Markov Chain Monte Carlo methodology for inference with generalised linear spatial models

Ioanna Lampaki

Submitted for the degree of Doctor of Philosophy

at Lancaster University,

August 2015

to my family

Declaration

I declare that this thesis entitled "MCMC methodology for inference on generalised linear spatial models" is the result of my own research under the guidance of Dr. Chris Sherlock except as otherwise cited in the references. This thesis has not been submitted in any form for the award of a higher degree elsewhere.

All the algorithms presented in Chapter 3 have been coded in R (R Core Team 2015) whereas the algorithms presented in Chapter 4 have been coded in the C programming language using the GNU Scientific Library (Galassi et al. 2010).

Ioanna Lampaki

Acknowledgements

This doctoral thesis was funded by the Faculty of Science and Technology of Lancaster University and many people have contributed to its completion in so many different ways. First and foremost, I would like to express my gratitude to my supervisor Chris Sherlock for his guidance and support. His help, intuition and excitement has been invaluable for the completion of this thesis. I would especially like to thank him for his patience, willingness and ability to always give answers to my never-ending questions. Working with him has been an invaluable experience and has taught me a lot. I am also indebted to him for proofreading this thesis and providing me with constructive comments. I would also like to thank Jonathan Tawn for his genuine interest and support.

I have been very lucky in sharing the same office for four years with some of the most wonderful people I have ever met. Special mention to Stephanie Wallace, Ross Towe, Karen Pye and my "neighbour" Simon Taylor who was always willing to help with any problem I had while I was learning to code in C. You have all been such a great company, supporters, office-mates and source of laugh and joy throughout these years. I would also like to thank Giorgos Sermaidis, Yanyun Wu and Ye Liu for all the good time we had.

My time in Lancaster would not have been the same without you. All of you are part of my best memories in Lancaster. A big thanks to Ioannis Papastathopoulos for his continuous encouragement and emotional support during this journey.

Lastly but definitely not least, the biggest thank you goes to my family, Vasilis, Tina, Liza and Eleni for their endless support and love. Also, to my adorable nieces, Agapi and Ioli for always making me eager to visit home and for making me understand that sometimes there might be more important things than this work out there. I wouldn't have made it without you. Thank you!

Abstract

Many real world phenomena are described through models that include an unobserved process which is usually characterised by a continuous distribution. Such models are widely used in geostatistics where a continuous spatial phenomenon is modelled through an underlying latent Gaussian process.

If the observed data are also Gaussian then inference for the underlying process and the model parameters is relatively straightforward. In many applications though the assumption of normally distributed data is not sensible and the assumption of Poisson or binomial data is more suitable. These models, with non-Gaussian data, are known as generalised linear spatial models (GLSM). In such cases, inference requires more sophisticated techniques and a common approach is the use of Markov chain Monte Carlo methods (MCMC). However, the correlation between the components of the latent process and the correlation between the latent process and the model parameters generally hinders the performance of any MCMC scheme which updates the latent process and the parameters sequentially.

In this thesis we focus on the Poisson GLSM and elaborate on the problem of the correlation within the latent process. In particular, our aim is to construct an efficient proposal distribution for sampling from the posterior distribution of the latent process conditionally on the other parameters. Initially, we investigate the idea of constructing a global normal approximation to the conditional posterior distribution of the latent process and use it as the proposal distribution in a simple and fast MCMC scheme. For this purpose, we initially employ various transformations of the data and find that some of the constructed schemes perform well in certain low dimensional scenarios. Subsequently, we construct one dimensional proposals for each component of the latent process through an approximation to each univariate marginal posterior conditional on a few principal components. The suggested MCMC scheme updates each component of the process separately and then proceeds by updating the few important principal components. As suggested by our results, this method has a stable and efficient performance in a variety of scenarios and dimensions.

Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Motivation

Geostatistics is a branch of spatial statistics focusing on the study of a continuous spatial phenomenon. Such phenomena could be the temperature, radioactivity or even the intensity of weed growth over a predefined area of study. This kind of phenomena can conceptually be described by a continuous stochastic process, which however, is not directly observed. Instead, what we have available is only a finite sample of observations of a random variable at specific locations over the study area. However, the locations of the data are not informative about the process we want to model. These observations, are usually assumed to be either identical to or a noisy version of the underlying true process or of a function of it. Interest usually lies in either predicting the realisation of the process at unsampled locations or making inference about the parameters of the model.

