\sim j} I(y = x_j) + \beta_v \sum_{i\sim l} I(y = x_l)\right\}}. \tag{20}$$

If we want the general pseudo-likelihood, we define the the sets $g(i)$ and $\partial_{g(i)}$ in the same way as before. The sites given by $g(i)$ are quadratic blocks of the lattice. As before we define the sets of random variables $\boldsymbol{x}_{g(i)} = \{x_k : k \in g(i)\}$ and $\boldsymbol{x}_{\partial_{g(i)}} = \{x_l : l \in \partial_{g(i)}\}$. We denote the vertical and horizontal neighbour pairs as $k \sim v$ and $k \sim h$ respectively, then we define the statistic,

$$G_i(\boldsymbol{x}_{g(i)}, \boldsymbol{x}_{\partial_{g(i)}}; \phi) = \alpha \sum_k x_k, \tag{21}$$

$$+ \beta_h \left(\sum_{k\sim h} I(x_k = x_{h_j}) + \sum_{k\sim h} I(x_k = x_{h_l}) \right) \tag{22}$$

$$+ \beta_v \left(\sum_{k\sim v} I(x_k = x_{v_j}) + \sum_{k\sim v} I(x_k = x_{v_l}) \right) \tag{23}$$

were $h_j, v_j \in g(i)$ and $h_l, v_l \in \partial_{g(i)}$. The expression we want to maximize then becomes

$$l_g(\phi; \boldsymbol{x}) = \sum_{i=1}^{N} \log \frac{\exp\left\{G_i(\boldsymbol{x}_{g(i)}, \boldsymbol{x}_{\partial_{g(i)}}; \phi)\right\}}{\sum_{\boldsymbol{x}_{g(i)}} \exp\left\{G_i(\boldsymbol{x}_{g(i)}, \boldsymbol{x}_{\partial_{g(i)}}; \phi)\right\}}. \tag{24}$$

In the denominator we sum over all possible configurations for the block given by $g(i)$.

3.2 Sisim, a geologically inspired model

For the Ising and extended Ising model we have two types of neighbour pairs, with one and three parameters respectively. Now we take a large step away from this, and introduce a model with 13 parameters, we use vector notation $\theta = (\theta_0, \theta_1, ..., \theta_{12})^T$. The parameters $\theta_1, \theta_2 ..., \theta_{12}$ correspond to 12 types of neighbour pairs, and θ_0 is an abundance parameter. In figure **??** we have shown all the types of neighbour pairs in a 3×3 clique.

This model is an attempt to recreate a given picture generated with sequential indicator simulation, we call it Sisim, so all the parameters have given values, contrary to

the Ising models were the parameter values may vary. The potential function for the model is $\Phi_C(\boldsymbol{x}_C) = \left(\theta_0 \sum_i x_i + \sum_{i\sim j_s} \theta_s x_i x_{j_s}\right)$, were $j \in \partial_i$, and the index s indicates which type of neighbour pair we have. If $s = 1$ we see in figure 5 that we have 6 possible neighbour pairs, if $s = 8$ or $s = 12$ we have only one pair. Note that we take the product of the adjacent variables, we do not use an indicator functions as we have done for the former models. The joint density for this model can then be expressed as

$$\pi(\boldsymbol{x}) = \frac{1}{Z(\theta)} \exp\left\{-\left(\sum_{i=1}^{N} \theta_0 x_i + \sum_{i\sim j_s} \theta_s x_i x_{j_s}\right)\right\}, \tag{25}$$

were $i \sim j_s$ consists of all the possible neighbour pairs. The normalising constant

$$Z(\theta) = \sum_{\boldsymbol{x}} \exp\left\{-\left(\sum_{i=1}^{N} \theta_0 x_i + \sum_{i\sim j_s} \theta_s x_i x_{j_s}\right)\right\}, \tag{26}$$

is found in the same manner as before. The normalized conditional distribution of x_i given the other variables is expressed as

$$\pi(x_i|\boldsymbol{x}_{\partial_i}) = \frac{\exp\left\{-\left(\theta_0 x_i + \sum_{i\sim j_s} \theta_s x_i x_{j_s}\right)\right\}}{\sum_y \exp\left\{-\left(\theta_0 y + \sum_{i\sim j_s} \theta_s y x_{j_s}\right)\right\}}, \tag{27}$$

since i has 24 neighbours, $i \sim j_s$ now consists of the same number of pairs. We use the expression above to get the log pseudo likelihood, which becomes

$$l_p(\theta; \boldsymbol{x}) = \sum_{i=1}^{N} \frac{\exp\left\{-\left(\theta_0 x_i + \sum_{i\sim j_s} \theta_s x_i x_{j_s}\right)\right\}}{\sum_y \exp\left\{-\left(\theta_0 y + \sum_{i\sim j_s} \theta_s y x_{j_s}\right)\right\}}. \tag{28}$$

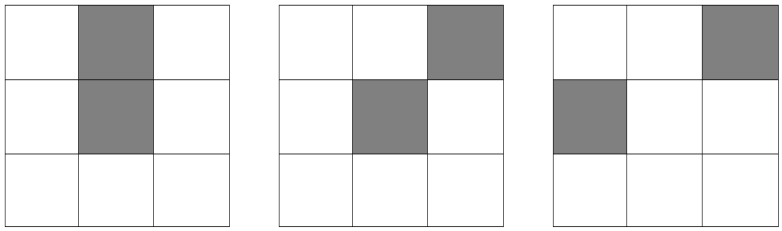
To get the general pseudo-likelihood, we define the the sets $g(i)$ and $\partial_{g(i)}$ in the same way as before. For the sets of random variables $\boldsymbol{x}_{g(i)} = \{x_k : k \in g(i)\}$ and $\boldsymbol{x}_{\partial_{g(i)}} = \left\{x_l : l \in \partial_{g(i)}\right\}$, we define the statistic

$$G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g}) = \sum_k x_k + \sum_{i\sim j_s} \theta_s x_i x_{j_s} + \sum_{i\sim l_s} \theta_s x_i x_{l_s}, \tag{29}$$

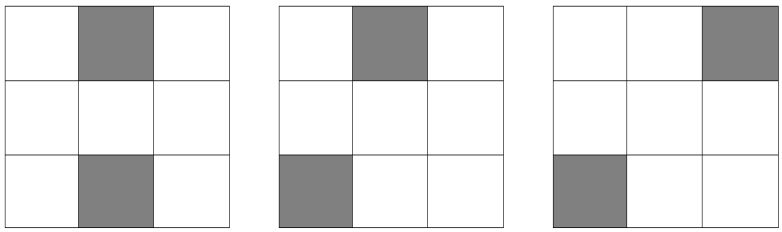
were $j_s \in g(i)$ and $l_s \in \partial_{g(i)}$. The general pseudo-likelihood then becomes

$$l_g(\theta; \boldsymbol{x}) = \sum_{i=1}^{N} \log \frac{\exp\left\{-G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g})\right\}}{\sum_{\boldsymbol{x}_g} \exp\left\{-G_i(\boldsymbol{x}_g, \boldsymbol{x}_{\partial_g})\right\}}. \tag{30}$$

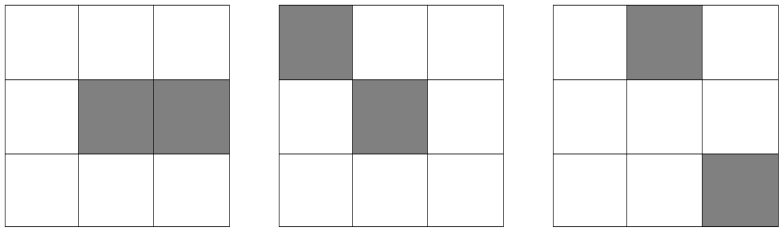
To get the estimated parameters we maximize equations (**??**) and (**??**) with respect to θ.



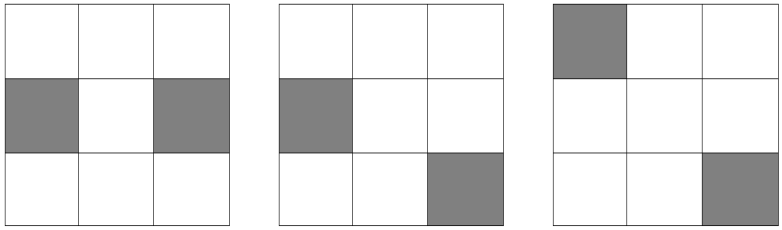
(a) θ_1 (b) θ_2 (c) θ_3



(d) θ_4 (e) θ_5 (f) θ_6



(g) θ_7 (h) θ_8 (i) θ_9



(j) θ_{10} (k) θ_{11} (l) θ_{12}

Figure 5: Examples of the 12 types of neighbour pairs for a 3×3 clique.

4 Practical issues with implementation

4.1 Optimization

To obtain an estimated parameter $\hat{\beta}$ with MPLE or MGPLE we need to solve the optimization problems,

$$\hat{\beta} = \arg\max_{\beta} l_p(\beta; \boldsymbol{x}) \tag{31}$$

$$\hat{\beta} = \arg\max_{\beta} l_g(\beta; \boldsymbol{x}), \tag{32}$$

respectively. There exists several robust and efficient numerical methods that solve optimization problems (Nocedal & Wright 2000). One can either implement one of them, or use bulit-in functions in mathematical software tools like `MATLAB` or `R`. We have done the latter, we have used the function `nlm` [1] in `R`, which stands for non-linear minimization and is a Newton-type algorithm. Since it is minimization method we use that

$$\hat{\beta} = \arg\max_{\beta} l(\beta; \boldsymbol{x}) = \arg\min_{\beta} \left\{ -l(\beta; \boldsymbol{x}) \right\}. \tag{33}$$

The Newton method is a line search algorithm, in each iteration we compute a search direction, and how far we move along it. The next iterate is given by

$$\beta_{k+1} = \beta_{k+1} + \epsilon_k p_k, \tag{34}$$

where the positive scaler ϵ_k is referred to as the step length, and p_k is the search direction. When $l(\beta_k; \boldsymbol{x})$ satisfies several conditions (Nocedal & Wright 2000), convergence is achieved. In the Newton method we compute the Hessian and gradient of $l(\beta_k; \boldsymbol{x})$, and search along the direction

$$p_k = -\nabla^2 l(\beta_k; \boldsymbol{x})^{-1} \nabla l(\beta_k; \boldsymbol{x}). \tag{35}$$

In `nlm` the Hessian and gradient are computed numerically.

4.2 Simulation of observations

The purpose of this text is to study the quality of parameter estimates when the number of observations of a field $\boldsymbol{x}$ increases. Since we do not have any observations, we simulate them with Gibbs' sampling (Hurn et al. 2003) and a method called Coupling from the past (Propp & Wilson 1996). In figure 6 we have generated some realizations of the Ising model.

Gibbs' sampling is a MCMC method that sequentially updates the state of each variable x_i from its full conditional distribution $\pi(x_i|\boldsymbol{x}_{-i})$, for the Ising model this distribution is equation (10), for the extended model in chapter 3.1 it is equation (19), and for the Sisim model it is equation (27). The initial state of the field before the sampling

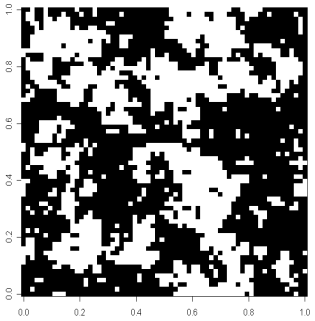
[1] `http://sekhon.berkeley.edu/stats/html/nlm.html`

(a) $\beta = 0.4$

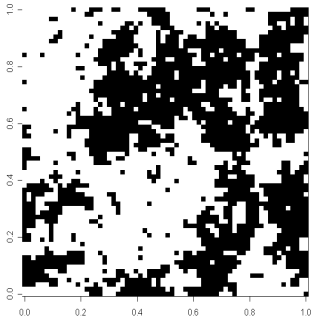
(b) $\beta = 0.4$

(c) $\beta = 0.6$

(d) $\beta = 0.6$



(e) $\beta = 0.8$



(f) $\beta = 0.8$

Figure 6: Realizations of the Ising model with different values for β. If β increases, variables tend to have the same colour as the majority of its neighbours, so we get larger areas of one colour.

starts can be, all variables have the same colour, or the state of each variable is randomly chosen to be either black or white. The sequence of the sites we visit may be in numerical order or random. We choose to update the variables in a random order. If we have a random site j, the probability of x_j being black is $p = \pi(x_j = \text{black}|\boldsymbol{x}_{-j})$. We draw a number u uniformly distributed between 0 and 1. If u is smaller or equal to p, we update x_j to be black, if u is larger x_j becomes white. To achieve convergence, this has to be done sufficiently many times. However the Ising model has a property called phase transition behaviour (Hurn et al. 2003), this means that the Gibbs' sampler performs poorly for values of β close to or larger than a critical value $\beta^* = log(1 + \sqrt{2}) = 0.881373$. We explained earlier that for large values of β, variables tend to have the same colour as the majority of its neighbours. When we update only one variable at the time and the probability for change of state is small, we experience problems with poor mixing and thus slow convergence.

Fortunately there are alternative methods that we can use. One is the Swendsen-Wang algorithm (Swendsen & Wang 1987) which has much better convergence properties for the Ising model. Coupling from the past (CFTP) is a method that give perfect realisations from the distribution of interest, so we do not have to think about convergence. However if we consider CPU time, this method also performs poorly when β is close to the critical value. For large values of β we first use CFTP to generate a realisation with a lower value β_l. Then we use this realisation as initial state in the Gibbs' sampler, thus we get much faster convergence. For $\beta < 0.70$ we use CFTP, whereas for larger values we use Gibbs' sampler with a CFTP generated realisation with $\beta = 0.7$ as initial state. For the Sisim model we do not use CFTP, only the Gibb's sampler.

4.3 Recursions for computing the normalising constant

In this paper we have discussed the issues with normalising constants and the limitations it gives us. Reeves & Pettitt (2002) propose a recursive scheme to find this constant. The method is closely related to the forward recursions in the forward-backward algorithm (Scott 2002) for hidden Markov chains. This method is by far better than a brute force approach. However, as we mentioned in the introduction, the random field can not be too large, so we will not use this method to compute the constant for our model on a 64×64 lattice, but it is perfect for the smaller blocks in the general pseudo-likelihood in chapter 2.2.2. First we discuss the recursions for a general model, then we give the specific recursions for each model.

4.4 Recursion for a general factorisable model

If a discrete valued vector $x_{1:n} = (x_1, x_2..., x_n)^T$ has unnormalized probability distribution $q(x_{1:n})$, and we have a valid factorization on the form

$$q(x_{1:n}) = q_1(x_1, x_2, .., x_{r+1})q_2(x_2, x_3, .., x_{r+2})...q_d(x_d, x_{d+1}, ..x_n), \tag{36}$$

where $r < n$ and $d = n - r$, we call this a lag-r model. The normalising constant is given by

$$Z = \sum_{x_{1:n}} q(x_{1:n}) = \sum_{x_{k+1:n}} \sum_{x_k} q_k(x_{k:n}) \sum_{x_{k-1}} q_{k-1}(x_{k-1:n-1}) \sum_{x_1} q_1(x_{1:r+1}). \tag{37}$$

This constant is computed in the following recursions,

$$Q_1(x_{2:r+1}) = \sum_{x_1} q_1(x_{1:r+1}) \tag{38}$$

$$Q_t(x_{t+1:t+r}) = \sum_{x_k} q_t(x_{t:t+r}) Q_{t-1}(x_{t:r+t-1}) \ , \ (t = 2, ...d) \tag{39}$$

$$Z = \sum_{x_{d+1:n}} Q_k(x_{d+1:n}) \tag{40}$$

We will only use the recursions above for the Ising model. When we the general pseudo-likelihood for the Sisim model, we compute the normalising constants with the brute force approach. This means that the blocks must be small.

4.5 Recursions for an Ising model

In this paper our blocks are quadratic, so a set of sites $\boldsymbol{x}_{g(i)} = \{x_k : k \in \{1, 2, ..n\}\}$ consist of $n = m \times m$ sites. We have from equation (15) the unormalized distribution

$$q(\boldsymbol{x}_{g_{(i)}} | \boldsymbol{x}_{\partial_{g(i)}}) = \exp\left\{\beta G_i(\boldsymbol{x}_{g(i)}, \boldsymbol{x}_{\partial_{g(i)}})\right\}. \tag{41}$$

In the spirit of the notation from the prevouis chapter we write it as

$$q(x_{1:n}) = \exp\left\{\beta\left(\sum_{k\sim j} I(x_k = x_j) + \sum_{k\sim l} I(x_k = x_l)\right)\right\}, \tag{42}$$

where $j \in g(i)$ and $l \in \partial_{g(i)}$, each neighbour pair enters the sum only once. Before we can use the recursions on $q(x_{1:n})$, we need to factorize it. This can be done in many ways, the only requirement is that it fulfils equation (36). We decide to factorize it in the following way. If we have a $m \times m$ lattice, which is a smaller block of a larger $M \times M$ lattice, $q(x_{1:n})$ may be written as the product of $d = m^2 - m$ factors. The first factor $q_1(x_{1:m+1})$, consists of all whitin-row interactions in the first row, the interaction between variables x_1 and x_{m+1}, and for $x_e : e \in \{1, 2.., m + 1\}$ we have to look at the interaction with sites in ∂_{g_i}. The other $d - 1$ factors $q_t(x_{t:t+m})$ consists of one whitin-row interaction and one between-row interaction. If x_{t+m} are on the boundary we also need the contribution from ∂_{g_i}. In figure 7 we have an example with a 4×4 lattice. For a lattice of arbitrary size the factors can be written as,

$$q_1(x_{1:m+1}) = \exp\left\{\beta\left(I(x_{m+1} = x_1) + \sum_{e=2}^{m} I(x_e = x_{e-1}) + \sum_{e=1}^{m+1} \sum_{e\sim l} I(x_e = x_l)\right)\right\},$$

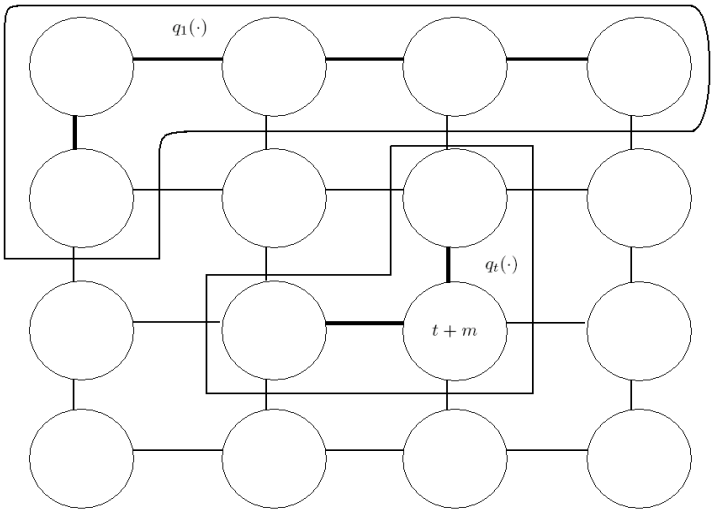


Figure 7: The factorization of $q(x_{1:n})$ on a 4×4 lattice. The interactions within the factors are the bold edges between the sites, if $t+m$ is on the boundary, we also have contributions from $\partial_{g(i)}$.

$$q_t(x_{t:t+m}) = \exp\left\{\beta\left(I(x_{t+m} = x_{t+m-1}) + I(x_{t+m} = x_t) + \sum_{(t+m)\sim l} I(x_{t+m} = x_l)\right)\right\}$$

for $t = 2, ..., n$. We use this factorization in the recursions from the previous chapter to compute the normalising constant for equation (15).

For the extended model from chapter 3.1 we have to separate between the vertical and horizontal neighbour pairs. For a set $\boldsymbol{x}_{g(i)} = \{x_k : k \in \{1, 2, ..n\}\}$ we denote the horizontal pairs as $k \sim h$ and the vertical ones as $k \sim v$. From (24) we get the distribution

$$q(x_{1:n}) = \exp\{\alpha \sum_{k=1}^{n} x_k + \beta_h \sum_{k\sim h} \big(I(x_k = x_{h_j}) + I(x_k = x_{h_l})\big) \tag{43}$$

$$+ \beta_v \sum_{k\sim v} \big(I(x_k = x_{v_j}) + I(x_k = x_{v_l})\big)\}, \tag{44}$$

where $v_j, h_j \in g(i)$ and $v_l, h_l \in \partial_{g(i)}$. We use the same factorization as above, and we get

$$q_1(x_{1:m+1}) = \exp\{\alpha \sum_{e=1}^{m+1} x_e + \beta_v I(x_{m+1} = x_1) + \beta_h \sum_{e=2}^{m} I(x_e = x_{e-1})$$

$$+ \sum_{e=1}^{m+1} (\beta_h \sum_{e\sim h} I(x_e = x_{h_l}) + \beta_v \sum_{e\sim v} I(x_e = x_{v_l}))\},$$

$$q_t(x_{t:t+m}) = \exp\{\alpha x_{t+m} + \beta_h I(x_{t+m} = x_{t+m-1}) + \beta_v I(x_{t+m} = x_t) + \sum_{(t+m)\sim h} \beta_h I(x_{t+m} = x_{h_l}) + \sum_{(t+m)\sim v} \beta_v I(x_{t+m} = x_{v_l}))\}$$

for $t = 2, ..., n$.