The typical modelling framework for such geostatistical problems is the generalised linear spatial model (GLSM). The GLSM is a generalised linear model but with an additional layer of stochasticity, the underlying latent process, in the linear predictor. The modelling of this process depends on the problem under study. It could for instance be modelled as a stationary or non-stationary Gaussian process, Gaussian Markov random field or we could even loosen the assumption of Gaussianity. In this Thesis we focus on the traditional GLSM as introduced in Diggle et al. (1998) where it is modelled through a stationary Gaussian process. The parameters involved in the model are of two types. Those used to model the trend and those used to model the spatial dependence, i.e., the covariance structure of the process. The GLSM gives the flexibility of modelling various types of data such as Poisson or binomial and also has as a special case the linear spatial model where the response variable is assumed to be normally distributed. In this thesis we deal with non-linear models and especially the Poisson GLSM.

Under the Bayesian framework, inference on the latent process and the parameters of the model relies on the use of MCMC methods since direct sampling from their joint posterior distribution is not possible; unless the data are Gaussian. Usually, in order to sample from this joint posterior an MCMC scheme will alternate between updating the parameters conditionally on the latent process and then updating the latent process conditionally on the current values of the parameters in the chain. Although such a scheme might appear straightforward to implement, it entails practical difficulties.

First of all, the latent process is usually of high dimension and this automatically hinders the performance of any MCMC scheme since mixing and convergence times increase with the dimension of the target; as also does their computational cost. Moreover, there is dependence between the latent process and the parameters and this will usually cause

the chain to converge and mix slowly. If the updating mechanism, say, for one of the parameters is not efficient this can also affect the mixing of the other components that are updated conditionally on that parameter.

Another challenging issue, is sampling from the posterior distribution of the latent process conditional on the parameters. The problem lies not only on the dependence of the latent process with the parameters but on the fact that the components of the process can be strongly correlated á posteriori. This posterior dependence in combination with the dimensionality of the process makes the construction of an efficient proposal challenging. Updating the latent process in blocks can lead to poor mixing for the same reasons as explained in the previous paragraph. On the other hand, updating the latent process in one step is also demanding since a proposal distribution that aattempts to match the shape of a correlated high dimensional target is hard to construct.

All parameters affect how strong the posterior correlation within the latent process will be but a very important contributing factor is the data and more specifically the amount of information they provide on the latent process. For instance, if the data are weak then the main contribution to the posterior distribution of the process will come from the prior, and the posterior dependence can therefore be strong. On the other hand, if the data are strongly informative then the likelihood will contribute more to the posterior distribution and the components of the latent process will be approximately independent *á posteriori* or at least less correlated than *á priori*. This is a key obstacle in constructing a generic MCMC scheme that would perform efficiently irrespective of the observed data.

Christensen et al. (2006) have suggested an MCMC scheme that attempts to provide a solution to all of the aforementioned problems. Their approach is based on a reparam-

eterisation of the process and the mean parameters so that these are all approximately independent with zero mean and unit variance. Additionally, these transformed random variables are also expected to be approximately independent of the parameters associated with the covariance structure of the process. They have provided a good, workable solution to the difficult problem of dependence between the latent process and the other parameters.

Finally, before closing this section we should mention that alternatives to MCMC methods for inference on GLSMs do exist such as the more recently introduced approach of integrated nested Laplace approximation (INLA) (Rue et al. 2009). INLA is a method of approximate Bayesian inference and unlike traditional MCMC it is a non-sampling based technique and provides approximate inference through a series of accurate Gaussian and Laplace approximations. Another fundamental difference between INLA and MCMC is that the former assumes that inferences are to be drawn on the marginal posterior distributions of each component of the latent process and these exactly are the posteriors that it attempts to approximate. The procedure followed in order to approximate these posterior marginals can be briefly summarised in the following steps. First of all, the joint posterior distribution of the model parameters is approximated using the Laplace approximation and subsequently the marginal posterior of each parameter is obtained by numerically integrating out the remaining model parameters. Then, the marginal posterior distribution of each component of the latent process given the model parameters is approximated. This is achieved using a series expansion that takes into account third order terms that account for the skewness. Finally, the model parameters are integrated out of the product of the two marginal approximations in order to obtain the marginal posterior of each component of the latent process. Great computational

gains are achieved when the underlying latent process has a Markov structure as illustrated in Rue et al. (2009). Although this is not required for the implementation of the suggested methodology, methods for approximating Gaussian processes through Gaussian Markov processes do exist Rue & Held (2005), Lindgren et al. (2011) and reduce the comptutauional complexity of the problem. For a comparative study between INLA and specfic MCMC methods we refer the reader to Taylor & Diggle (2012).

1.2 Scope and Outline

The focus of the present thesis is to provide an efficient MCMC scheme for inference on generalised linear spatial models. Since Christensen et al. (2006) have provided a framework for breaking the dependence between the process and the parameters we constrain our focus on the development of an efficient proposal for the latent process. Therefore, the methodology developed in Chapter 3 and Chapter 4 considers that all parameters in the model are fixed and interest lies only on sampling from the posterior distribution of the process given all the parameters. Although our focus is placed on the Poisson generalised linear spatial model, most of the proposed methodology could also be extended to the case of a Binomial generalised linear spatial model.

In Chapter 2 we provide material that is relevant and needed for the rest of the thesis. In Section 2.1 we outline some fundamental MCMC algorithms that will be used later on and discuss some issues related to their practical implementation, performance and efficiency. Section 2.2 introduces the reader to the area of geostatistics and the associated models. We describe the formulation and components of the linear spatial model along with the inferential procedure usually used and, in Section 2.2.6, extend these in the case of the generalised linear spatial model, which is the focus of this thesis. Finally, in

Section 2.2.7 we review some of the MCMC schemes that have been suggested in the literature for inference on the generalised linear spatial model.

In Chapter 3 we focus on constructing a single global normal approximation to the posterior density of interest. In particular, we explore the idea of applying a normal approximation to the conditional density of a transformation of the data. This enables us to work under the framework of the linear Gaussian model, where the form of the posterior is tractable. In that way we are able to find an approximate posterior distribution for the latent variables and use this as a proposal in a simple, fast and straightforward-to-implement MCMC scheme.

In Chapter 4 we employ concepts of multivariate analysis and rather than constructing a global approximation to the target we use an approximation to each univariate marginal posterior conditional on a few principal components. In that way, we develop a two-stage proposal mechanism where each component of the process is updated separately and subsequently, the mixing of the algorithm is improved by additionally updating the few important principal components. This results in an efficient algorithm with stable performance across different datasets and dimensions.

Both in Chapter 3 and Chapter 4 the efficiency of the constructed algorithms is assessed and compared against existing schemes through extensive simulation studies.

CHAPTER 2

Background material

2.1 A review of Markov chain Monte Carlo algorithms

Markov chain Monte Carlo (MCMC) methods constitute a unified framework which enables sampling from complicated and analytically intractable distributions. This is achieved by constructing an ergodic Markov chain with stationary distribution identical to the distribution of interest. In our case this stationary distribution will be the posterior distribution, $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, of a d-dimensional vector of continuous random variables $\boldsymbol{\theta}$, given the observed data, $\boldsymbol{y}$. Once a sufficiently large sample is obtained, Monte Carlo estimates of any functional of $\boldsymbol{\theta}$, that we are interested in, can be obtained.

This section introduces the reader to the MCMC algorithms that are used in this thesis, namely the Metropolis-Hasting (MH), Random Walk Metropolis (RWM), Metropolis Adjusted Langevin Algorithm (MALA) and Riemann manifold MALA (MMALA) algorithm. In particular, we begin by setting out the basic MH algorithm and outline how the

rest are special cases of it. Theoretical properties and technical conditions for the convergence of these algorithms have been investigated over the last 20 years. However, we are mainly interested in their practical implementation and therefore focus on providing an intuitive interpretation of each algorithm and aspects that define their performance. Unless otherwise stated, the two main sources of this section are Gilks et al. (1996) and Gamerman & Lopes (2006).

2.1.1 Metropolis-Hastings algorithm (MH)

The MH algorithm, Metropolis et al. (1953) and Hastings (1970), uses an appropriate transition density such that the constructed Markov chain converges to the desired stationary distribution. Let $g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ denote the probability (density) of moving to $\boldsymbol{\theta}^*$ given that we are at $\boldsymbol{\theta}$. A sufficient condition for a Markov chain to have a stationary distribution, $\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\boldsymbol{y})$ is, for $g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ to satisfy the detailed balance,

$$\pi(\boldsymbol{\theta})g(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \pi(\boldsymbol{\theta}^*)g(\boldsymbol{\theta}^*, \boldsymbol{\theta}).$$

This condition is equivalent to the statement that, at stationarity, the chance of being at $\boldsymbol{\theta}$ and moving to $\boldsymbol{\theta}^*$ is the same as the chance of being at $\boldsymbol{\theta}^*$ and moving to $\boldsymbol{\theta}$. The MH algorithm proposes the chain to move to a new state $\boldsymbol{\theta}^*$ which is generated from a proposal distribution $q(\boldsymbol{\theta}^*|\boldsymbol{\theta})$. However, the proposed value is not always accepted, but is accepted according to some probability $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$. This acceptance rate is chosen such that the probability (density) of moving to $\boldsymbol{\theta}^*$ given that we are in $\boldsymbol{\theta}$, where $\boldsymbol{\theta}^* \neq \boldsymbol{\theta}$, is,

$$g(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*), \tag{2.1.1}$$

whereas the probability (mass) of proposing a value and rejecting it so that we stay at $\boldsymbol{\theta}$; is given by

$$g(\boldsymbol{\theta}, \boldsymbol{\theta}) = 1 - \int q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)d\boldsymbol{\theta}^*.$$