5 Parameter estimation and discussion

In the previous chapters we have established the methods maximum pseudo-likelihood and maximum general pseudo-likelihood estimation, which we abbreviate MPLE and MGPLE respectively. For the latter we use blocks of size $|g(i)| = m \times m$. When we want to specify this, we use the abbreviation $m \times m$ MGPLE. We have also explained some of the theory behind the methods, and dealt with the practical issues of computing constants, optimization, and how to generate realisations. The foundation has been constructed, and it is time to reap the benefit of our work. We want to compare the performance of the methods when we take into consideration the amount of data, the value of the true parameter and which model we are estimating from. For the MGPLE it is also interesting to see how it performs when we increase the size of the blocks. In this chapter we give several examples for each model, and we try to answer these questions.

5.1 Ising model

We simulate observations $\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^K$ with the methods we explained in chapter 4.2. The maximum number of realisations are $K = 300$, and that is also the case for the other models. The realisations are generated with the true parameter values $\beta = \{0.40, 0.60, 0.80, 0.82, 0.83\}$. We use the expressions in equation (11) and (16) to get the log pseudo- and general log pseudo-likelihood,

$$l_p(\beta; \boldsymbol{x}^1, .., \boldsymbol{x}^K) = \sum_{k=1}^{K} \log\, l_p(\beta; \boldsymbol{x}^k) \tag{45}$$

$$l_g(\beta; \boldsymbol{x}^1, .., \boldsymbol{x}^K) = \sum_{k=1}^{K} \log\, l_g(\beta; \boldsymbol{x}^k), \tag{46}$$

respectively. To get the estimated parameters from the likelihoods above, we solve the optimization problems

$$\hat{\beta} = \arg\max_{\beta} l_p(\beta; \boldsymbol{x}^1, .., \boldsymbol{x}^K) \tag{47}$$

$$\hat{\beta} = \arg\max_{\beta} l_g(\beta; \boldsymbol{x}^1, .., \boldsymbol{x}^K), \tag{48}$$

according to the routine outlined in chapter 4.1. For the MGPLE we use blocks of size 2×2 and 3×3.

If we want to evaluate the methods, we repeat the optimization in equation (47) and (48) n times with new observations generated in each run. There are several statistics we can use to examine the quality of the estimation. One is the bias of an estimator. If we have n estimates of a parameter β, in our case $n = 50$, we define it as

$$\text{bias}(\hat{\beta}) = \frac{1}{n}\sum_{i=1}^{n}\hat{\beta}_i - \beta_0, \tag{49}$$

where β_0 is the true parameter and $\hat{\beta}_i$ is one of the n estimates. The bias measures the deviation, both positive and negative, from the true parameter. We prefer the bias to be as small as possible. In figure 8 we show the bias of MPLE and MGPLE plotted against the number of realisations. In this figure, and all of its kind, we use log scale on the x-axis to make the figure easier to interpret. We see that for both methods the bias approaches 0 when the number of observations increase. However, the biases fluctuate, especially for $k < 100$, and they are not giving us any strong indications of which method performs the best.

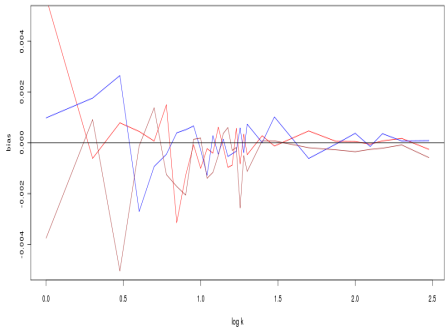
The estimated standard deviation

$$s(\hat{\beta}) = \sqrt{\frac{1}{n-1}\sum_{i=1}^{n}\left(\hat{\beta}_i - \frac{1}{n}\sum_{i=1}^{n}\hat{\beta}_i\right)}, \tag{50}$$

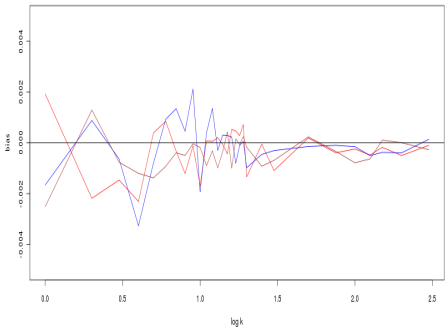
also contains information about how the methods perform. If it is small we take it as a good sign. We will use notation $s_p(\hat{\beta})$ and $s_m(\hat{\beta})$ when we want to separate between the standard deviations for MPLE and $m \times m$ MGPLE, respectively. In figure 9 we see that the standard deviations decrease when the number of observations grow. Now it is easier to separate between the methods. We see most of the time that $s_2(\hat{\beta})$ and $s_3(\hat{\beta})$ is smaller than $s_p(\hat{\beta})$, and in turn $s_2(\hat{\beta})$ is smaller than $s_3(\hat{\beta})$. For $k > 200$ we the standard deviations approach a stable level, and in most cases with $s_p(\hat{\beta})$ slightly higher than the other two.

It is also interesting to look at the scattering of the n estimates, a confidence interval can give us a notion of how these values vary. In figure 10 we have plotted the lower and upper quantile of a 0.95 confidence interval for $\frac{1}{n}\sum_{i=1}^{n}\hat{\beta}_i$. The intervals for MGPLE are smaller than the ones for MPLE, and most of the time the 3×3 block again gives slightly better result than the smaller block. Again we observe that the quality of the estimates get better for both methods when the number of observations increase.

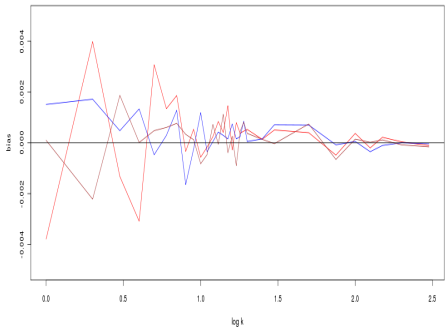
Figures 8 - 10 show us that the estimates get better when the amount of data increase. When the numbers of observations exceed 100, there is little differences between the methods. It is when k is small, we observe the largest difference. Both 3×3 and 2×2 MGPLE give better result than MPLE. Concerning the size of the blocks, 3×3 gives better result, but the difference is not dramatic. We also see that the choice of true parameter β has little to say for the success of our estimation.



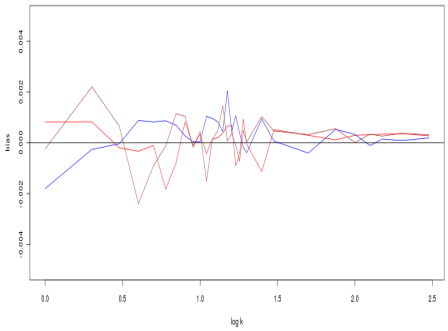
(a) $\beta = 0.4$



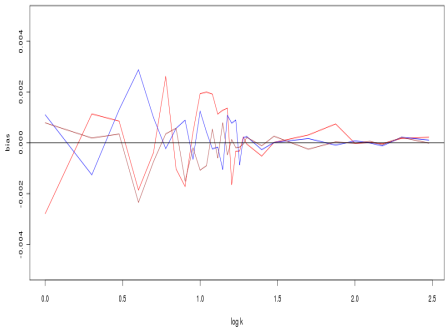
(b) $\beta = 0.6$



(c) $\beta = 0.8$

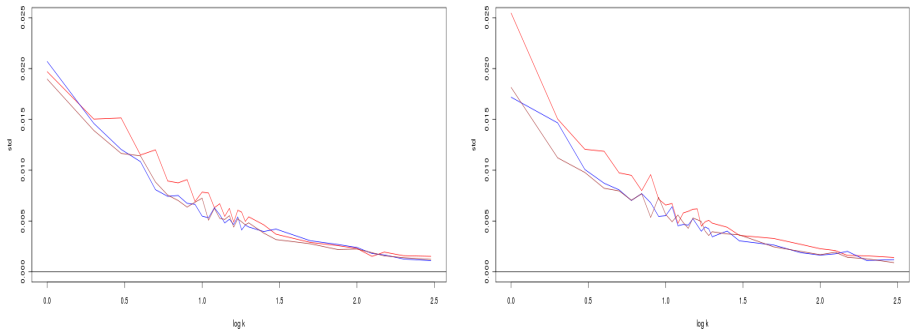


(d) $\beta = 0.82$



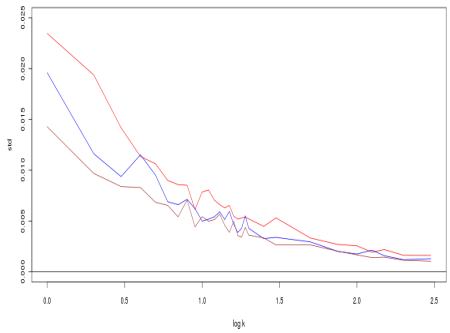
(e) $\beta = 0.83$

Figure 8: The red, blue and brown lines are the biases when we use MPLE, 2×2 MGPLE, and 3×3 MGPLE respectively.

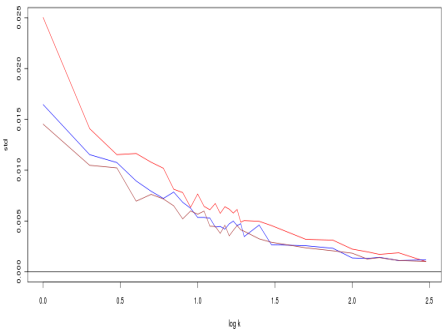


(a) $\beta = 0.4$

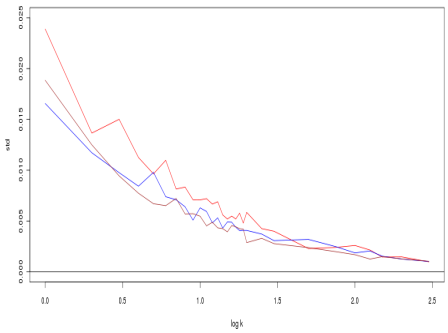
(b) $\beta = 0.6$



(c) $\beta = 0.8$

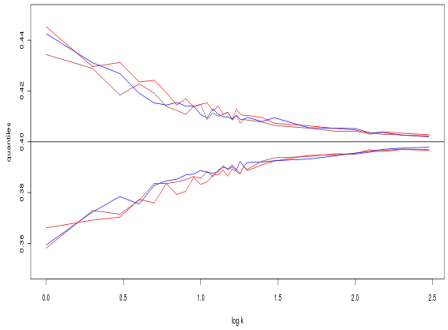


(d) $\beta = 0.82$

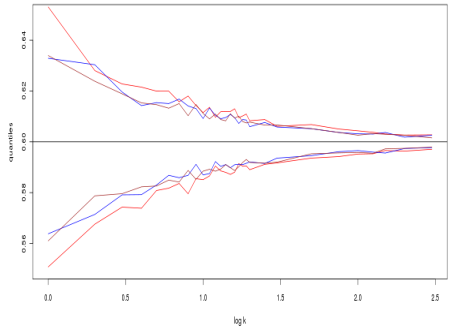


(e) $\beta = 0.83$

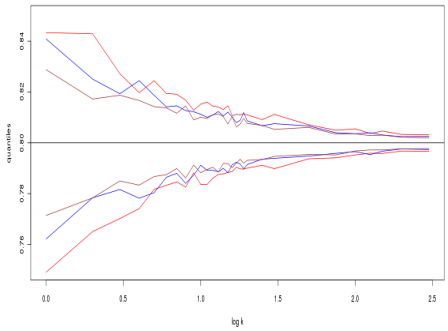
Figure 9: The red, blue and brown lines are the standard deviations when we use MPLE, 2×2 MGPLE, and 3×3 MGPLE respectively.



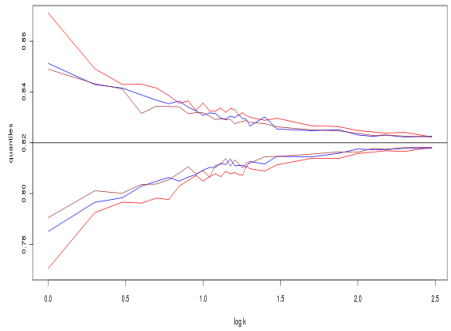
(a) $\beta = 0.4$



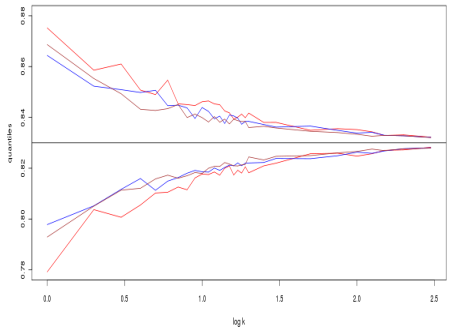
(b) $\beta = 0.6$



(c) $\beta = 0.8$



(d) $\beta = 0.82$



(e) $\beta = 0.83$

Figure 10: The lines are the upper and lower quantiles of a 0.95 confidence interval, and the black line is the value of the true parameter. The red, blue and brown lines are the quantiles when we use MPLE, 2×2 MGPLE, and 3×3 MGPLE respectively.

5.2 Extended Ising model

In the previous chapter we only had one parameter to consider, and we got good results for both methods. Now the realisations $\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^K$ depend on three parameters, and we want to see how the methods cope with the increased complexity. To get the log pseudo- and general log pseudo-likelihood, we use equations (19) and (24), and with $\phi = (\alpha, \beta_h, \beta_v)^T$ we get

$$l_p(\phi; \boldsymbol{x}^1, .., \boldsymbol{x}^K) = \sum_{k=1}^{K} \log\, l_p(\phi; \boldsymbol{x}^k) \tag{51}$$

$$l_g(\phi; \boldsymbol{x}^1, .., \boldsymbol{x}^K) = \sum_{k=1}^{K} \log\, l_g(\phi; \boldsymbol{x}^k). \tag{52}$$

To get the estimated parameters $\hat{\alpha}, \hat{\beta}_h$ and $\hat{\beta}_v$, we solve the optimization problems

$$\hat{\phi} = \arg\max_{\phi} l_p(\phi; \boldsymbol{x}^1, .., \boldsymbol{x}^K) \tag{53}$$

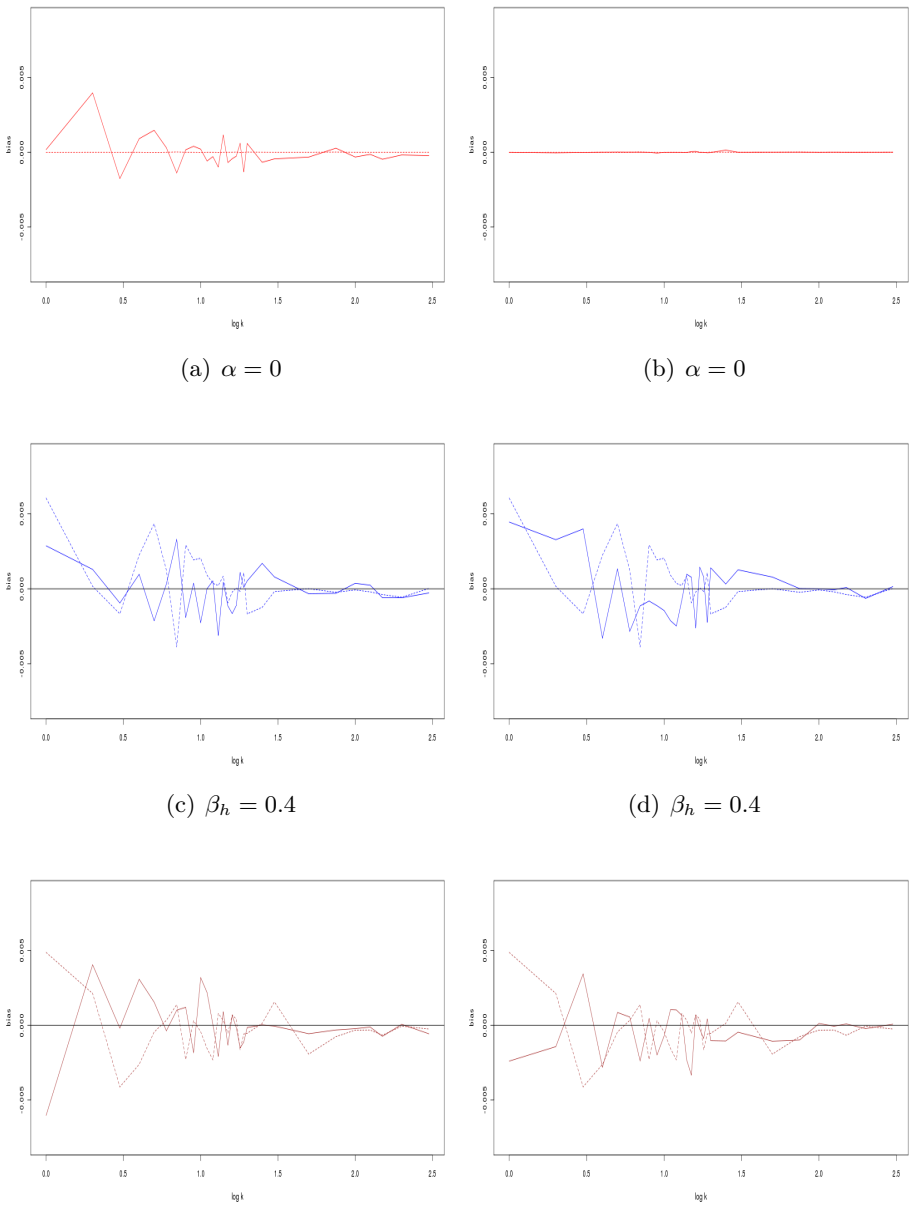
$$\hat{\phi} = \arg\max_{\phi} l_g(\phi; \boldsymbol{x}^1, .., \boldsymbol{x}^K). \tag{54}$$

in the same way as before.

For this model we have many possible combinations of the three parameters. First we try $\alpha = 0$ and $\beta = \beta_h = \beta_v$, and we generate realizations with $\beta = \{0.40, 0.80, 0.82\}$. We want to see if the methods recognize that they are estimating from an Ising model. To evaluate the performance of MPLE, 2×2 MGPLE and 3×3 MGPLE, we apply the same statistics as we did in the previous chapter. In figures 11 - 13 we see that that we do not have any bias when $\hat{\alpha}$ is found with MGPLE, wheras we do have bias when we use MPLE. Though we also see that for MPLE the bias of $\hat{\alpha}$ smaller than the bias of $\hat{\beta}_h$ and $\hat{\beta}_v$, so it also recognizes the Ising model. However it is not as spot on as MGPLE. Again we note that the biases fluctuate before they approach 0.

When we look at the standard deviations in figure 14, we again draw the conclusion that both methods recognizes the Ising model, and MGPLE is by far the better method. The standard deviation of $\hat{\alpha}$ completely hugs the zero line. Though MPLE does not give as small standard deviations for $\hat{\alpha}$, we see that it is smaller than $s(\hat{\beta}_h)$ and $s(\hat{\beta}_v)$. In figure 15 - 17 we see for 2×2 and 3×3 MGPLE that the confidence interval for $\frac{1}{n}\sum_{i=1}^{n} \hat{\alpha}_i$ completely collapses. This is not the case for MPLE, but we see that the interval is smaller than for the other parameters. Which leads us over to the question of how the methods compares when it comes to estimating the other parameters. The difference between the methods are not large. In figure 14 we see that $s_2(\hat{\beta}_h)$ and $s_2(\hat{\beta}_v)$ are most of the time smaller than $s_p(\hat{\beta}_h)$ and $s_p(\hat{\beta}_v)$, however the difference is not large. For 2×2 and 3×3 MGPLE we see that the the largest block gives the smallest standard deviation most of the time. By studying figures 15 - 17 we see that the 3×3 block gives the narrowest intervals.

Now it is time to move away from the Ising model, and we estimate from realisations generated with $\alpha \neq 0$ and $\beta_h \neq \beta_v$. First we use true values $\beta_h = 0.83, \beta_v = 0.80$

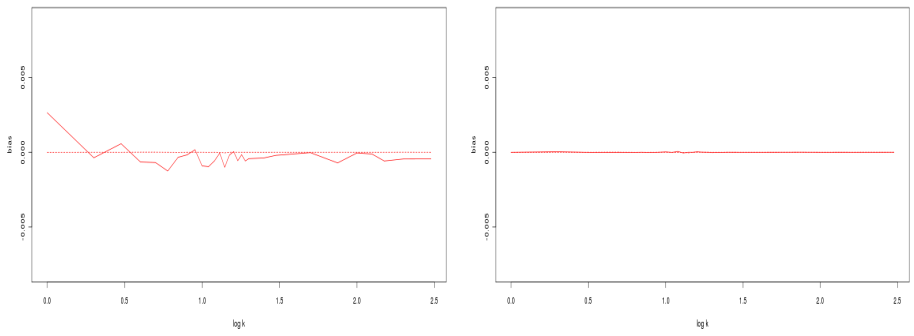


(a) $\alpha = 0$ (b) $\alpha = 0$

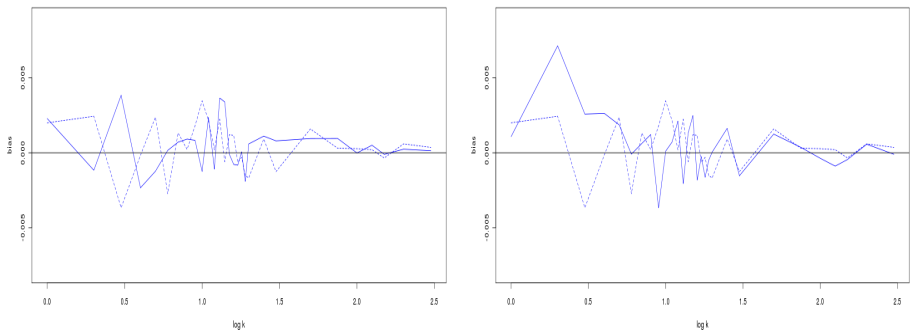
(c) $\beta_h = 0.4$ (d) $\beta_h = 0.4$

(e) $\beta_v = 0.4$ (f) $\beta_v = 0.4$

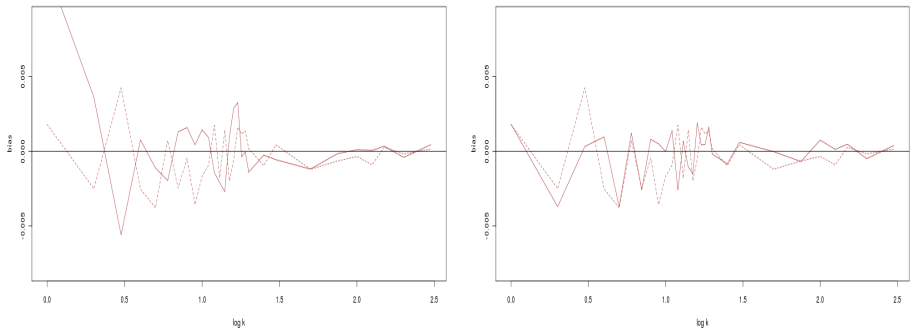
Figure 11: We use red, blue and brown for $\hat{\alpha}$, $\hat{\beta}_v$ and $\hat{\beta}_v$, respectively. The first column show the bias when we use MPLE, the solid line, and 2×2 MGPLE, which is the dotted line. In the second column we have replotted the dotted line along with the bias we get from using 3×3 MGPLE, which is the solid line.



(a) $\alpha = 0$ (b) $\alpha = 0$

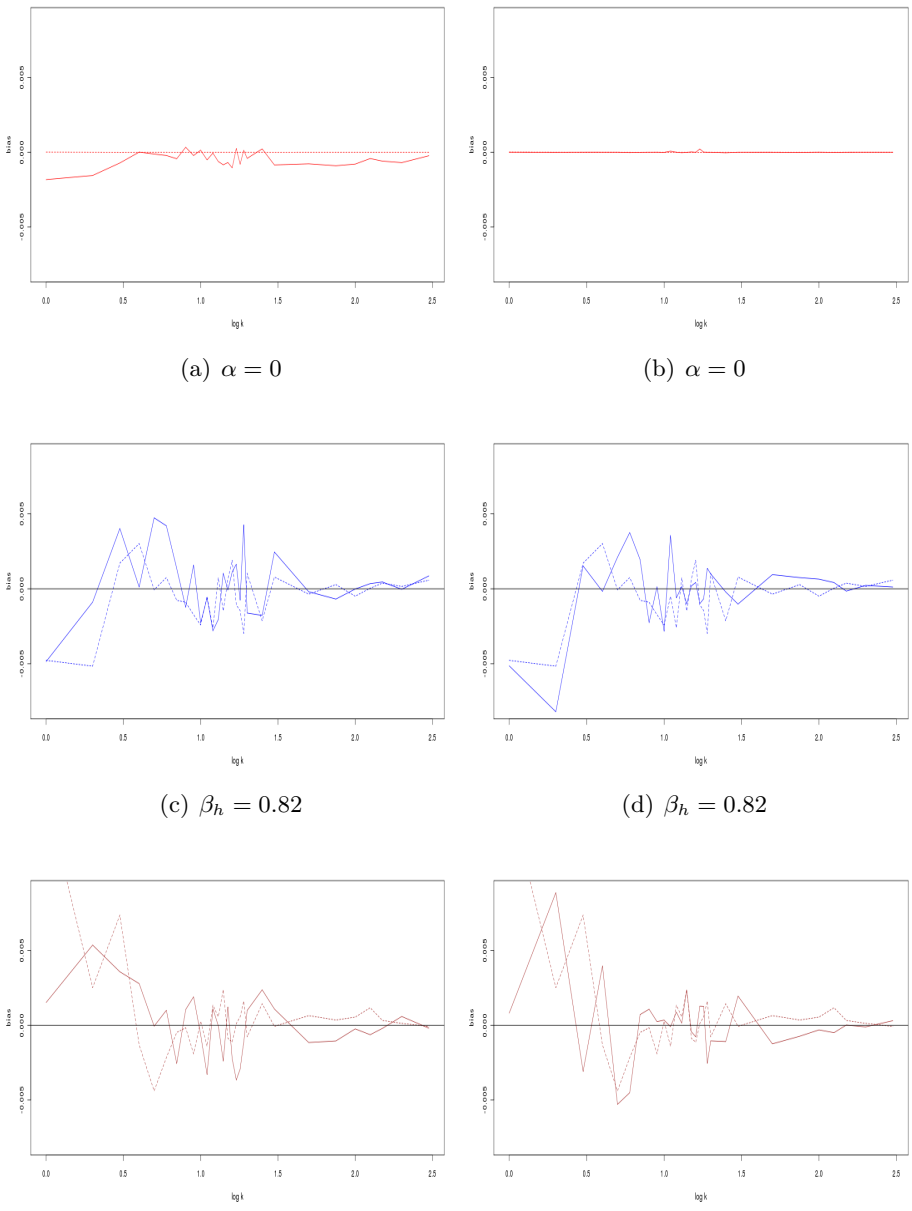


(c) $\beta_h = 0.8$ (d) $\beta_h = 0.8$



(e) $\beta_v = 0.8$ (f) $\beta_v = 0.8$

Figure 12: We use red, blue and brown for $\hat{\alpha}$, $\hat{\beta}_v$ and $\hat{\beta}_v$, respectively. The first column show the bias when we use MPLE, the solid line, and 2×2 MGPLE, which is the dotted line. In the second column we have replotted the dotted line along with the bias we get from using 3×3 MGPLE, which is the solid line.

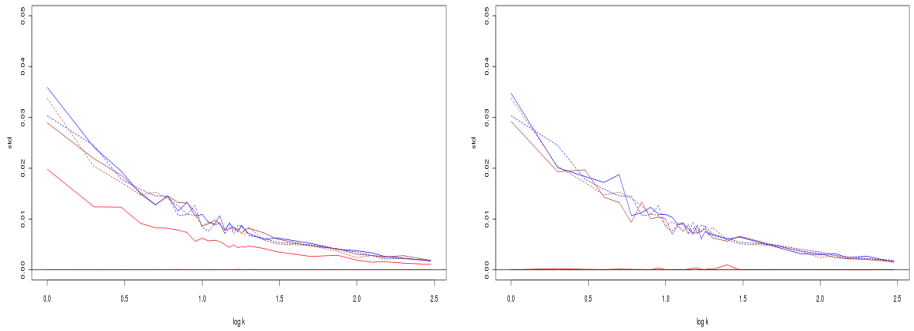


(a) $\alpha = 0$ (b) $\alpha = 0$

(c) $\beta_h = 0.82$ (d) $\beta_h = 0.82$

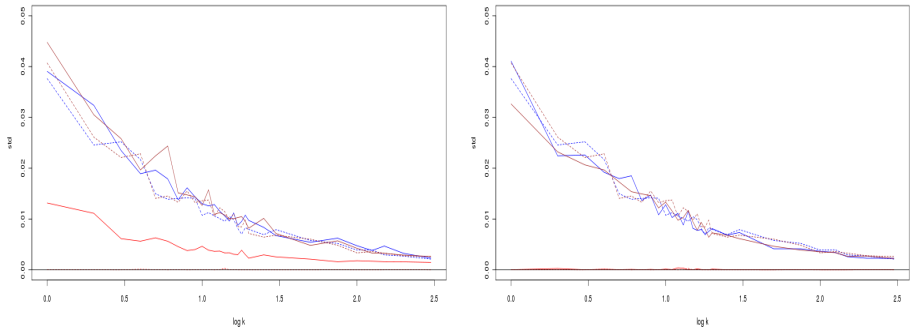
(e) $\beta_v = 0.82$ (f) $\beta_v = 0.82$

Figure 13: We use red, blue and brown for $\hat{\alpha}$, $\hat{\beta}_v$ and $\hat{\beta}_v$, respectively. The first column show the bias when we use MPLE, the solid line, and 2×2 MGPLE, which is the dotted line. In the second column we have replotted the dotted line along with the bias we get from using 3×3 MGPLE, which is the solid line.



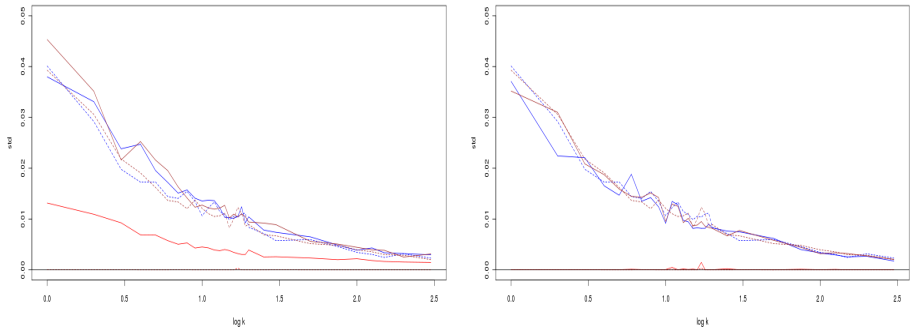
(a) $\alpha = 0, \beta_h = 0.4, \beta_h = 0.4$

(b) $\alpha = 0, \beta_h = 0.4, \beta_h = 0.4$



(c) $\alpha = 0, \beta_h = 0.8, \beta_h = 0.8$

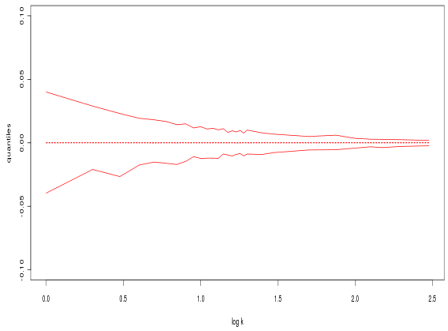
(d) $\alpha = 0, \beta_h = 0.8, \beta_h = 0.8$



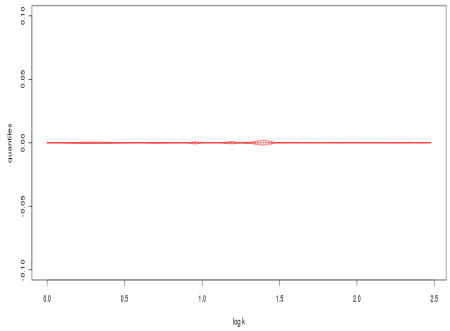
(e) $\alpha = 0, \beta_h = 0.82, \beta_h = 0.82$

(f) $\alpha = 0, \beta_h = 0.82, \beta_h = 0.82$

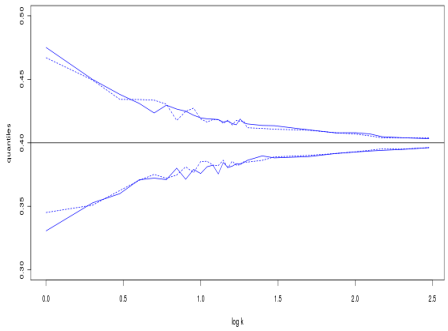
Figure 14: The standard deviations of the parameter estimates. We use the same colour code and layout as in figure 11.



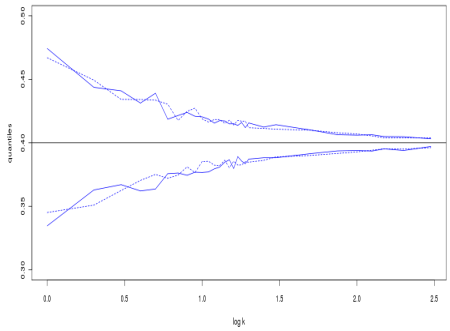
(a) $\alpha = 0$



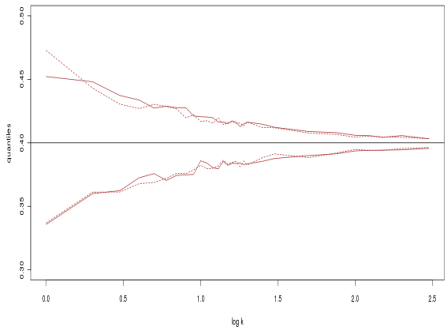
(b) $\alpha = 0$



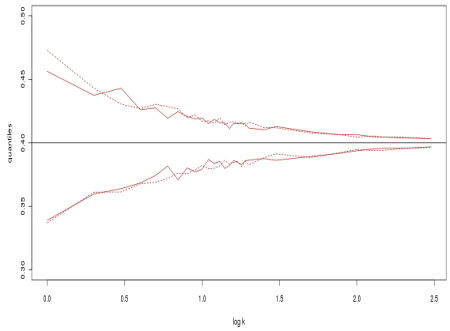
(c) $\beta_h = 0.4$ and $\beta_v = 0.4$



(d) $\beta_h = 0.4$ and $\beta_v = 0.4$

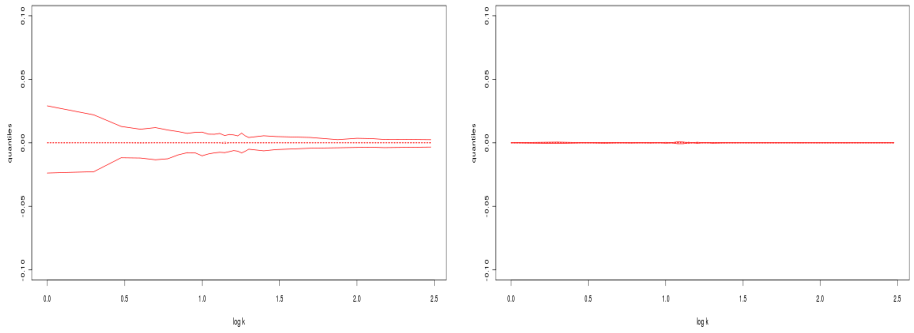


(e) $\beta_h = 0.4$ and $\beta_v = 0.4$



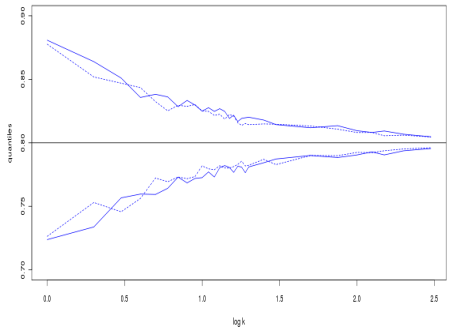
(f) $\beta_h = 0.4$ and $\beta_v = 0.4$

Figure 15: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

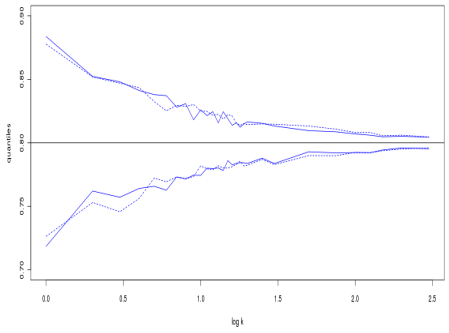


(a) $\alpha = 0$

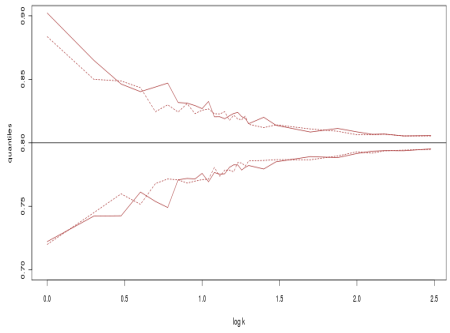
(b) $\alpha = 0$



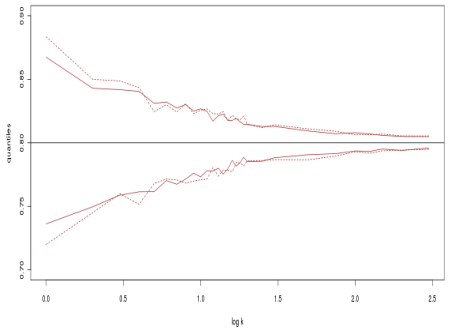
(c) $\beta_h = 0.4$ and $\beta_v = 0.4$



(d) $\beta_h = 0.4$ and $\beta_v = 0.4$

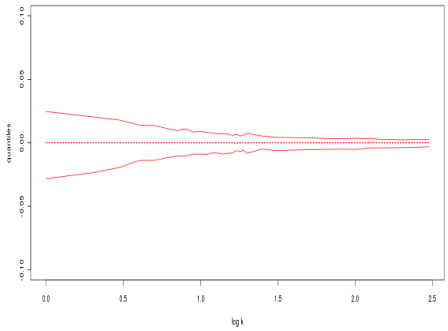


(e) $\beta_h = 0.4$ and $\beta_v = 0.4$

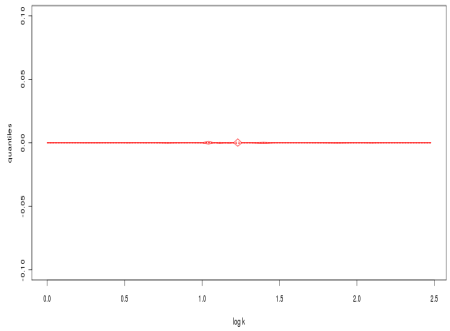


(f) $\beta_h = 0.4$ and $\beta_v = 0.4$

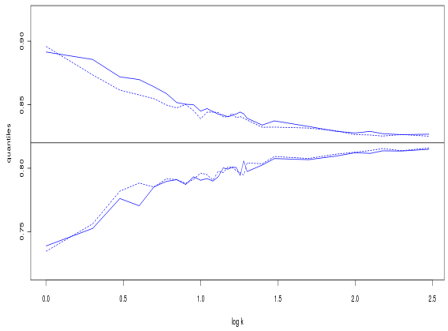
Figure 16: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.



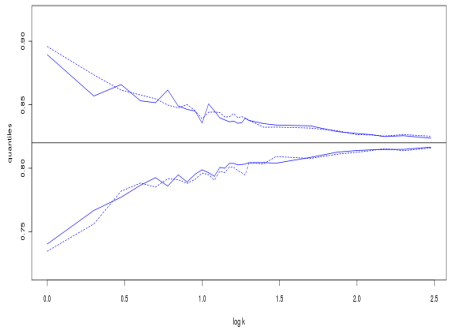
(a) $\alpha = 0$



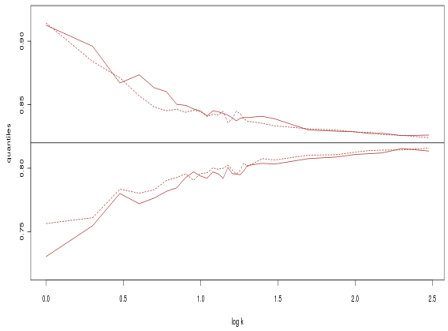
(b) $\alpha = 0$



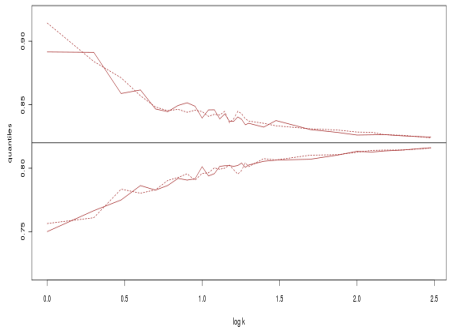
(c) $\beta_h = 0.4$ and $\beta_v = 0.4$



(d) $\beta_h = 0.4$ and $\beta_v = 0.4$



(e) $\beta_h = 0.4$ and $\beta_v = 0.4$



(f) $\beta_h = 0.4$ and $\beta_v = 0.4$

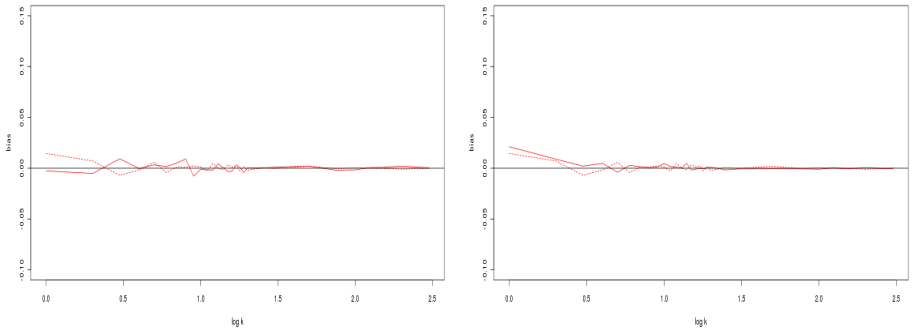
Figure 17: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

and $\alpha = \{0.1, 0.5, 1\}$. We are specially interested in studying the significance of the abundance parameter α on our estimation. In the previous section both methods had great success estimating $\alpha = 0$, as observed in figures 11 - 17. With $\alpha \neq 0$ the results tell a different story. In figures 18 -20 we see that the bias of $\hat{\alpha}$ increases with true value. This is also the case with the bias of $\hat{\beta_h}$ and $\hat{\beta_v}$, though not as much. Note that the intervals on the y-axes in the following figures are wider than the intervals in the preceding ones.