In general, the probability of moving to any set C given that we are at $\boldsymbol{\theta}$ is given by

$$g(\boldsymbol{\theta}, C) = \int_C q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)d\boldsymbol{\theta}^* + I(\boldsymbol{\theta} \in C)\left\{1 - \int q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)d\boldsymbol{\theta}^*\right\}.$$

The so called acceptance ratio $\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)$ is defined in such a way that when combined with the transition kernel gives a chain satisfying the detailed balance equation. In particular it is set to be

$$\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right), \tag{2.1.2}$$

ensuring that the chain has as stationary distribution $\pi(\boldsymbol{\theta}) = \pi(\boldsymbol{\theta}|\boldsymbol{y})$.

Proposition 2.1.1. *The Metropolis Hastings algorithm satisfies the detailed balance equations.*

Proof.

$$\begin{aligned}
g(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\pi(\boldsymbol{\theta}) &= q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\alpha(\boldsymbol{\theta}, \boldsymbol{\theta}^*)\pi(\boldsymbol{\theta}) \\
&= q(\boldsymbol{\theta}^*|\boldsymbol{\theta})\min\left(1, \frac{\pi(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}\right)\pi(\boldsymbol{\theta}) \\
&= \min\left(\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta}), \pi(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)\right) \\
&= \min\left(\frac{\pi(\boldsymbol{\theta})q(\boldsymbol{\theta}^*|\boldsymbol{\theta})}{\pi(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)}, 1\right)\pi(\boldsymbol{\theta}^*)q(\boldsymbol{\theta}|\boldsymbol{\theta}^*) \\
&= q(\boldsymbol{\theta}|\boldsymbol{\theta}^*)\alpha(\boldsymbol{\theta}^*, \boldsymbol{\theta})\pi(\boldsymbol{\theta}^*) \\
&= g(\boldsymbol{\theta}^*, \boldsymbol{\theta})\pi(\boldsymbol{\theta}^*)
\end{aligned}$$

□

In practice, in order to draw samples from $\pi(\boldsymbol{\theta}|\boldsymbol{y})$ the MH algorithm proceeds as follows:

- Initialise $\boldsymbol{\theta} = \boldsymbol{\theta}^0$
- Draw $\boldsymbol{\theta}^{prop}$ from a density $q(\boldsymbol{\theta}^{prop}|\boldsymbol{\theta}^{cur})$
- Calculate the acceptance probability

$$\alpha(\boldsymbol{\theta}^{cur}, \boldsymbol{\theta}^{prop}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{prop})q(\boldsymbol{\theta}^{cur}|\boldsymbol{\theta}^{prop})}{\pi(\boldsymbol{\theta}^{cur})q(\boldsymbol{\theta}^{prop}|\boldsymbol{\theta}^{cur})}\right) \tag{2.1.3}$$

- Draw $u \sim \mathrm{U}[0,1]$
- If $u \leq \alpha(\boldsymbol{\theta}^{cur}, \boldsymbol{\theta}^{prop})$, set $\boldsymbol{\theta}^{cur} = \boldsymbol{\theta}^{prop}$, else keep $\boldsymbol{\theta}^{cur}$ unchanged.
- Whether or not the proposal was accepted store $\boldsymbol{\theta}^{cur}$ as the next element of the chain

From (2.1.3), it is easy to notice that any proposed values outside the support of π will have an acceptance ratio of 0, since $\pi(\boldsymbol{\theta}^{prop}) = 0$,

$$\frac{\pi(\boldsymbol{\theta}^{prop})q(\boldsymbol{\theta}^{cur}|\boldsymbol{\theta}^{prop})}{\pi(\boldsymbol{\theta}^{cur})q(\boldsymbol{\theta}^{prop}|\boldsymbol{\theta}^{cur})} = 0,$$

and will therefore be rejected. Hence, ideally we want to use a proposal such that $\text{support}(q(\cdot|\boldsymbol{\theta})) \subseteq \text{support}\,(\pi)$. This however, might sometimes be computational infeasible in practice especially for complicated high dimensional targets constrained in subsets of $\mathbb{R}$. In such cases, one would use a proposal with support greater than that of the target and use rejection sampling in order to only propose sensible values of $\boldsymbol{\theta}$ (see algorithm A1 in Section 3.2.2 for such a practice).

Finally, the ergodicity of the resulting Markov chain can be guaranteed if it is π-irreducible and aperiodic. The property of π-irreducibilty ensures that the whole support

of the taregt, π, can be explored in a finite number of transition steps. More formally,

Definition 2.1.2. *A Markov chain is π-irreduciple if for every $\theta \in \Theta$ there exists $n \in \mathbb{Z}_+$ such that $g^n(\theta, C) > 0$ for all subsets $C \subseteq \Theta$ where $\pi(C) > 0$.*

However, even if the chain is π-irreducible convergence to π might not be guaranteed. To overcome such an issue we also need the property of aperiodicity. Aperiodicity ensures that chain does not have any cyclic behaviour. In particular,

Definition 2.1.3. *A Markov chain with stationary distribution π is called aperiodid if there do no exist disjoint subsets of $\boldsymbol{\Theta}$, $\Theta_1, ..., \Theta_N$, for $N \leq 2$, with $g(\theta, \Theta_{i+1}) = 1$ for all $\theta \in \Theta_i$ and also $g(\theta, \Theta_1) = 1$ for every $\theta \in \Theta_N$.*

2.1.2 Efficient MCMC: convergence and mixing

The key conditions in order to draw valid inference based on samples drawn from a constructed using MCMC is that the chain has converged to $\pi(\cdot)$, in total variation sense (see Definition 2.1.4), and has also adequately explored the support of the distribution. These two properties, namely convergence and mixing, are the ones that determine the efficiency of an MCMC scheme.

Consider that we are interested in estimating the expectation of some real-valued function $r(\boldsymbol{\theta})$ under $\pi(\cdot)$, i.e., $r := \mathbb{E}_\pi[r(\boldsymbol{\theta})]$, using the output of an MCMC scheme that was run for n iterations. Since in practice the chain is unlikely to start in stationarity, it will require a certain number of iterations, c, until it has effectively converged and is producing samples from π. Including the initial c draws in the estimation of r would bias the estimate $\hat{r}_n$. It is common practice, therefore, to base inference on the last $n - c$

samples and estimate r through,

$$\hat{r}_n = \frac{1}{n-c} \sum_{i=c+1}^{n} r(\boldsymbol{\theta}_i)$$

This procedure of discarding an initial number of iterations is known as burn-in. Apart from visual inspection of traceplots of the chain there are many diagnostics in the literature for assessing whether a chain has converged and, if so, the number of iterations that were needed for convergence to be achieved; being therefore a very useful tool for defining an appropriate burn-in period. There are available both single chain diagnostics such as the ones proposed by Geweke (1992) and Raftery & Lewis (1992) and multiple chain diagnostics such as the diagnostic of Gelman & Rubin (1992) and its multivariate extension of Brooks & Gelman (1998*a*). All of these diagnostics, assess whether the distribution of either, parts of the same chain or two different chains are similar or not by comparing either the first two moments or a certain set of quantiles of the empirical distribution of the chains.

Mixing on the other hand relates to the dependence between the samples drawn under $\pi(\cdot)$. If the samples drawn are strongly dependent then the chain will be slowly mixing, meaning that the process will move slowly. As such, longer runs would be needed to adequately explore the target distribution and provide accurate estimates, $\hat{r}_n$, with standard errors equivalent to those obtained had the samples been drawn independently from π. In particular, if we set $N = n - c$ and denote by $\boldsymbol{\theta}_0$ the first sample drawn after the burn-in, then, under stationarity, the variance of $\hat{r}_N$ is given by,

$$v_r = \frac{N}{\text{ESS}} \mathbb{V}\text{ar}[r(\boldsymbol{\theta}_i)],$$

where

$$\text{ESS} := \frac{N}{1 + 2\sum_{i=1}^{l} \text{Cor}(r(\boldsymbol{\theta}_0), r(\boldsymbol{\theta}_i))} \tag{2.1.4}$$

and l is the first time that $\text{Cor}(r(\boldsymbol{\theta}_i), r(\boldsymbol{\theta}_{i+l}))$ falls below some predefined level, so as to be considered negligible. The denominator of (2.1.4) is the estimated integrated autocorrelation time, and ESS represents the number of independent samples to which the N drawn dependent samples are equivalent; in the sense of providing estimates with similar standard errors.

A certain limit theorem holds for any Markov chain that converges to π at a geometric rate; such chains are called geometrically ergodic.