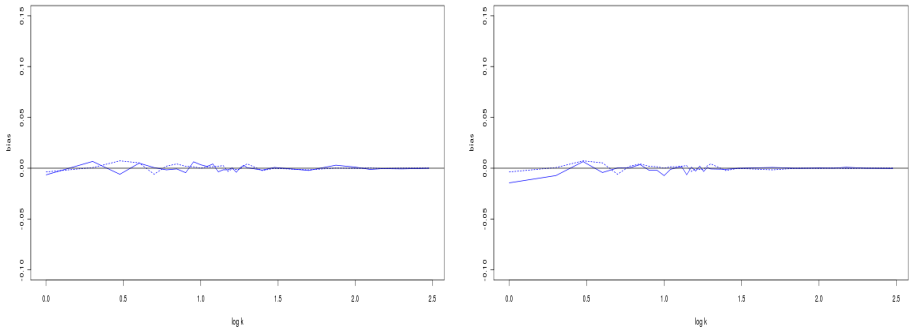
The significance of α is furthermore observed in figure 21. When $\alpha = 0.1$ it does not look that bad, the standard deviations are close to each other. However, when $\alpha = 0.5$ and $\alpha = 0.5$, we clearly see all the standard deviations increase, and $s_p(\hat{\alpha})$, $s_2(\hat{\alpha})$ and $s_3(\hat{\alpha})$ are larger than the others. We experience the same behaviour when we look at the confidence intervals in figures 22 - 24. When $\alpha = 0.1$ none of the intervals stand out, but when α increases it looks worse. All the intervals widen, and the intervals of $\frac{1}{n}\sum_{i=1}^{n}\hat{\alpha}_i$ widens a lot more than the interval for the other parameters.

Since the y-axes have been chosen to accommodate the case with $\alpha = 1$, it is not easy to see how MPLE and MGPLE fares when compared to each other when $\alpha = 0.1$ and $\alpha = 0.5$. In the appendix we have in figures 37 - 39 reproduced 21 - 23 with a smaller interval on the y-axis. For both $\alpha = 0.1$ and $\alpha = 0.5$ we see in figure 37 that $s_2(\hat{\alpha})$ most of the time is smaller than $s_p(\hat{\alpha})$, and in turn $s_3(\hat{\alpha})$ is smaller than $s_2(\hat{\alpha})$. This is also the case for the other parameters, but it is not as obvious. In figure 38 we see that we get narrower intervals by using MGPLE, specially for $\hat{\alpha}$. If we look at the confidence interval in figure 39, we see that we gain less by using MGPLE, and as mentioned before the intervals are much wider. This is also the case when $\alpha = 1$.

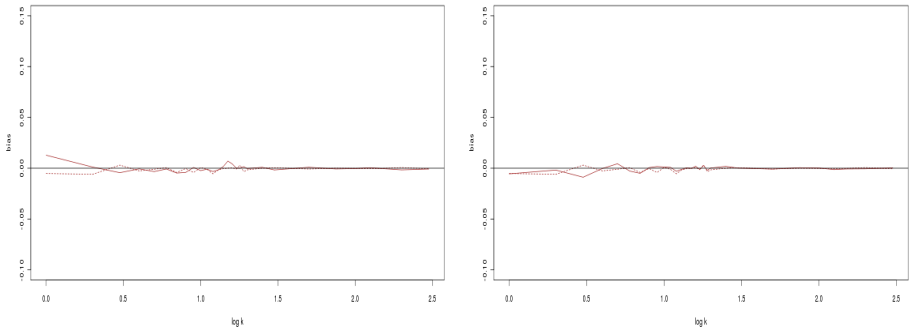
In the next example we will use the same values for α, but we will increase the difference between β_h and β_v and we will use $\beta_h = 0.40$ and $\beta_v = 0.80$. The result of our evaluations are presented in figures 25 - 31. Note that the intervals on the y-axes are narrower in these figures than in the preceding ones. We have reproduced figures 29 and 30 in the appendix as figures 41 and . When we look at these figures we draw the same conclusions as in the previous case. We experience the same problems when α increases. However in this case the standard deviations in figure 28 are smaller, and the confidence intervals in figures 29 - 31 are narrower, so the problems are not of the same magnitude.



(a) $\alpha = 0.1$ (b) $\alpha = 0.1$

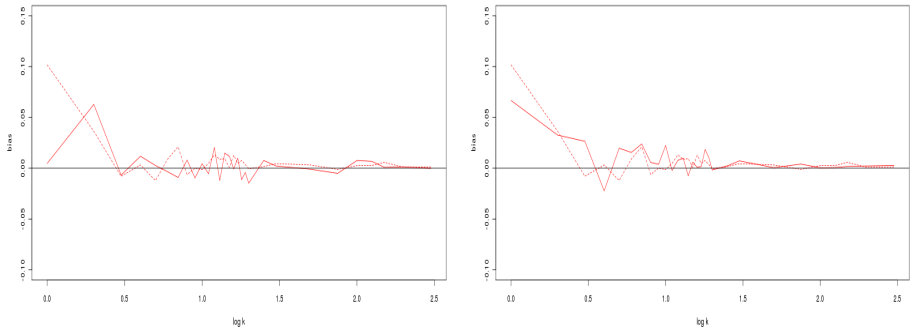


(c) $\beta_h = 0.83$ (d) $\beta_h = 0.83$



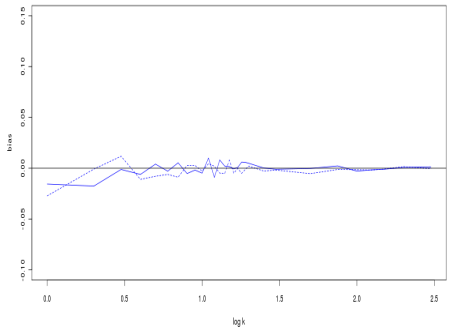
(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

Figure 18: The bias of the parameter estimates. We use the same colour code and layout as in figure 11.

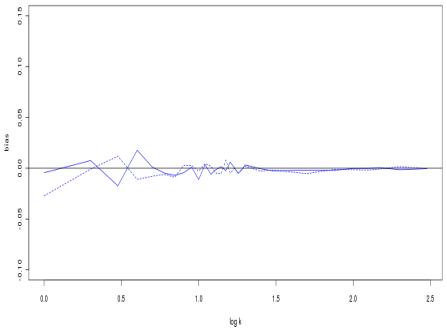


(a) $\alpha = 0.5$

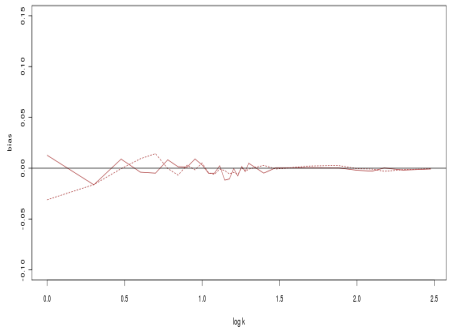
(b) $\alpha = 0.5$



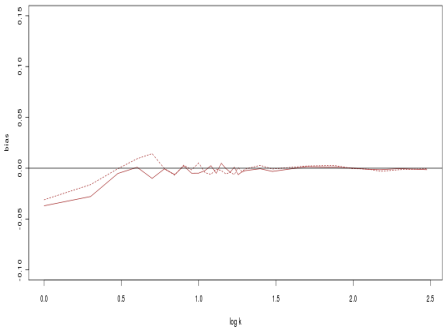
(c) $\beta_h = 0.83$



(d) $\beta_h = 0.83$

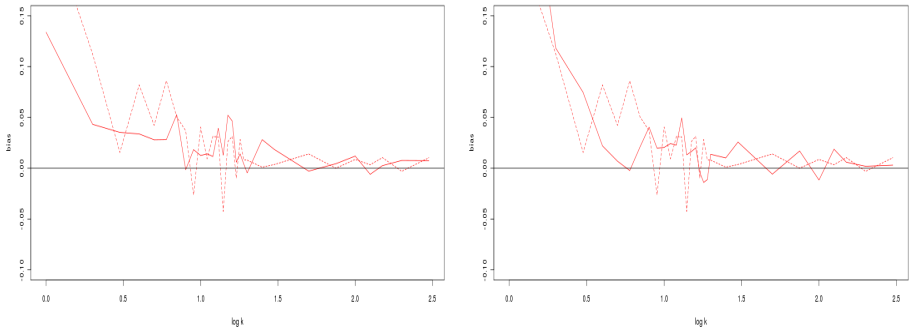


(e) $\beta_v = 0.80$

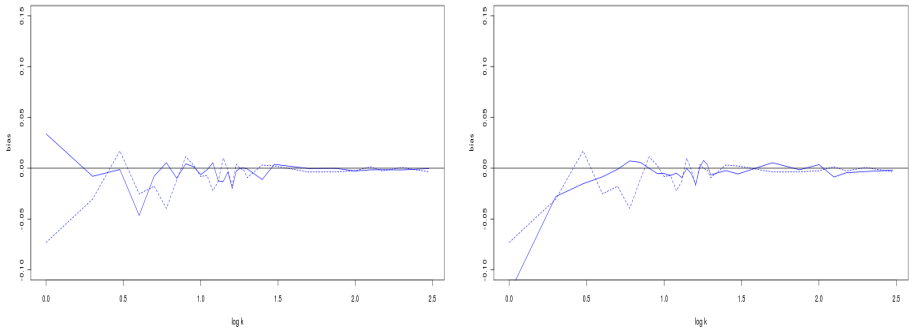


(f) $\beta_v = 0.80$

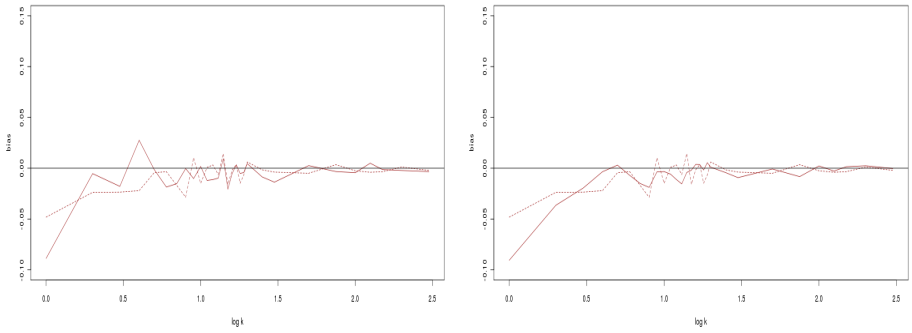
Figure 19: The bias of the parameter estimates. We use the same colour code and layout as in figure 11.



(a) $\alpha = 1$ (b) $\alpha = 1$

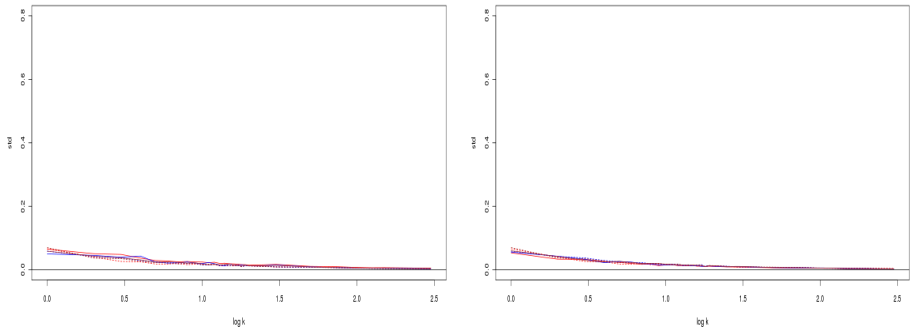


(c) $\beta_h = 0.83$ (d) $\beta_h = 0.83$



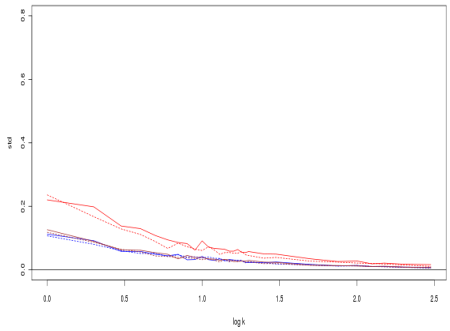
(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

Figure 20: The bias of the parameter estimates. We use the same colour code and layout as in figure 11.

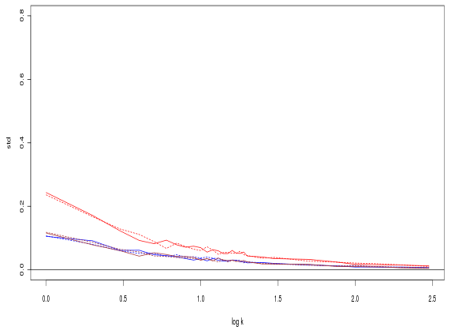


(a) $\alpha = 0.1, \beta_h = 0.83, \beta_h = 0.80$

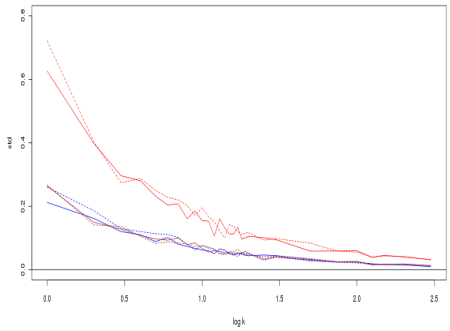
(b) $\alpha = 0.1, \beta_h = 0.83, \beta_h = 0.80$



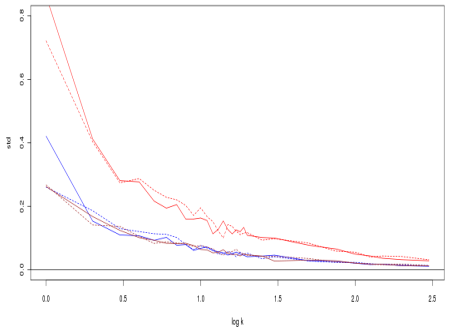
(c) $\alpha = 0.5, \beta_h = 0.83, \beta_h = 0.80$



(d) $\alpha = 0.5, \beta_h = 0.83, \beta_h = 0.80$

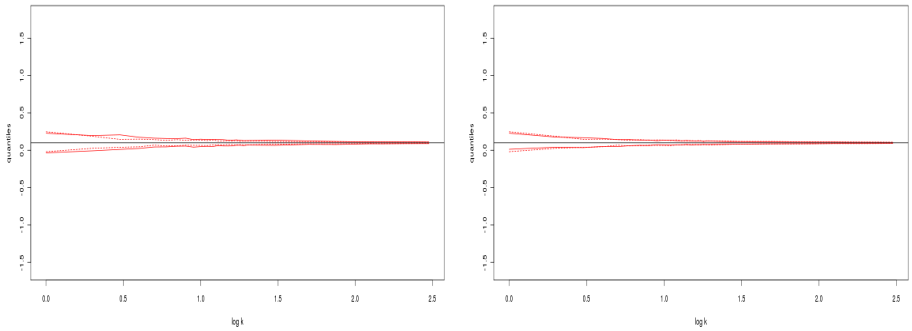


(e) $\alpha = 1, \beta_h = 0.83, \beta_h = 0.80$



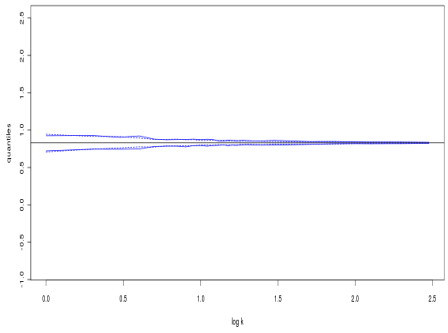
(f) $\alpha = 1, \beta_h = 0.83, \beta_h = 0.80$

Figure 21: The standard deviations of the parameter estimates. We use the same colour code and layout as in figure 11.

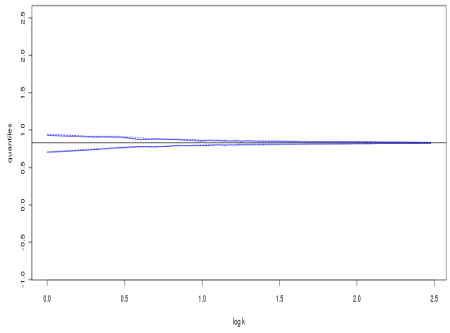


(a) $\alpha = 0.1$

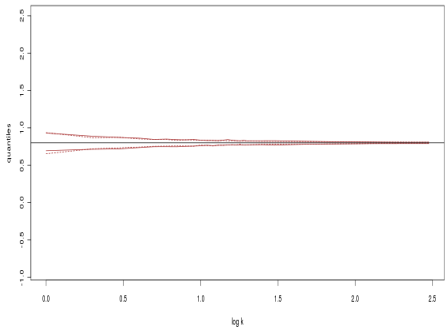
(b) $\alpha = 0.1$



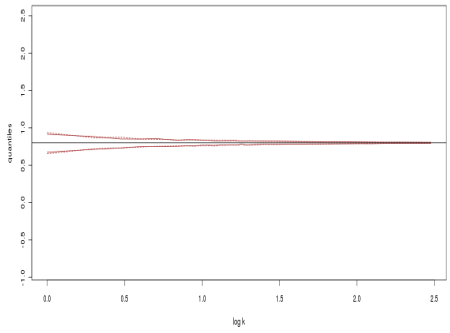
(c) $\beta_h = 0.83$



(d) $\beta_h = 0.83$

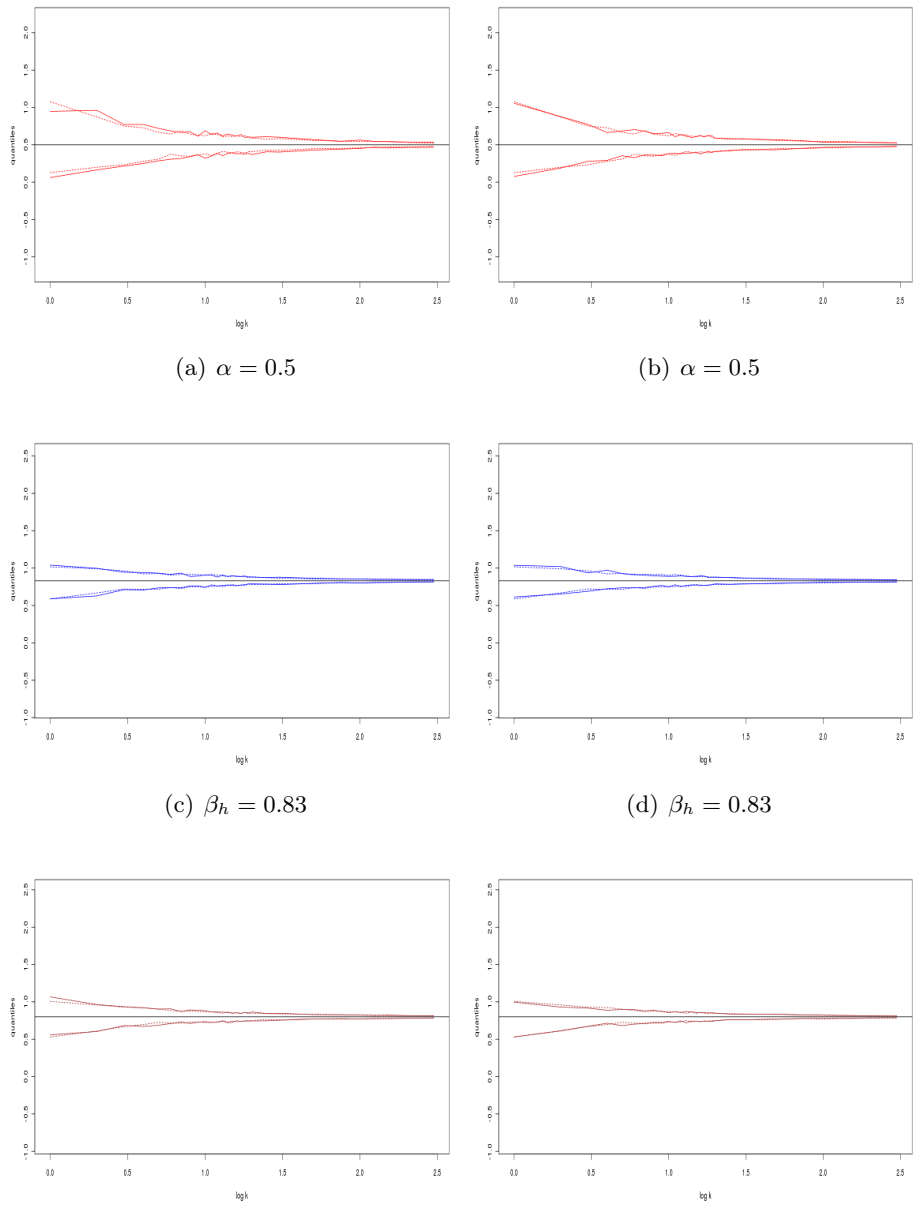


(e) $\beta_v = 0.80$



(f) $\beta_v = 0.80$

Figure 22: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

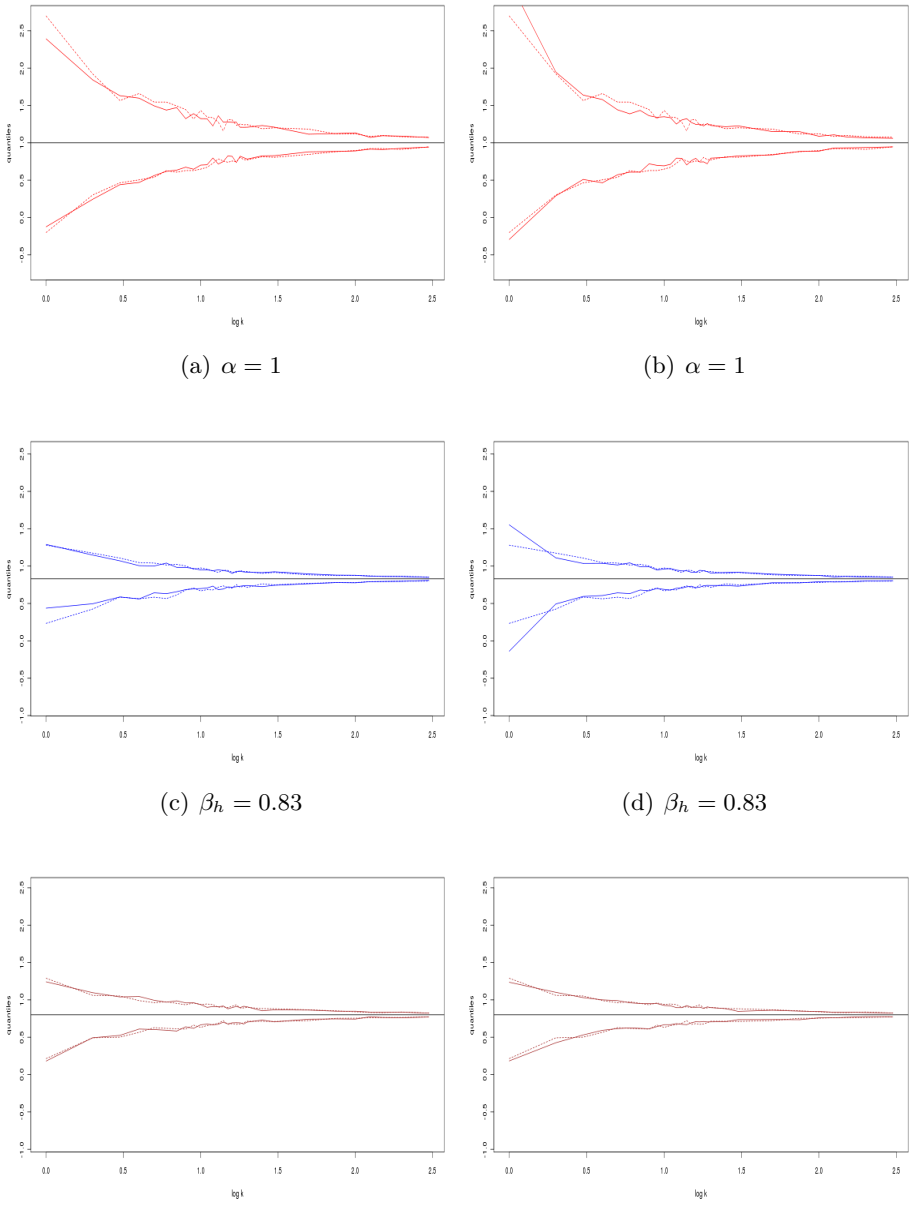


(a) $\alpha = 0.5$ (b) $\alpha = 0.5$

(c) $\beta_h = 0.83$ (d) $\beta_h = 0.83$

(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

Figure 23: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

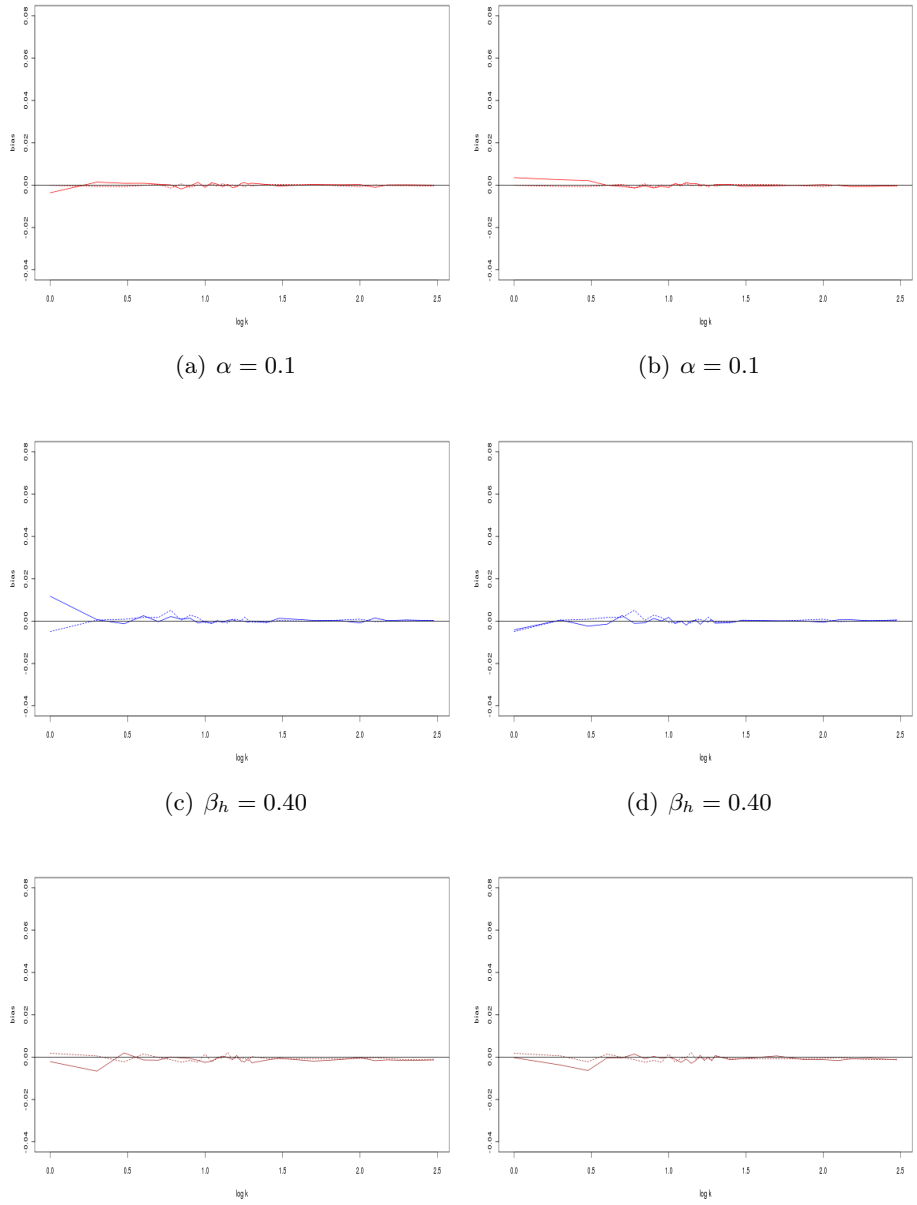


(a) $\alpha = 1$ (b) $\alpha = 1$

(c) $\beta_h = 0.83$ (d) $\beta_h = 0.83$

(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

Figure 24: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

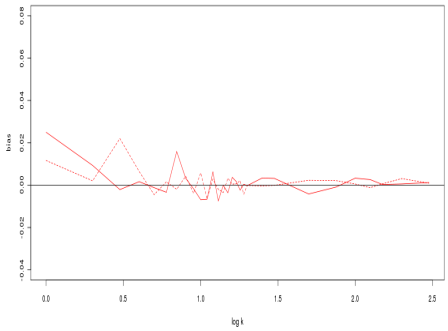


(a) $\alpha = 0.1$ (b) $\alpha = 0.1$

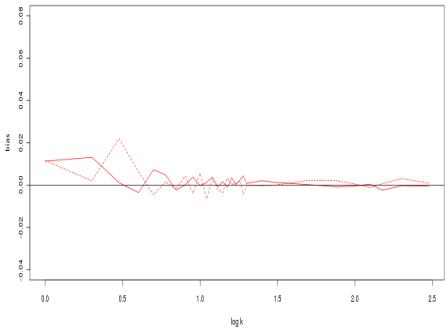
(c) $\beta_h = 0.40$ (d) $\beta_h = 0.40$

(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

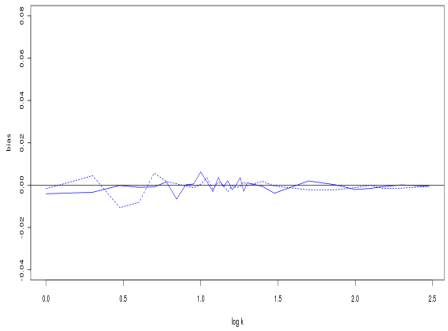
Figure 25: The bias of the parameter estimates. We use the same colour code and layout as in figure 11.



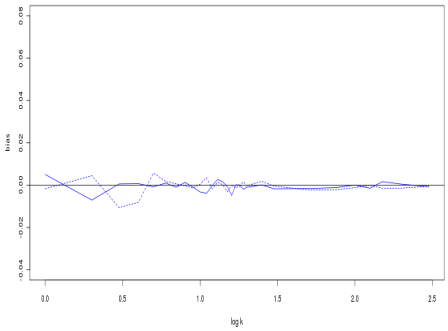
(a) $\alpha = 0.5$



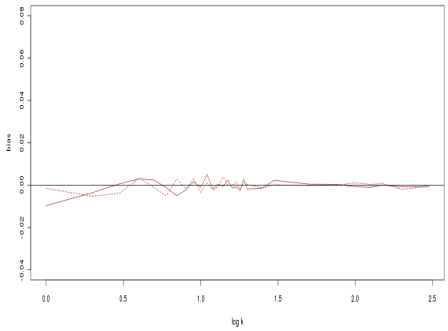
(b) $\alpha = 0.5$



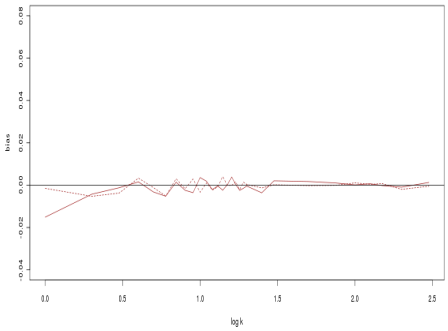
(c) $\beta_h = 0.40$



(d) $\beta_h = 0.40$

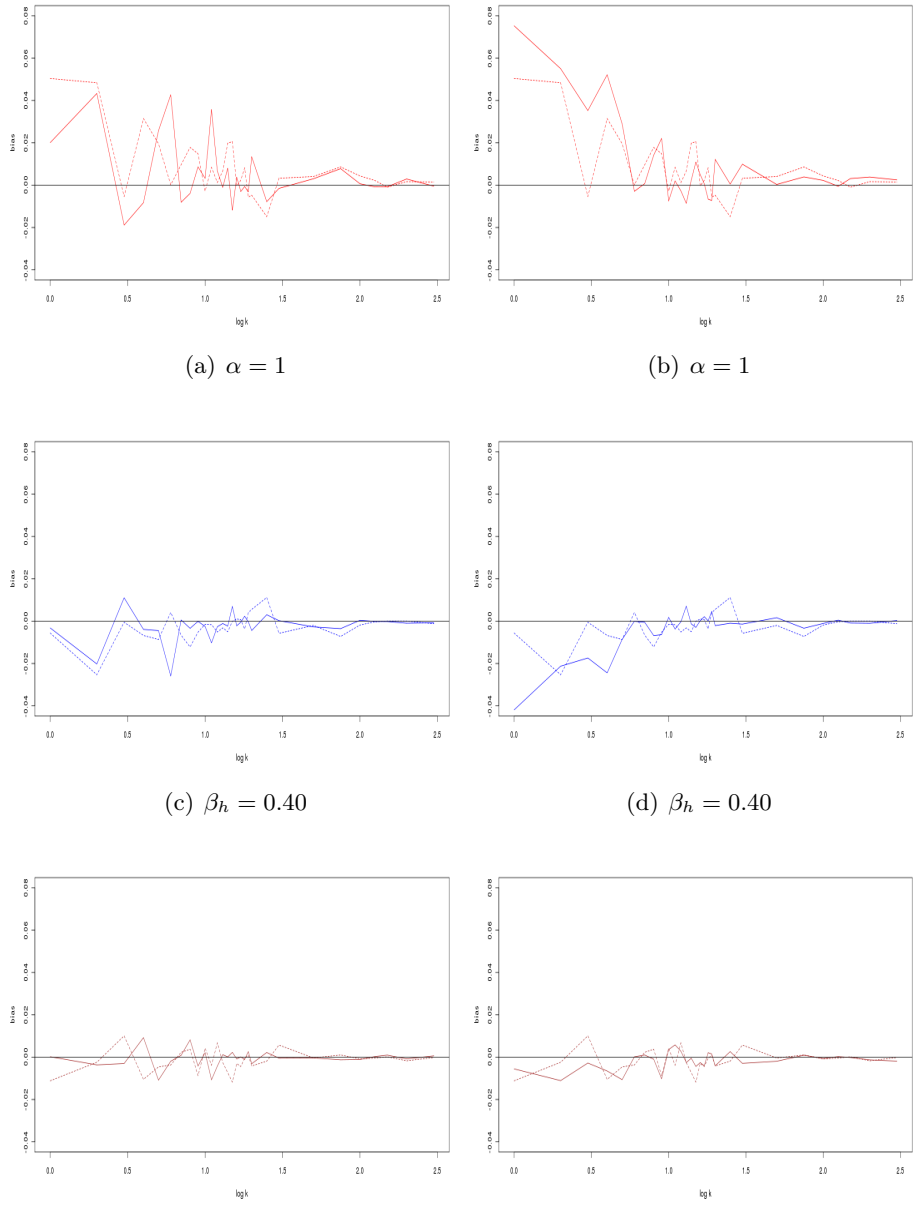


(e) $\beta_v = 0.80$



(f) $\beta_v = 0.80$

Figure 26: The bias of the parameter estimates. We use the same colour code and layout as in figure 11.

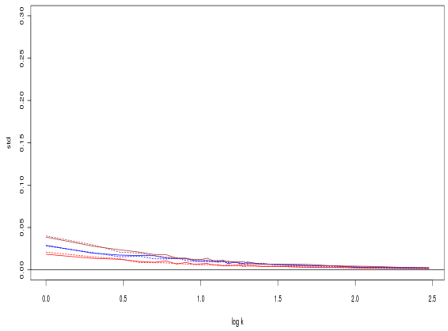


(a) $\alpha = 1$ (b) $\alpha = 1$

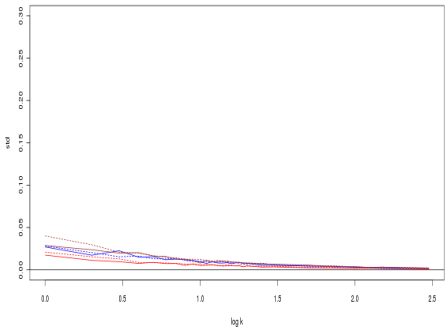
(c) $\beta_h = 0.40$ (d) $\beta_h = 0.40$

(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

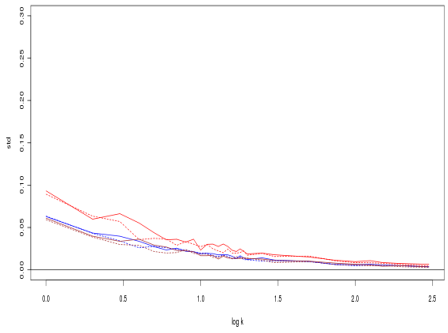
Figure 27: The bias of the parameter estimates. We use the same colour code and layout as in figure 11.



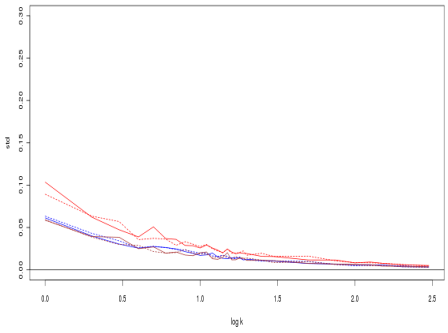
(a) $\alpha = 0.1, \beta_h = 0.40, \beta_h = 0.80$



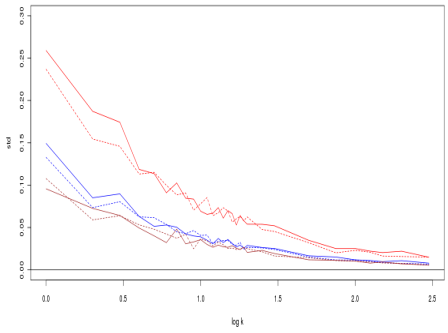
(b) $\alpha = 0.1, \beta_h = 0.40, \beta_h = 0.80$



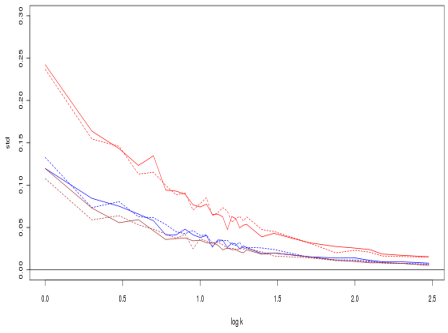
(c) $\alpha = 0.5, \beta_h = 0.40, \beta_h = 0.80$



(d) $\alpha = 0.5, \beta_h = 0.40, \beta_h = 0.80$

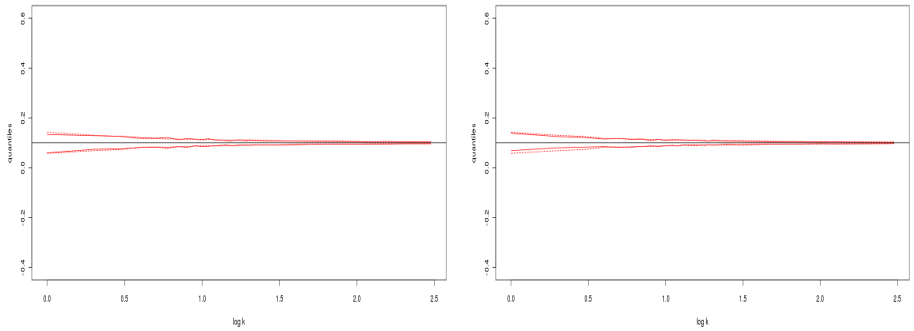


(e) $\alpha = 1, \beta_h = 0.40, \beta_h = 0.80$



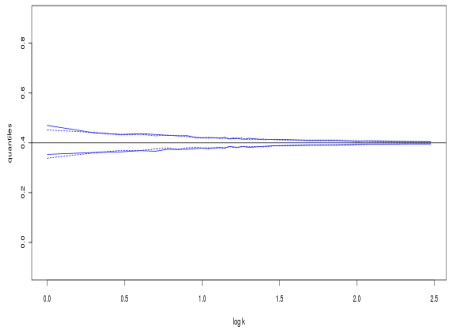
(f) $\alpha = 1, \beta_h = 0.40, \beta_h = 0.80$

Figure 28: The standard deviations of the parameter estimates. We use the same colour code and layout as in figure 11.