Definition 2.1.4. *Let the distribution of the chain, started at an initial point $\boldsymbol{\theta}$, after n iterations be $g^n(\boldsymbol{\theta}, \cdot)$. An ergodic Markov chain, is geometrically ergodic with stationary distribution π if there exist a positive constant $b < 1$ and a real valued function M such that,*

$$\|g^n(\boldsymbol{\theta}, \cdot) - \pi(\cdot)\| \leq M(\boldsymbol{\theta})b^n,$$

$\forall \boldsymbol{\theta}, n \in \mathbb{Z}^+$ and where $||\cdot||$ denotes the total variation distance. For two densities m_1, m_2 on E, the total variation distance is defined as, $\|m_1 - m_2\| := \sup_{A \subset E} |m_1(A) - m_2(A)|$. Geometrically ergodic chains satisfy the following central limit theorem,

$$\hat{r}_N \xrightarrow{d} \text{normal}\left(\mathbb{E}_\pi[r(\boldsymbol{\theta})], N^{1/2} v_r\right).$$

Ideally we would like to have a sampler that converges quickly to the stationary distribution of interest and also mixes fast. In practice mixing is usually assessed by visual inspection of traceplots, autocorrelation plots and estimation of ESS.

2.1.3 Random walk Metropolis (RWM)

A special case of the MH algorithm is the Random Walk Metropolis, as introduced in Metropolis et al. (1953) where the proposal distribution $q$is symmetric and centred on the current value of $\boldsymbol{\theta}$. In this case, the only difference in the above algorithm is the simplification of the acceptance ratio to $\alpha(\boldsymbol{\theta}^{cur}, \boldsymbol{\theta}^{prop}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{prop})}{\pi(\boldsymbol{\theta}^{cur})}\right)$, since, due to the symmetry of q, $q(\boldsymbol{\theta}^{cur}|\boldsymbol{\theta}^{prop}) = q(\boldsymbol{\theta}^{prop}|\boldsymbol{\theta}^{cur})$. A widely used proposal for the RWM that satisfies the above is the normal distribution where the proposed jumps $(\boldsymbol{\theta}^{prop} - \boldsymbol{\theta}^{cur})$ are normally distributed with mean 0, i.e.,

$$\boldsymbol{\theta}^{prop} \sim \text{MVN}(\boldsymbol{\theta}^{cur}, \lambda I), \tag{2.1.5}$$

where the proposal variance, λ, is called the 'tuning parameter' and I denotes the identity matrix. In this case the RWM algorithm proposes a value from the above proposal and this is always accepted if this move is uphill, i.e., if the proposal has a higher posterior density than the current one; it is accepted with probability $\alpha(\boldsymbol{\theta}^{cur}, \boldsymbol{\theta}^{prop})$ otherwise.

For kernels of the form (2.1.5), the choice of λ is of vital importance since it determines how large or small the jumps from $\boldsymbol{\theta}^{\boldsymbol{cur}}$ to $\boldsymbol{\theta}^{prop}$ will be. If λ is too small then the proposed and current values are going to be too similar leading to high acceptance rates but also to high autocorrelations in the sample since accepted moves will be very close together. On the other hand, large values of λ would result in low acceptance rates and high autocorrelation since the chain rarely moves. In both cases the chain needs a large number of iterations to explore the target. It has been shown (Roberts et al. 1997) that in certain scenarios the optimal acceptance rate for the RWM is between 20% and 30%, and uses a tuning, $\lambda \propto d^{-1}$ where d is the dimension of the target distribution $\pi(\boldsymbol{\theta})$.

Finally, in the above we have oulined the simplest version of RWM with common variance λ for all d components of $\boldsymbol{\theta}$ and no correlation structure. This however in practice is quite unrealistic in many cases. We discuss how this can be overcome in Section 2.1.5.

2.1.4 Metropolis adjusted Langevin algorithm (MALA)

The RWM is a local algorithm in that the proposed value is in some sense close to the current value. This closeness is quantified by $\sqrt{\lambda}$. In what follows, we outline the Metropolis Adjusted Langevin Algorithm (MALA) which constitutes a more sophisticated algorithm and it is an extension of the RWM. For a more detailed description we refer the reader to Roberts & Tweedie (1996), Roberts & Rosenthal (1998) and Roberts & Rosenthal (2001).

It appears sensible to try encourage the chain to propose values with higher posterior density so as to achieve higher acceptance rates.

A Langevin diffusion process $\boldsymbol{\theta}_t$, for a density $\pi(\boldsymbol{\theta})$, evolves according to the following stochastic differential equation,

$$d\boldsymbol{\theta}_t = \frac{1}{2}\nabla \log \pi(\boldsymbol{\theta}_t)dt + d\boldsymbol{B}_t, \tag{2.1.6}$$

with $\boldsymbol{B}_t$ denoting a d-dimensional Brownian motion. Under certain technical conditions (see references above), this has as asymptotic distribution, $\pi(\boldsymbol{\theta}_t)$, as $t \to \infty$. The discrete analogue of the above process can be written as,

$$\boldsymbol{\theta}_t = \boldsymbol{\theta}_{t-1} + \frac{\lambda}{2}\nabla \log \pi(\boldsymbol{\theta}_{t-1}) + \lambda^{1/2} Z,$$

where $Z \sim \mathrm{N}(\mathbf{0}, I)$ and therefore

$$\boldsymbol{\theta}_t \sim \mathrm{MVN}\left(\boldsymbol{\theta}_{t-1} + \frac{\lambda}{2}\nabla \log \pi(\boldsymbol{\theta}_{t-1}), \lambda I\right).$$

Hence, if we use this as the proposal mechanism (with $\boldsymbol{\theta}_t = \boldsymbol{\theta}^{prop}$ and $\boldsymbol{\theta}_{t-1} = \boldsymbol{\theta}^{cur}$) we might suspect that convergence to the target $\pi(\boldsymbol{\theta})$ is faster than the usual RWM. As we see, this proposal incorporates gradient information in the mean and for that reason it tends to move the chain to regions of higher posterior density. In that way, we encourage the chain to move towards the nearest posterior mode and stay in the main posterior mass of the distribution. As with the RWM, the parameter λ defines the size of the proposed jumps.

Since now the shape of the proposal is closer to that of the target in conrast to the RWM, fewer proposals will be rejected and for that reason the optimal acceptance probability, $\alpha(\boldsymbol{\theta}^{cur}, \boldsymbol{\theta}^{prop})$, for MALA proposals is around $40\% - 60\%$. In particular, Roberts & Rosenthal (1998) showed that for target distributions consisting of iid components the optimal acceptance rate is close to 57.4% and is achieved for values of $\lambda \propto d^{-1/3}$. Summarising, as $d \to \infty$ the MALA algorithm results in higher optimal proposal variances along with higher optimal acceptance rates and therefore better mixing properties.

Nonetheless, besides these advantages of MALA a word of caution is needed if the target has tails as light as or lighter than those of a Gaussian density. In this case, when in the tails of the posterior, the magnitude of the gradient of the log posterior can be so large that the subsequenr proposal will be even further out in the tails. This could lead to not exploring the main body. For instance, let $\pi(\theta|\boldsymbol{y}) \propto e^{-\theta^4/4}$ leading to $\nabla \log(\pi(\theta|\boldsymbol{y})) = -\theta^3$. Then for large values of θ, the magnitude of the gradient will

be even larger, compared to θ, having as a result a great increase in the mean of the proposal distribution so that the proposed value will usually lie in the tails of the target. In order to tackle this problem, Roberts & Tweedie (1996) describe the truncated MALA where they place an upper bound on the magnitude of the gradient term in the mean of the proposal.

Furthermore, the MALA algorithm is likely to perform worse than the RWM when dealing with multimodal targets. If the current position of the chain is close to one mode, the MALA will tend to move the chain towards that mode and keep it always there. Therefore, it is possible to stay trapped in a particular mode for many iterations. Given that in practice the algorithm is run for a finite number of iterations it is possible that the drawn posterior samples would not represent the true target leading to wrong inferences.

2.1.5 Preconditioned MALA and RWM

So far, we have assumed the use of a constant tuning, λ, for all the components of $\boldsymbol{\theta}$ and no covariance structure. However, in practice this could be quite unrealistic since each component can have different variance and there may also exist correlations. Therefore, principle components of $\boldsymbol{\theta}$ with smallest variance will be mixing well whereas those with larger variances, will be mixing poorly. So it would be better to use a covariance matrix where the diagonal elements need not be the same, and if correlations are present in the posterior then the off-diagonal elements would not be zero. This technique is known as preconditioning.

Roberts & Stramer (2003) introduce the preconditioned MALA where the proposal takes

the form,

$$\boldsymbol{\theta}_t \sim \text{MVN}\left(\boldsymbol{\theta}_{t-1} + \frac{\lambda}{2}\boldsymbol{M}\nabla \log \pi(\boldsymbol{\theta}_{t-1}), \lambda \boldsymbol{M}\right). \tag{2.1.7}$$

The same approach can be applied in the case of the RWM, e.g. Sherlock et al. (2010). The corresponding preconditioned RWM uses the following proposal, distribution,

$$\boldsymbol{\theta}_t \sim \text{MVN}\left(\boldsymbol{\theta}_{t-1}, \lambda \boldsymbol{M}\right).$$

The main question though, is how to choose this covariance matrix. Ideally, we would like to have a proposal distribution which mimics the target distribution in the sense of having similar curvature. One approach for finding a suitable covariance matrix for our proposal would be to run a simple MALA/RWM algorithm with only a constant tuning for a fixed number of iterations, estimate the covariance matrix from the drawn posterior samples and use this matrix as $\boldsymbol{M}$ for the preconditioned MALA/RWM. However, this approach is quite empirical as the shape of the target can depend on the current position. Additionally, more than a few iterations may be needed to obtain a covariance matrix close to the true one.