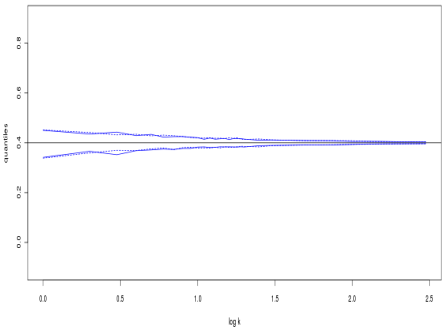


(a) $\alpha = 0.1$

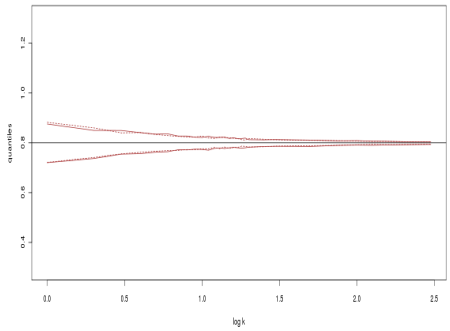
(b) $\alpha = 0.1$



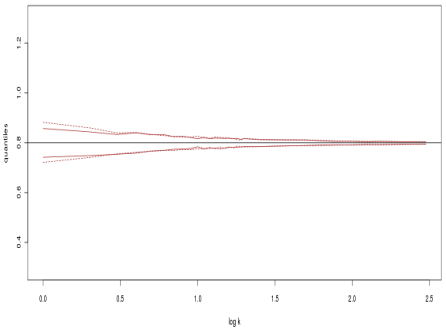
(c) $\beta_h = 0.40$



(d) $\beta_h = 0.40$

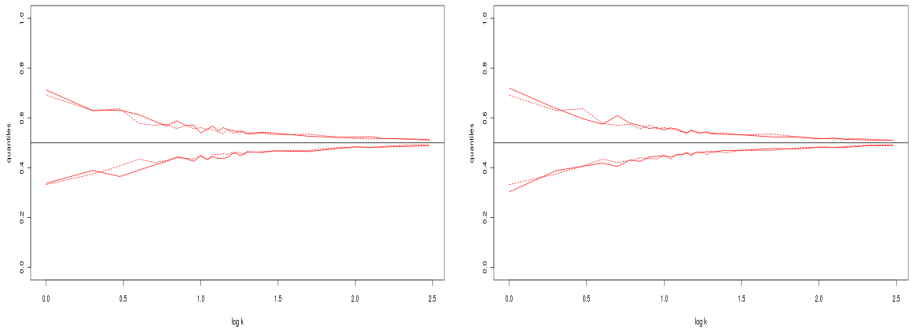


(e) $\beta_v = 0.80$



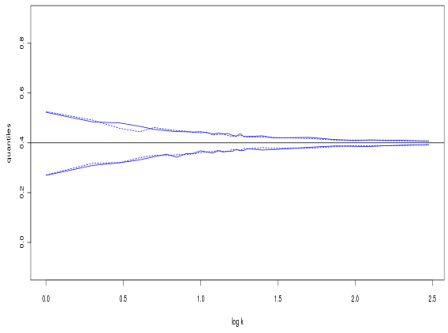
(f) $\beta_v = 0.80$

Figure 29: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

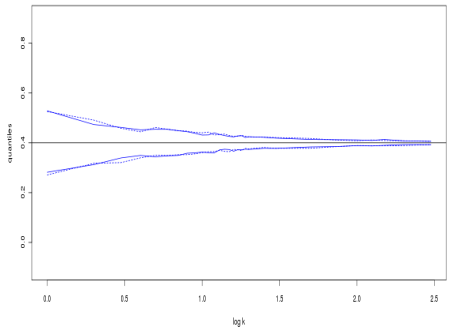


(a) $\alpha = 0.5$

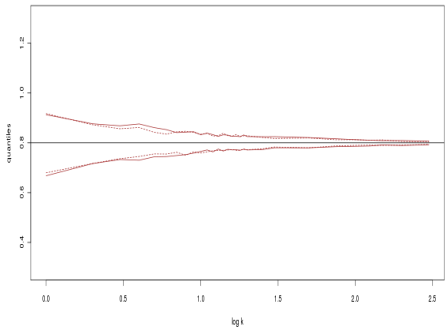
(b) $\alpha = 0.5$



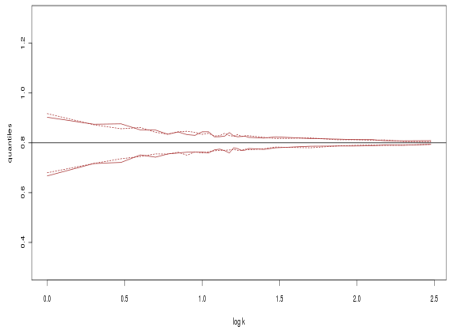
(c) $\beta_h = 0.40$



(d) $\beta_h = 0.40$

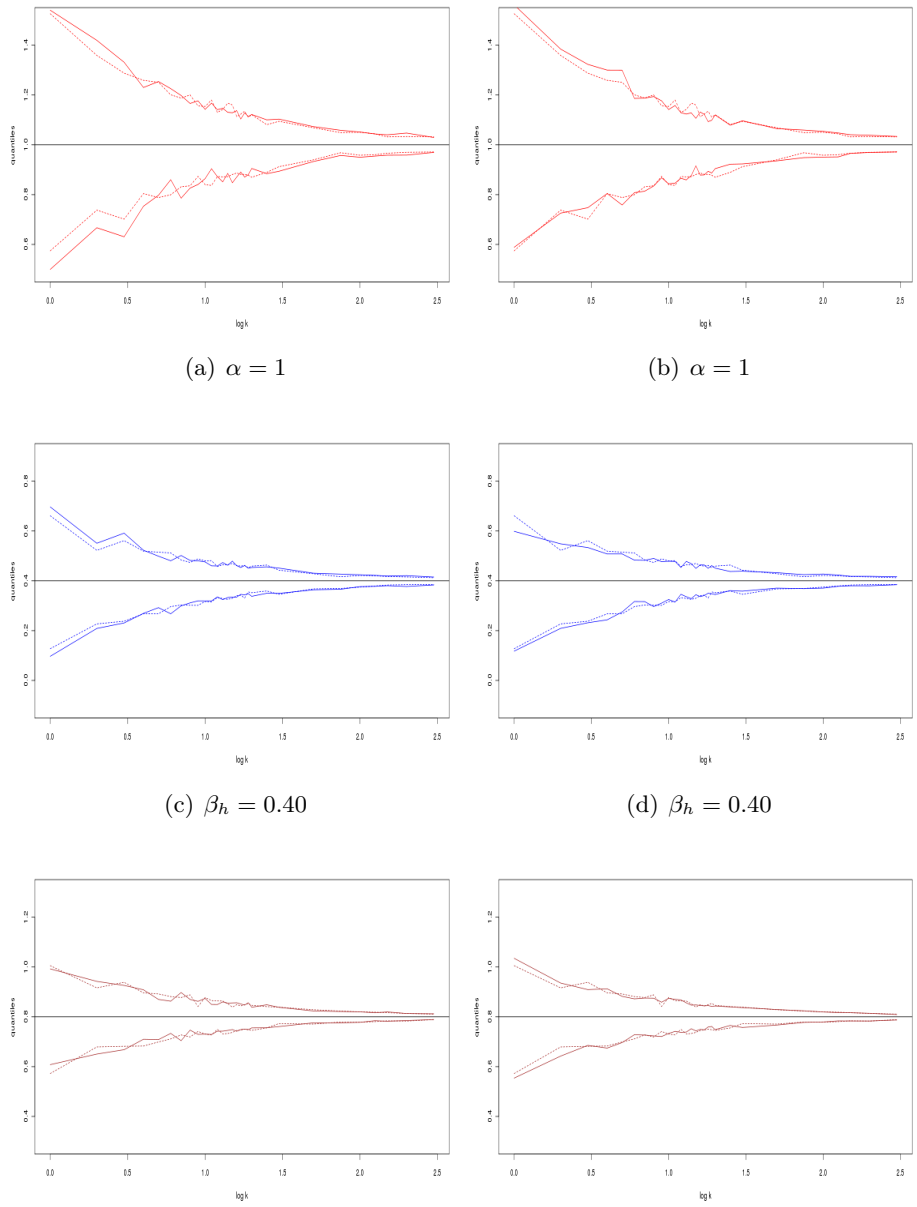


(e) $\beta_v = 0.80$



(f) $\beta_v = 0.80$

Figure 30: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.



(a) $\alpha = 1$ (b) $\alpha = 1$

(c) $\beta_h = 0.40$ (d) $\beta_h = 0.40$

(e) $\beta_v = 0.80$ (f) $\beta_v = 0.80$

Figure 31: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

5.3 The Sisim model

We will now estimate the parameters of the model in chapter 3.2. Contrary to the other models we do not longer choose the true parameter values when we generate observations. Instead we have given parameters $\theta_0, \theta_1, ..., \theta_{12}$, which the observations depend on, and we want to find out how good the methods estimate them. From equation (28) and (30) we get

$$l_p(\theta; \boldsymbol{x}^1, .., \boldsymbol{x}^K) = \sum_{k=1}^{K} \log l_p(\theta; \boldsymbol{x}^k) \tag{55}$$

$$l_g(\theta; \boldsymbol{x}^1, .., \boldsymbol{x}^K) = \sum_{k=1}^{K} \log l_g(\theta; \boldsymbol{x}^k), \tag{56}$$

which is the log pseudo- and general pseudo-likelihood, respectively. To get the estimated parameter vector $\hat{\theta} = (\hat{\theta_0}, \hat{\theta_1}, ..., \hat{\theta}_{12})^T$ we solve the problems

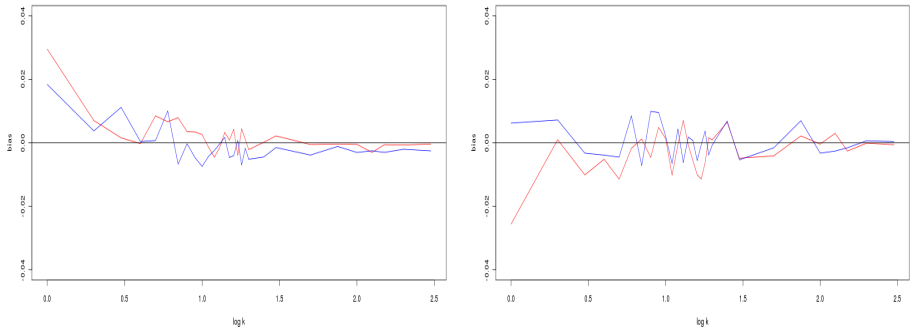
$$\hat{\theta} = \arg\max_{\theta} l_p(\theta; \boldsymbol{x}^1, .., \boldsymbol{x}^K) \tag{57}$$

$$\hat{\theta} = \arg\max_{\theta} l_g(\theta; \boldsymbol{x}^1, .., \boldsymbol{x}^K). \tag{58}$$

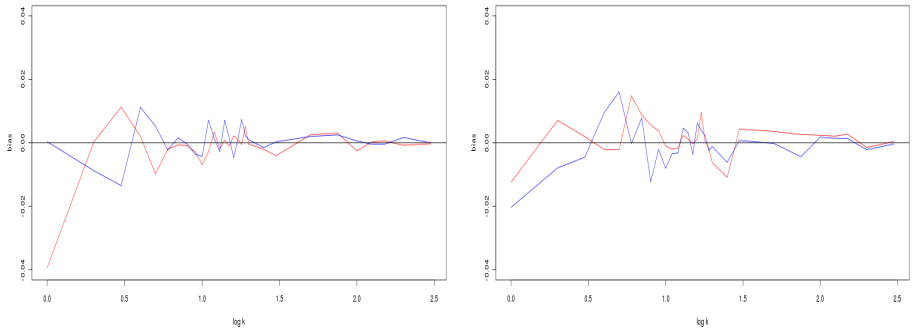
As before we repeat the optimization above 50 times, and we apply the same statistics as we did for the Ising models. We will only consider one block, and it is of size 2×2. In the preceding figures we will show the results for some of the parameters. The result for the rest may be found in appendix A in figures 43 - 48.

In figure 32 we see that the biases approaches 0 when the number of observations increase. When we look at the standard deviations we see in figure 33 that $s_2(\cdot)$ is smaller than $s_p(\cdot)$ most of the time, for $k > 100$ this difference is small. Furthermore we see that the standard deviations in 46(a) - 46(a) approach the same level. In figure 34 we draw the conclusion that the confidence intervals we get by using MGPLE, is a bit norrower than the ones we get with MPLE.

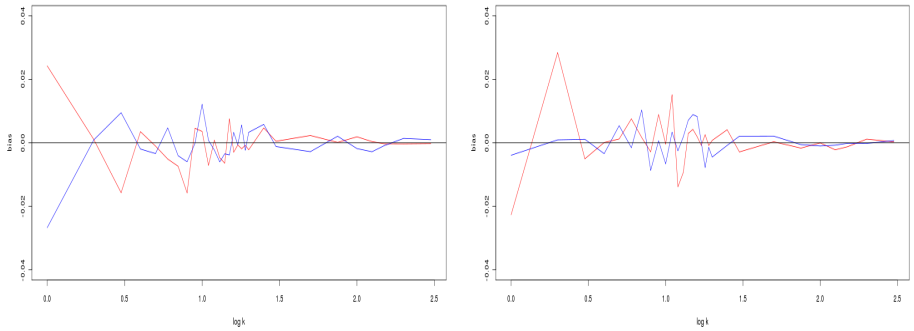
Another way to test the methods, is to first estimate parameters $\hat{\theta}$ from observations $\boldsymbol{x}^1, \boldsymbol{x}^2, ..., \boldsymbol{x}^K$, and then generate new realisations with the estimated parameter. If we compare the true realisations with the new ones, is it is possible to tell the difference? The answer to this conundrum is given in figures 35 and 35. In the first row of figure 35 we have generated three realisations with parameters estimated from $l_p(\theta; \boldsymbol{x}^1, .., \boldsymbol{x}^5)$. In the second row we have shown three of the observations. In the third row we have used $l_p(\theta; \boldsymbol{x}^1, .., \boldsymbol{x}^{50})$ to estimate the parameter, and in the last row we have shown three of the 50 observations. We have repeated this procedure in figure35 with MGPLE. By studying the new realisations we do not notice anything that makes them stand out compared with the observations, at least not with the naked eye. However we also see that the observations in rows two and four vary a lot. If we had been able to see a difference between them and the new realisations, the error of the estimation must have been severe. In appendix A we have examples were we have used $K = 10$ and $K = 100$, they are shown in figures figures 35 and 35.



(a) θ_0 (b) θ_2

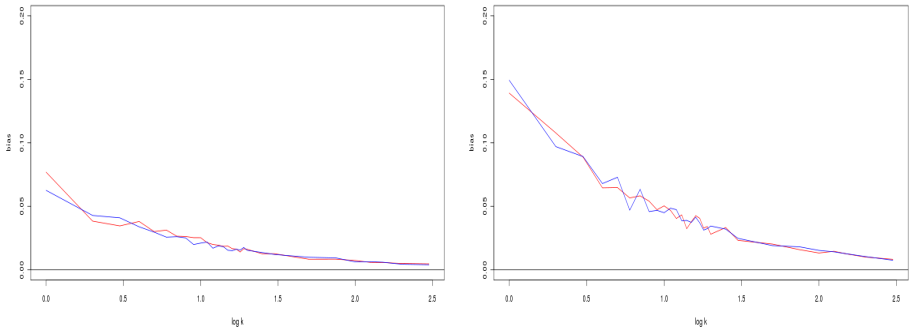


(c) θ_4 (d) θ_7

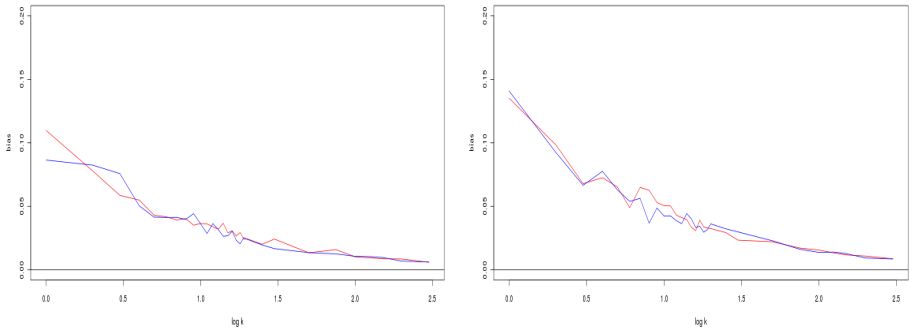


(e) θ_9 (f) θ_{12}

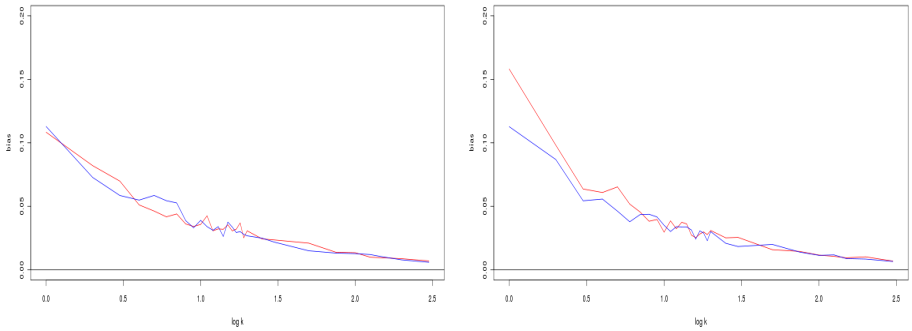
Figure 32: The red and blue lines are the biases when we use MPLE and 2×2 MGPLE, respectively



(a) θ_0 (b) θ_2

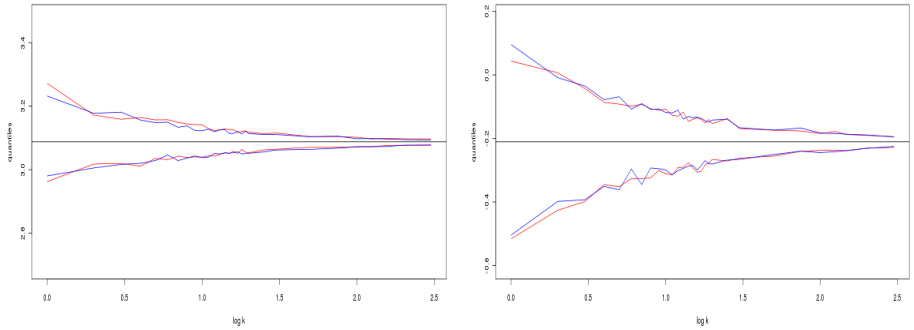


(c) θ_4 (d) θ_7



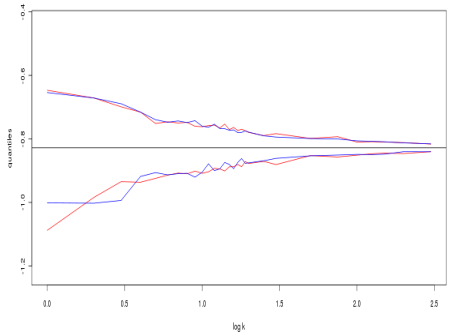
(e) θ_9 (f) θ_{12}

Figure 33: The red and blue lines are the standard deviations when we use MPLE and 2×2 MGPLE, respectively.

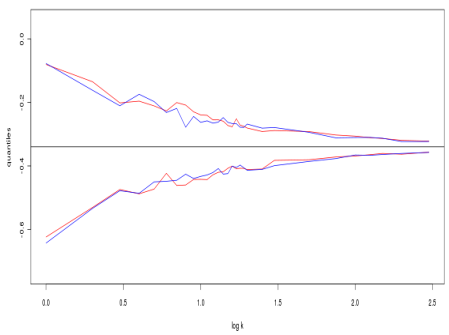


(a) θ_0

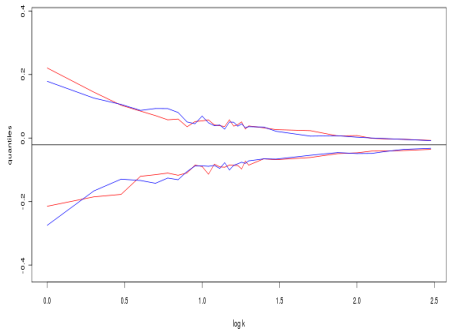
(b) θ_2



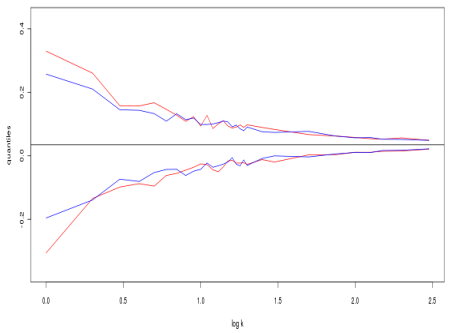
(c) θ_4



(d) θ_7

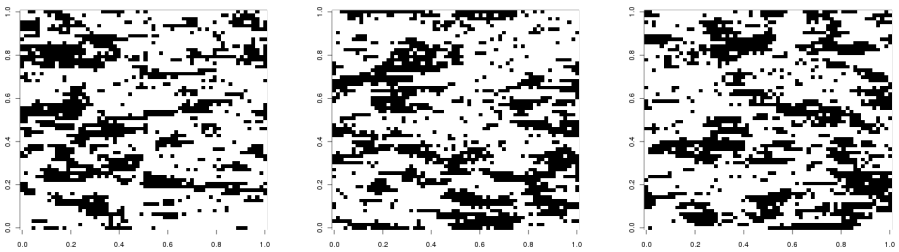


(e) θ_9

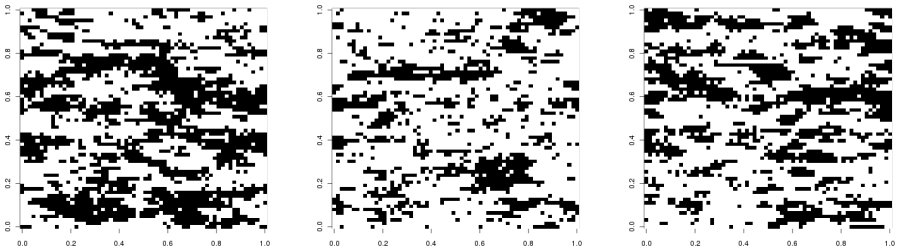


(f) θ_{12}

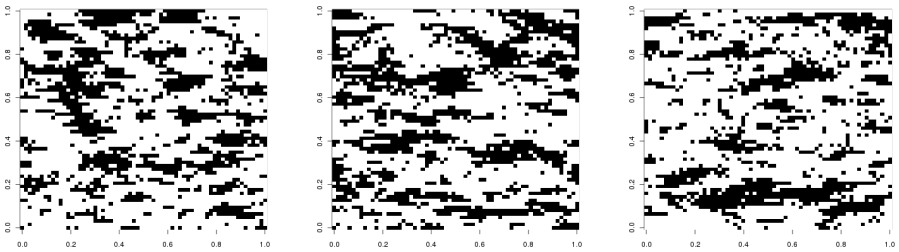
Figure 34: The lines are the upper and lower quantiles of a 0.95 confidence interval, and the black line is the value of the true parameter. The red and blue lines are the quantiles when we use MPLE and 2×2 MGPLE, respectively.



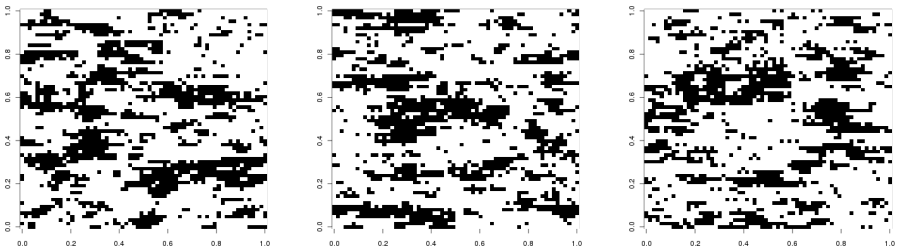
(a) $k = 5$ (b) $k = 5$ (c) $k = 5$



(d) $k = 5$ (e) $k = 5$ (f) $k = 5$

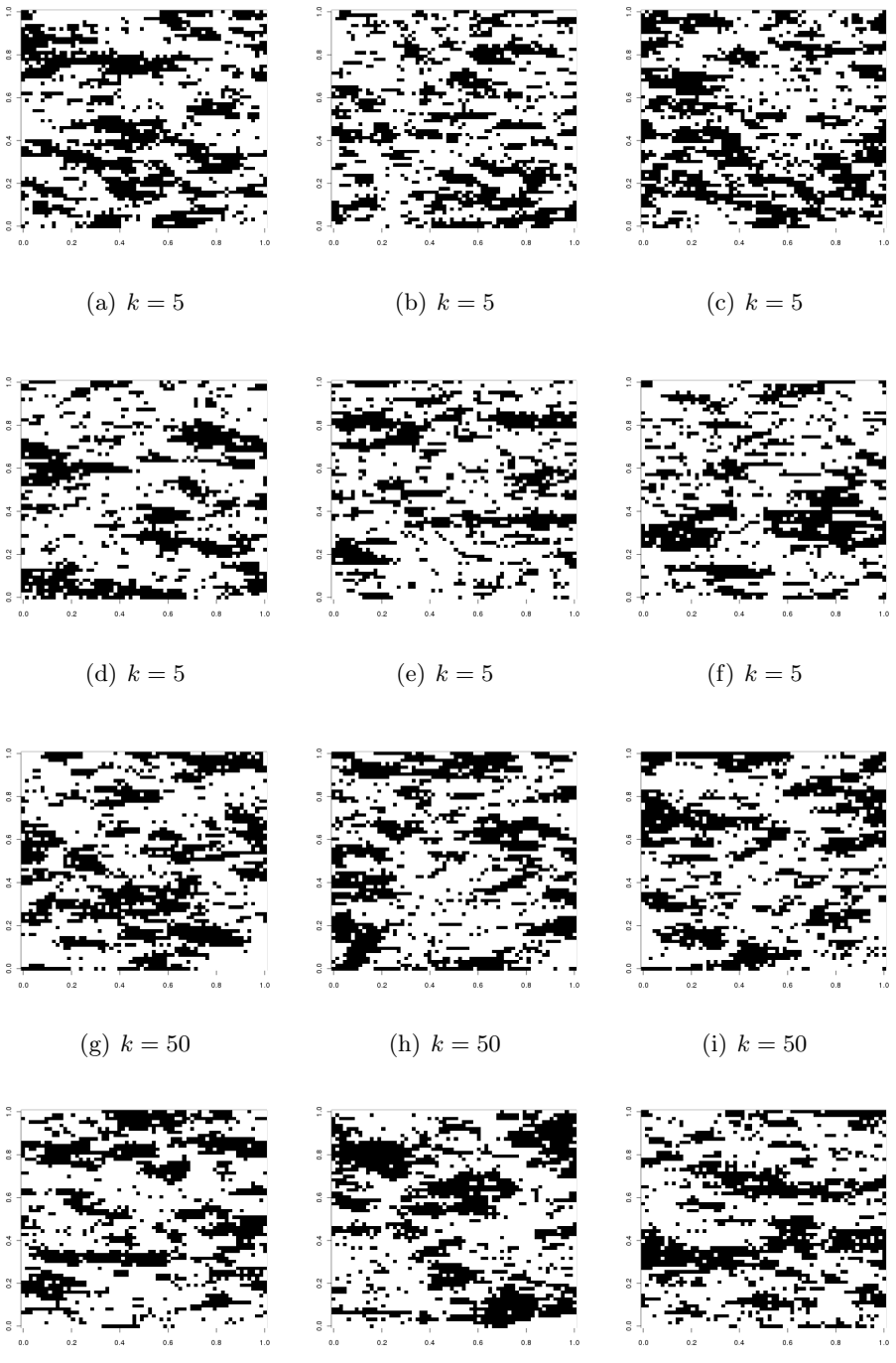


(g) $k = 50$ (h) $k = 50$ (i) $k = 50$



(j) $k = 50$ (k) $k = 50$ (l) $k = 50$

Figure 35: Realisations generated with parameters found with MPLE.



(a) $k = 5$ (b) $k = 5$ (c) $k = 5$

(d) $k = 5$ (e) $k = 5$ (f) $k = 5$

(g) $k = 50$ (h) $k = 50$ (i) $k = 50$

(j) $k = 50$ (k) $k = 50$ (l) $k = 50$

Figure 36: Realisations generated with parameters found with MGPLE.

6 Closing remarks

The purpose of this paper has been to study and compare the performance of MPLE and MGPLE. From the results in chapter 5 we draw the conclusion that the methods perform better when the number of observations increases. We also see that MGPLE is the best method, and if we use a block of size 3×3, we get better result than when we use a 2×2 block, though the performance is not dramatically improved. When the number of observations exceed 100, the difference between the methods dwindles. However there are nuances in the results for the models we have considered.

If we look at the standard deviation and the width of the confidence intervals, we get the best results for the Ising model, which is not surprising since it is the least complicated model. For the extended Ising model in chapter 5.2 we see that the outcome of our estimation depends on the true parameter α. If it is close to 0 we get the best results. When we estimate from observations generated with $\alpha = 0$ and $\beta_h = \beta_v$, both methods have success with estimating α, but MGPLE is by far the better method. When $\alpha \neq 0$ and $\beta_h \neq \beta_v$ we see that the quality of the estimates worsen with the increasing value of α. For the Sisim model we see both methods do a good job with estimating the parameters. MGPLE is slightly better in most cases, but this varies between the parameters. However, for this model we only used a 2×2 block.

In our examples we have only used blocks of size 3×3 and 2×2. Unfortunately we did not have the time to test MGPLE with blocks of larger size. It would also have been interesting to look at more parameter combinations for the extended Ising model. If we have $\beta_h = 0.40$ and $\beta_v = 0.80$, the results are better than the case with $\beta_h = 0.83$ and $\beta_v = 0.80$. We would like to find out if it is because the difference between the parameters is greater, or if it is because one of them have a small value. However the computer computations have been very time consuming, with Coupling from the past as the bottle neck, so we did not have the time to run all the simulations we wanted to.

References

Besag, J. E. (1974), 'Spatial interacion and the statistical analysis of lattice systems (with discussion)', *J. Roy. Statist. Soc. Ser. B* **36**, 192–236.

Geyer, C. J. & Thompson, E. A. (1992), 'Constrained Monte Carlo maximum likelihood for dependent data', *J. Roy. Statist. Soc. Ser. B* **54**, 657–699.

Grimmet, G. (1987), 'Interacting particle systems and random media: An overview', *Inetrnational Statistics Review* **55**, 49–62.

Huang, F. & Ogata, Y. (2002), 'Generalized pseudo-likelihood estimates for Markov random fields on lattices', *Ann. Ist. Statist. Math.* **54**, 1–18.

Hurn, M. A., Husby, O. K. & Rue, H. (2003), A tutorial of image analysis, *in* J. Møller, ed., 'Spatial Statistics and Computational Methods', Springer.

Nocedal, J. & Wright, S. J. (2000), *Numerical Optimization*, Springer.

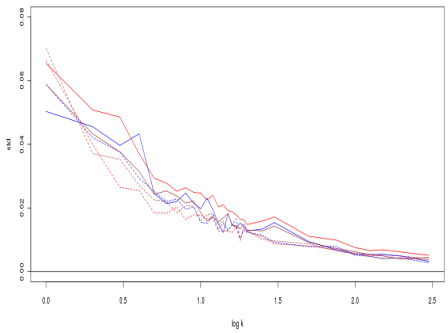
Propp, J. G. & Wilson, D. B. (1996), 'Exact sampling with coupled Markov chains and applications to statistical mechanics Carlo simulation', *Random Sructures Algorithms* **9**, 223–252.

Reeves, R. & Pettitt, A. N. (2002), 'Efficient recursions for general factorizable models', *Biometrika* **91**, 751–757.

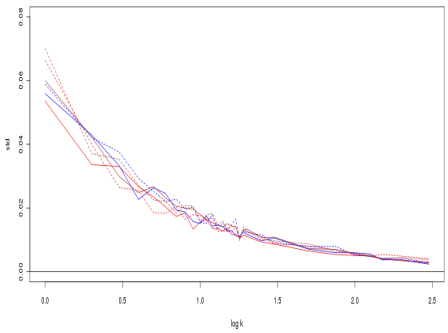
Scott, A. L. (2002), 'Bayesian methods for hidden Markov models: Recursive computation in the 21st century', *Journal of the American Statistical Association* **97**, 337–351.

Swendsen, R. & Wang, J. (1987), 'Nonuniversal critical dynamics in Monte Carlo simulation', *Physical Review letters* **58**, 86–88.

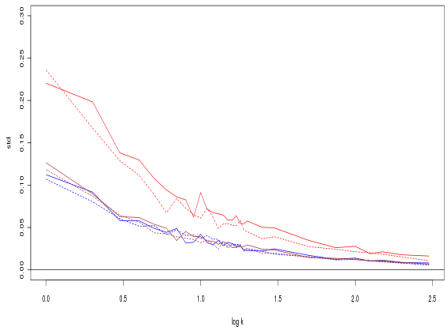
A Figures



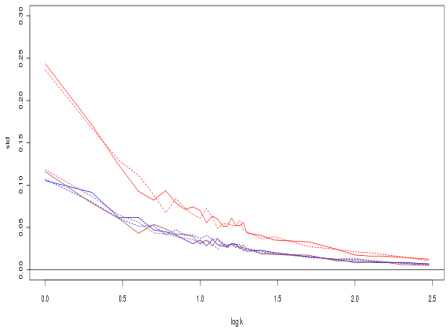
(a) $\alpha = 0.1, \beta_h = 0.83, \beta_h = 0.80$



(b) $\alpha = 0.1, \beta_h = 0.83, \beta_h = 0.80$

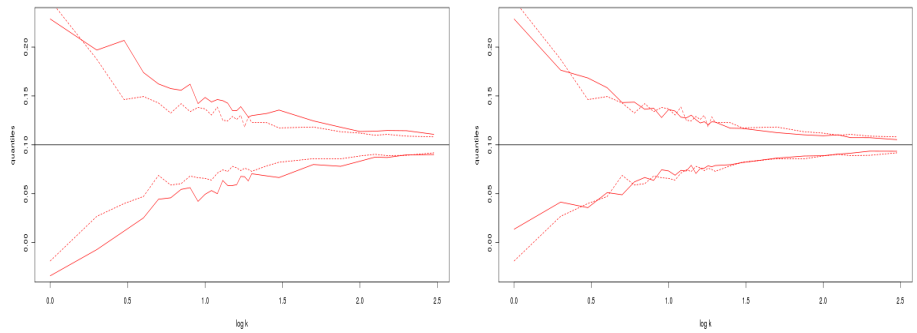


(c) $\alpha = 0.5, \beta_h = 0.83, \beta_h = 0.80$



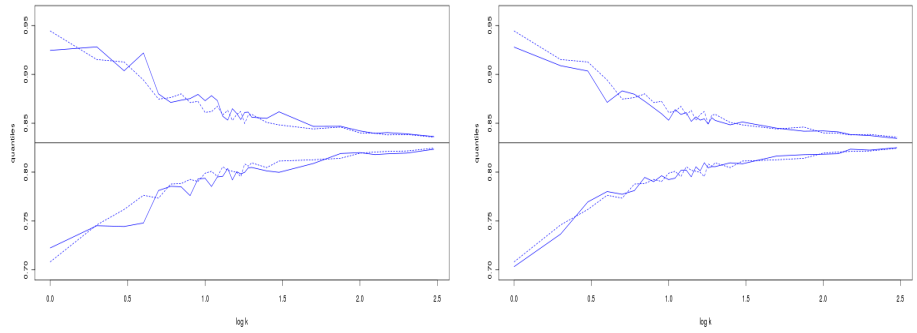
(d) $\alpha = 0.5, \beta_h = 0.83, \beta_h = 0.80$

Figure 37: The standard deviations of the parameter estimates. We use the same colour code and layout as in figure 11.



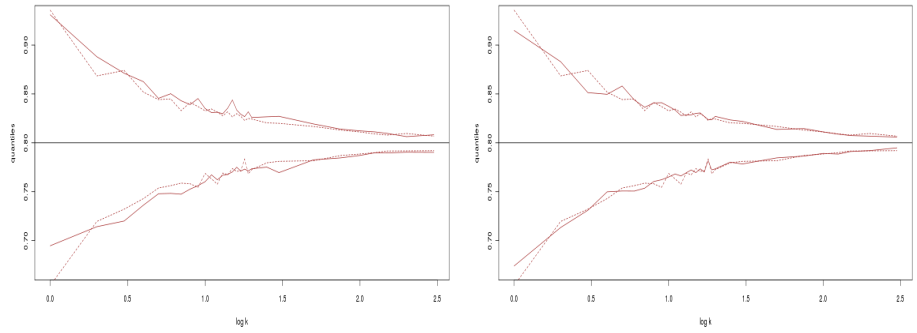
(a) $\alpha = 0.1$

(b) $\alpha = 0.1$



(c) $\beta_h = 0.83$

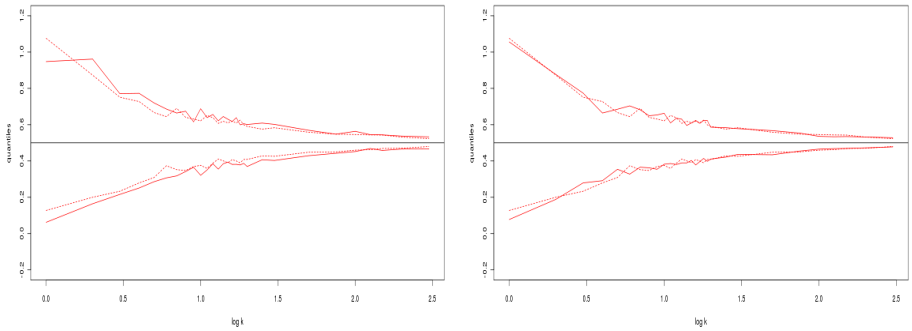
(d) $\beta_h = 0.83$



(e) $\beta_v = 0.80$

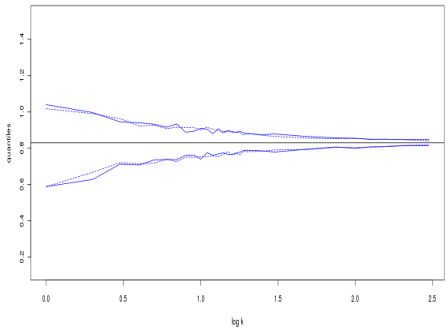
(f) $\beta_v = 0.80$

Figure 38: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

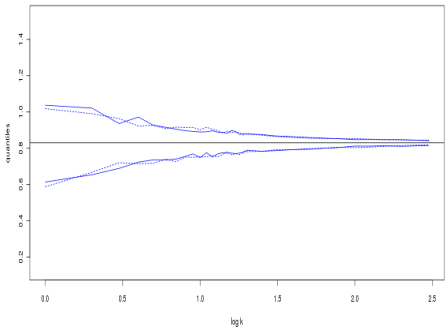


(a) $\alpha = 0.5$

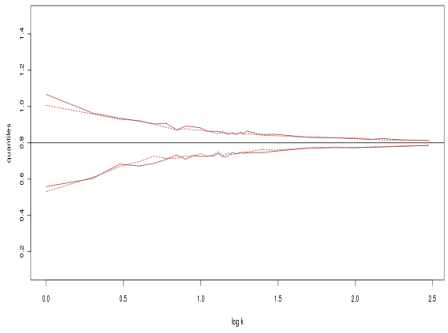
(b) $\alpha = 0.5$



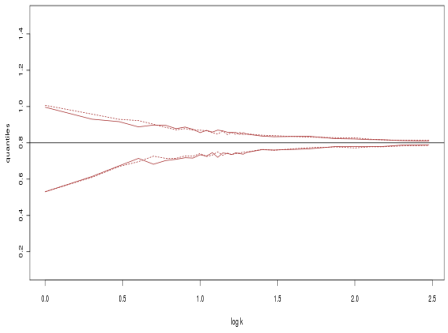
(c) $\beta_h = 0.83$



(d) $\beta_h = 0.83$

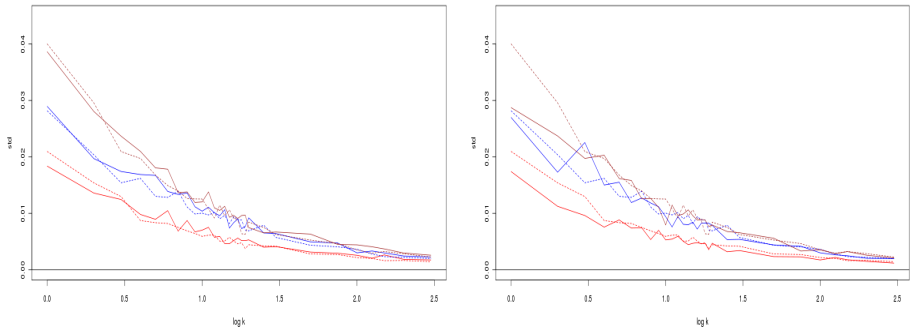


(e) $\beta_v = 0.80$



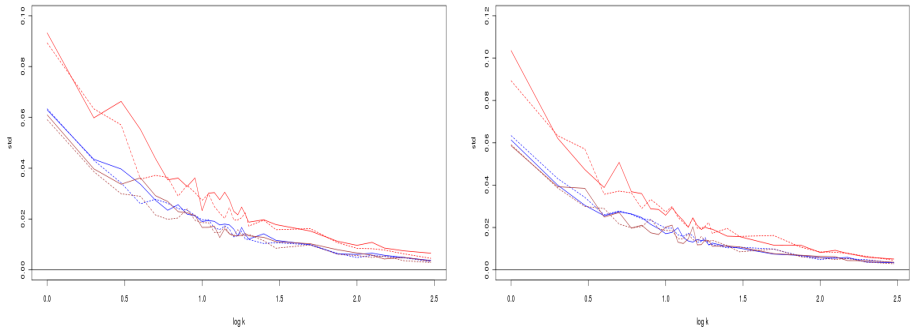
(f) $\beta_v = 0.80$

Figure 39: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.



(a) $\alpha = 0.1, \beta_h = 0.40, \beta_h = 0.80$

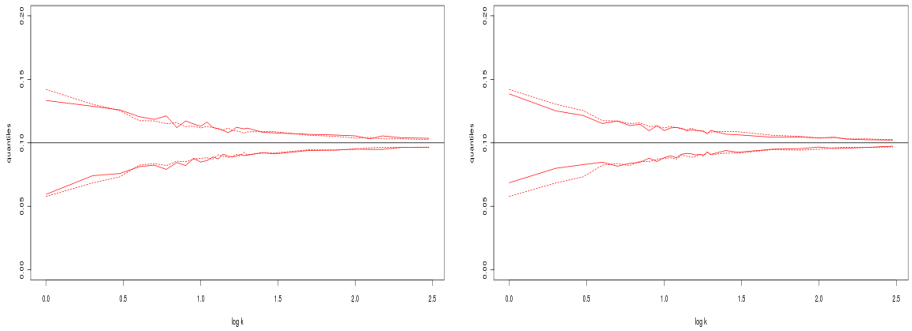
(b) $\alpha = 0.1, \beta_h = 0.40, \beta_h = 0.80$



(c) $\alpha = 0.5, \beta_h = 0.40, \beta_h = 0.80$

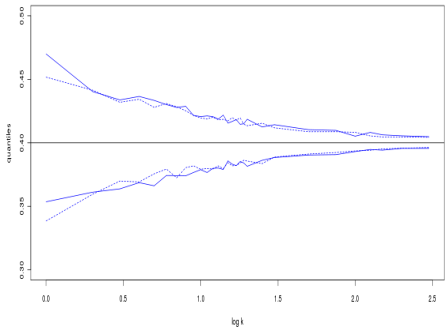
(d) $\alpha = 0.5, \beta_h = 0.40, \beta_h = 0.80$

Figure 40: The standard deviations of the parameter estimates. We use the same colour code and layout as in figure 11.

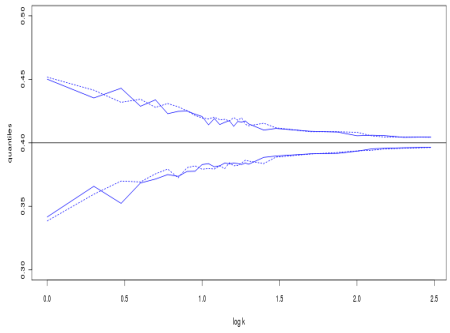


(a) $\alpha = 0.1$

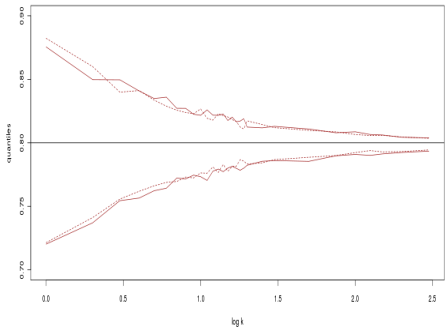
(b) $\alpha = 0.1$



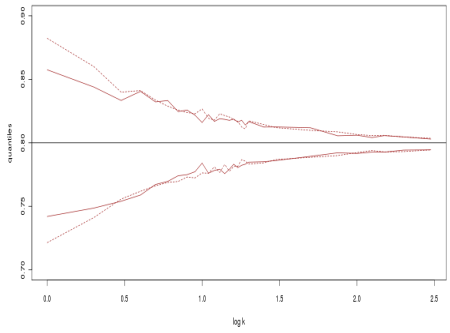
(c) $\beta_h = 0.40$



(d) $\beta_h = 0.40$

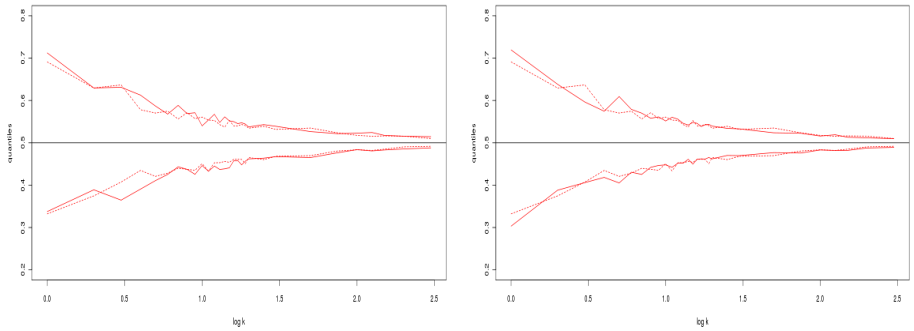


(e) $\beta_v = 0.80$



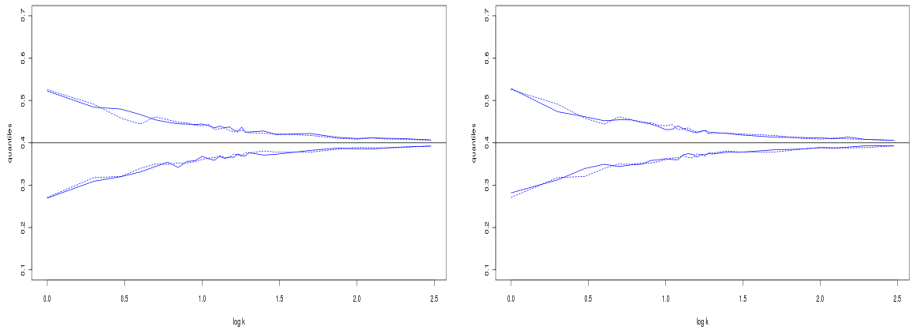
(f) $\beta_v = 0.80$

Figure 41: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.



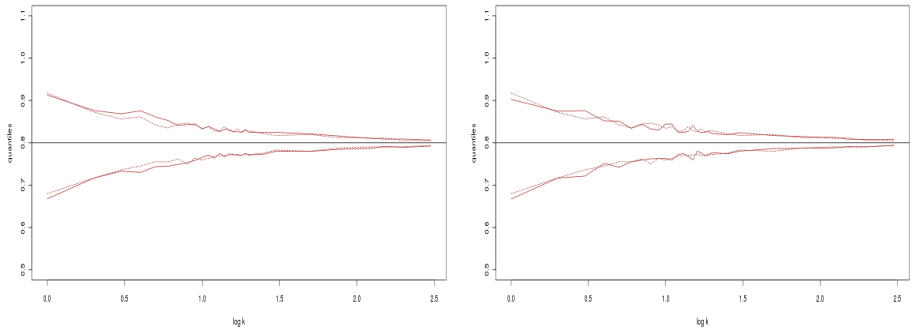
(a) $\alpha = 0.5$

(b) $\alpha = 0.5$



(c) $\beta_h = 0.40$

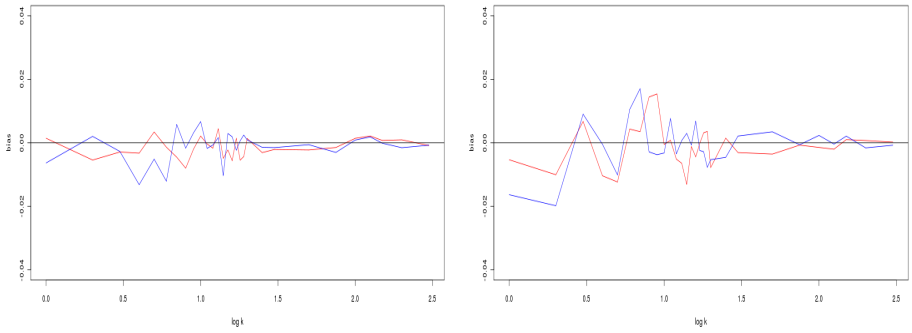
(d) $\beta_h = 0.40$



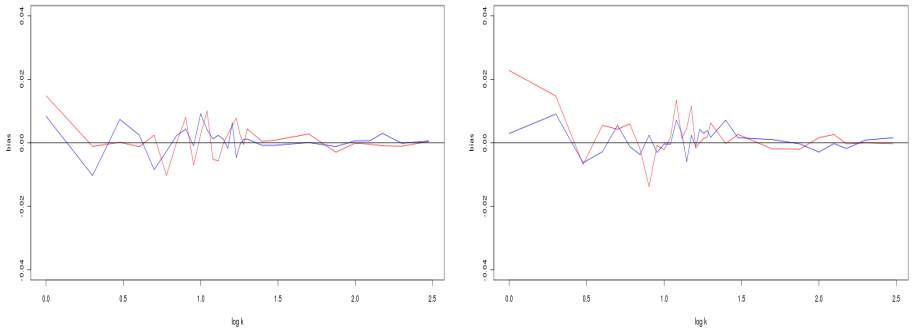
(e) $\beta_v = 0.80$

(f) $\beta_v = 0.80$

Figure 42: The quantiles of a 0.95 confidence interval. We use the same colour code and layout as in figure 11.

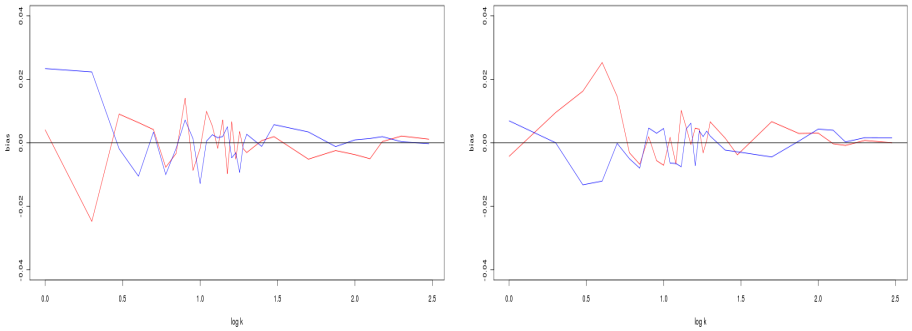


(a) θ_0 (b) θ_2

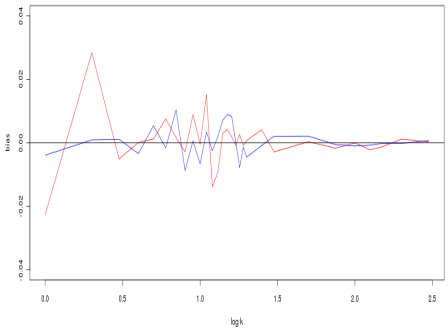


(c) θ_4 (d) θ_7

Figure 43: The red and blue lines are the biases when we use MPLE and 2×2 MGPLE, respectively.

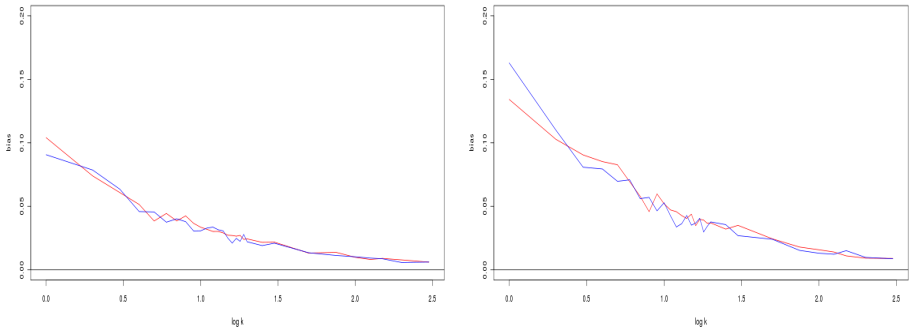


(a) θ_0 (b) θ_2

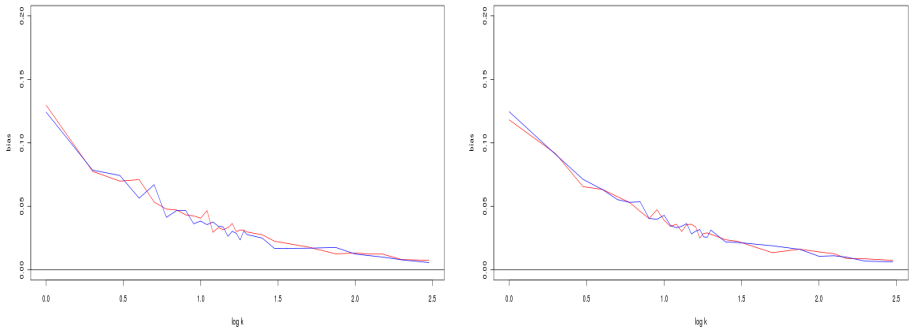


(c) θ_4

Figure 44: The red and blue lines are the biases when we use MPLE and 2×2 MGPLE, respectively.

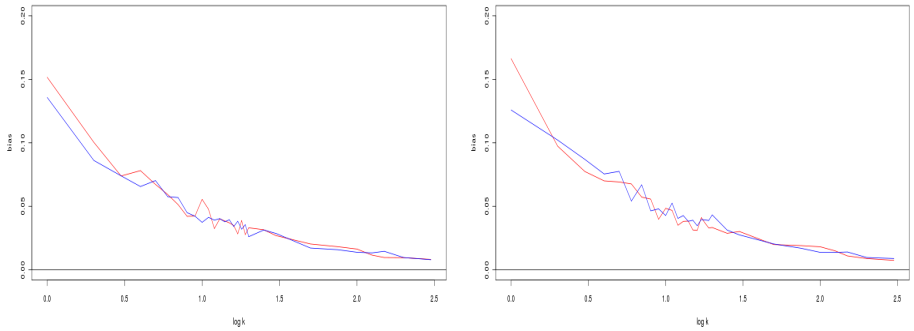


(a) θ_0 (b) θ_2

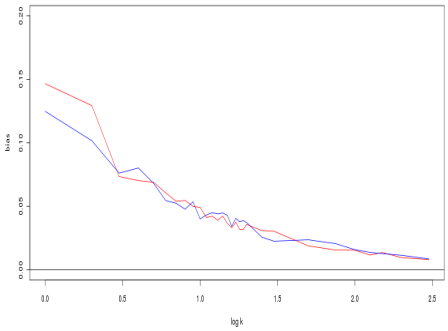


(c) θ_4 (d) θ_7

Figure 45: The red and blue lines are the standard deviations when we use MPLE and 2×2 MGPLE, respectively.

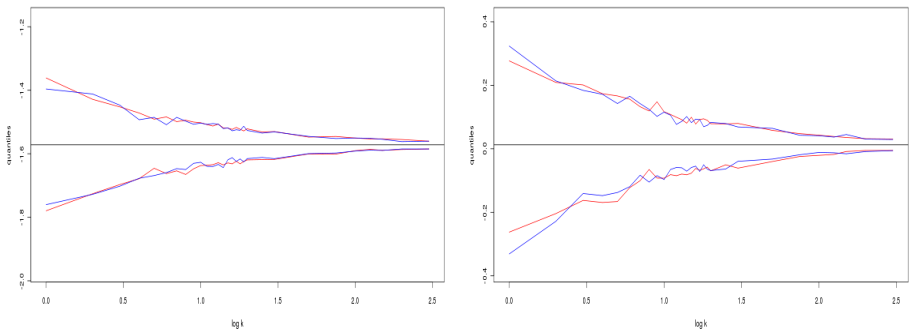


(a) θ_0 (b) θ_2

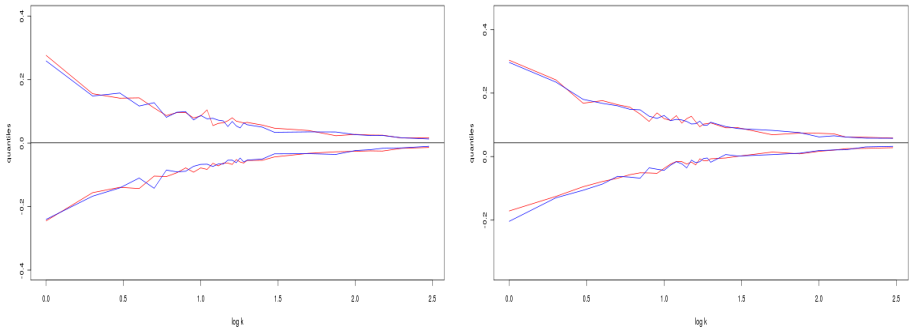


(c) θ_4

Figure 46: The red and blue lines are the standard deviations when we use MPLE and 2×2 MGPLE, respectively.

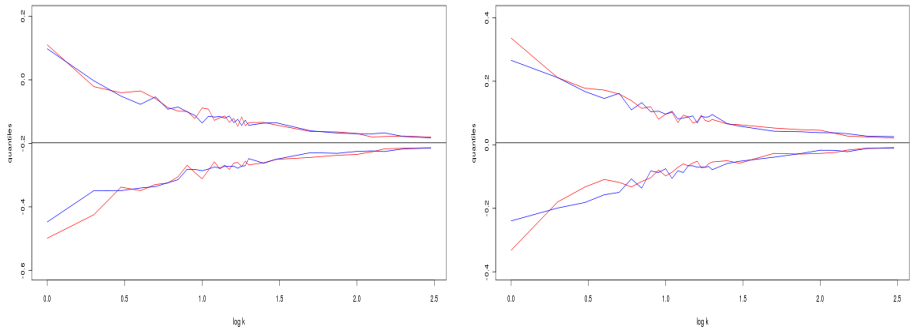


(a) θ_0 (b) θ_2



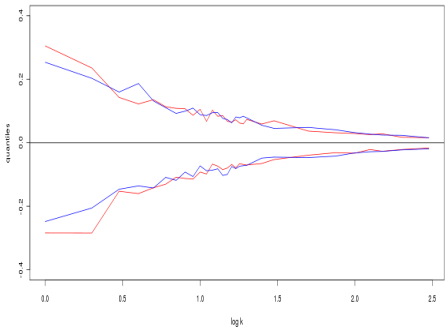
(c) θ_4 (d) θ_7

Figure 47: The lines are the upper and lower quantiles of a 0.95 confidence interval, and the black line is the value of the true parameter. The red and blue lines are the quantiles when we use MPLE and 2×2 MGPLE, respectively.



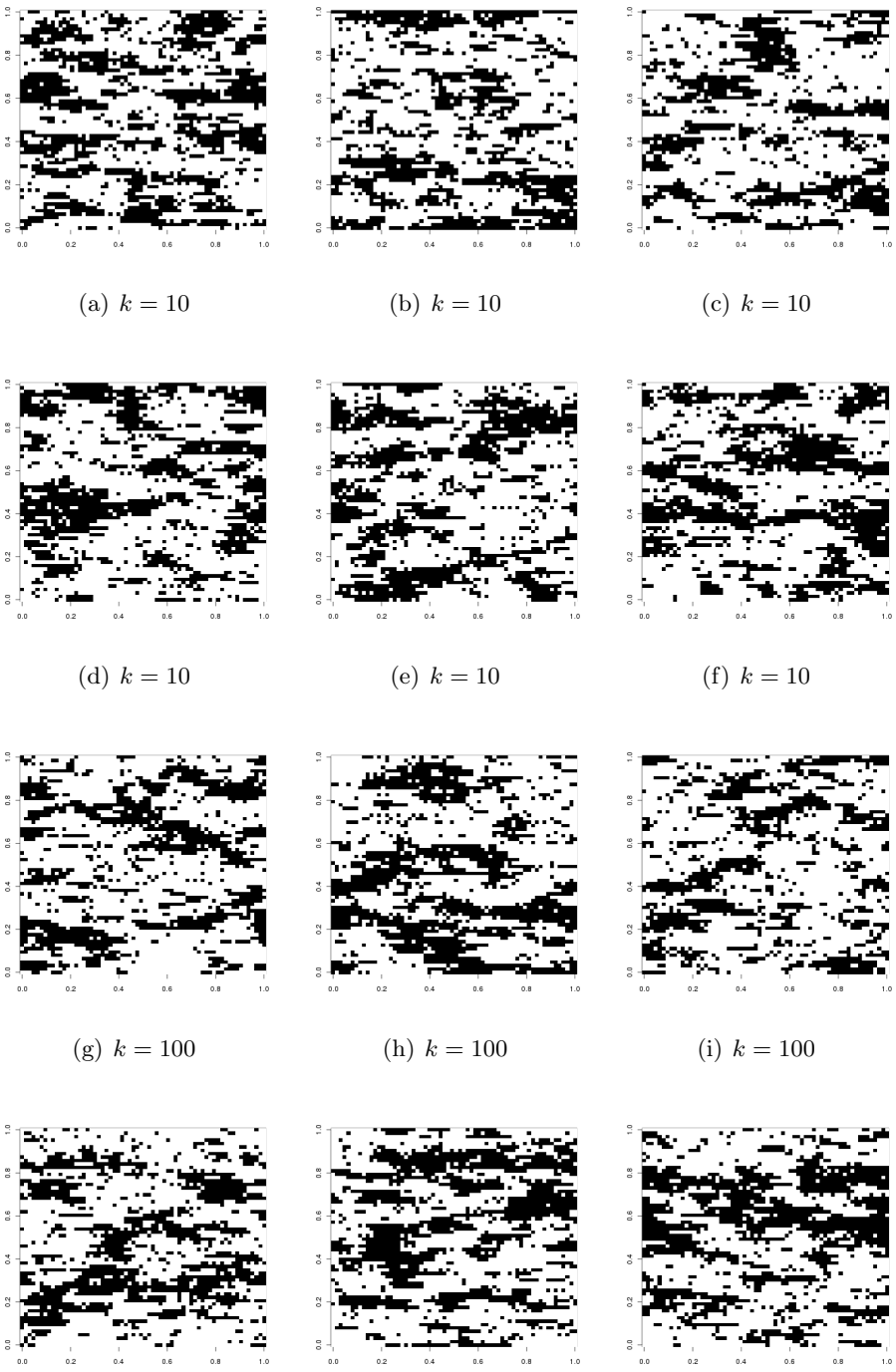
(a) θ_0

(b) θ_2



(c) θ_4

Figure 48: The lines are the upper and lower quantiles of a 0.95 confidence interval, and the black line is the value of the true parameter. The red and blue lines are the quantiles when we use MPLE and 2×2 MGPLE, respectively.



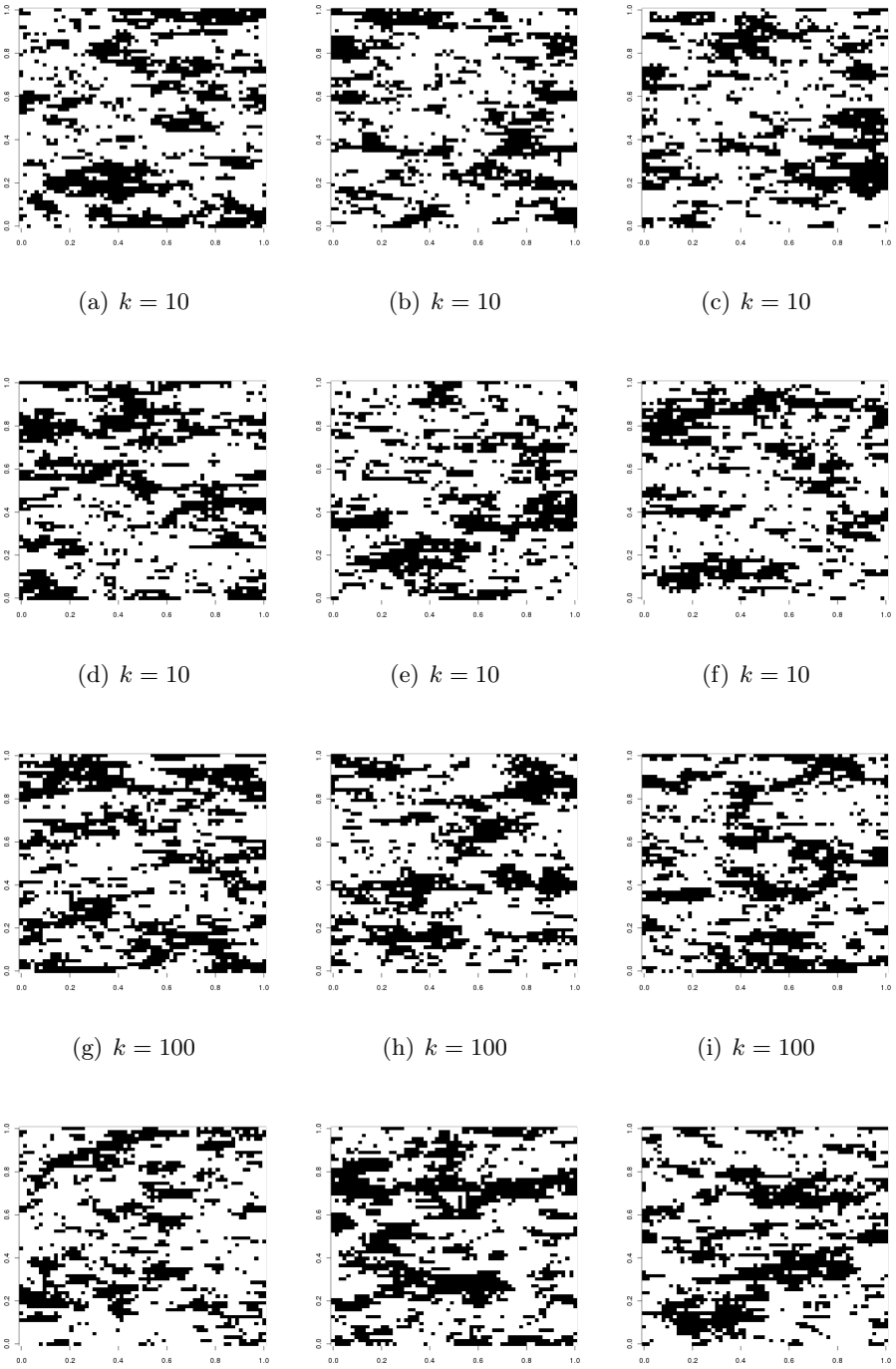
(a) $k = 10$ (b) $k = 10$ (c) $k = 10$

(d) $k = 10$ (e) $k = 10$ (f) $k = 10$

(g) $k = 100$ (h) $k = 100$ (i) $k = 100$

(j) $k = 100$ (k) $k = 100$ (l) $k = 100$

Figure 49: Realisations generated with parameters found with MPLE.



(a) $k = 10$ (b) $k = 10$ (c) $k = 10$

(d) $k = 10$ (e) $k = 10$ (f) $k = 10$

(g) $k = 100$ (h) $k = 100$ (i) $k = 100$

(j) $k = 100$ (k) $k = 100$ (l) $k = 100$

Figure 50: Realisations generated with parameters found with MGPLE.