2.1.6 Manifold MALA and simplified manifold MALA

Let the likelihood of $\boldsymbol{\theta}$ be $L(\boldsymbol{\theta})$ and let the prior distribution of $\boldsymbol{\theta}$ be $p(\boldsymbol{\theta})$. It is known from likelihood theory that, asymptotically, a consistent estimate for the covariance matrix of $\boldsymbol{\theta}$ is the inverse of the expected Fisher information matrix,

$$-E_{\boldsymbol{y}|\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial \boldsymbol{\theta} \partial \boldsymbol{\theta}'} \log\left\{L(\boldsymbol{\theta})\right\}\right]^{-1}.$$

Subsequently, taking the equivalent measure for the posterior density of $\boldsymbol{\theta}$, $\pi(\boldsymbol{\theta}|\boldsymbol{y})$, results in a consistent estimate of the posterior covariance matrix of $\boldsymbol{\theta}$ and is given by

$$\boldsymbol{M}(\boldsymbol{\theta}) = -E_{\boldsymbol{y}|\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial\boldsymbol{\theta}\partial\boldsymbol{\theta}'}\log\{\pi(\boldsymbol{\theta}|\boldsymbol{y})\}\right]^{-1},$$

The idea of using $\boldsymbol{M}(\boldsymbol{\theta})$ as the preconditioning matrix in the MALA proposal (2.1.7) is exploited in Girolami & Calderhead (2011) and the resulting algorithm is known as simplified manifold MALA (sMMALA).

Girolami & Calderhead (2011), employ concepts of Riemann geometry and Hamiltonian dynamics and construct efficient algorithms working well in high dimensions with strong posterior correlations. The authors, construct two algorithms namely manifold MALA (MMALA) and Riemann manifold Hamiltonian Monte Carlo (RMHMC). The RMHMC lies beyond the material used/covered in this thesis and we therefore restrict ourselves in briefly describing the idea of MMALA since a simplified version of it will be used later in the thesis. For an introduction and review of Hamiltonian Monte Carlo schemes we refer the reader to Chapter 5 of Brooks et al. (2011)

In analogy to (2.1.6), the underlying diffusion of the simple preconditioned MALA is described by,

$$d\boldsymbol{\theta}_t = \frac{1}{2}\boldsymbol{M}\nabla\log\pi(\boldsymbol{\theta}_t)dt + \boldsymbol{M}^{1/2}d\boldsymbol{B}_t,$$

where the preconditioning matrix $\boldsymbol{M}$ is fixed. Girolami & Calderhead (2011) construct a preconditioning matrix that is position specific and therefore define the above diffusion with a position dependent volatility matrix as shown below,

$$d\boldsymbol{\theta}_t = \frac{1}{2}\{\boldsymbol{M}(\boldsymbol{\theta}_t)\nabla\log\pi(\boldsymbol{\theta}_t)dt + \boldsymbol{\Delta}(\boldsymbol{\theta}_t)\} + \boldsymbol{M}^{1/2}(\boldsymbol{\theta}_t)d\boldsymbol{B}_t,$$

where

$$\Delta_i(\boldsymbol{\theta}_t) = |M(\boldsymbol{\theta}_t)|^{1/2} \sum_{j=1}^{d} \frac{\partial}{\partial \boldsymbol{\theta}_j} \left\{ [\boldsymbol{M}(\boldsymbol{\theta}_t)]_{ij} \, |M(\boldsymbol{\theta}_t)|^{-1/2} \right\}. \tag{2.1.8}$$

In the above expression we have accounted for the transcription error in the drift term clarified by Xifara et al. (2014). Using an expansion of the gradient term in (2.1.8), Girolami & Calderhead (2011) discretise the above equation to obtain the proposal density

$$\boldsymbol{\theta}^{prop} \sim \text{MVN}\left(\boldsymbol{\mu}\left(\boldsymbol{\theta}^{cur}, \lambda\right), \lambda \boldsymbol{M}\left(\boldsymbol{\theta}^{cur}\right)\right) \tag{2.1.9}$$

with the i-th element of $\boldsymbol{\mu}\left(\boldsymbol{\theta}^{cur}, \lambda\right)$ given by,

$$\begin{aligned} \boldsymbol{\theta}_i^{cur} \quad &+ \quad \frac{\lambda}{2} \left\{ \boldsymbol{M}\left(\boldsymbol{\theta}^{cur}\right) \nabla_{\boldsymbol{\theta}} \log \pi\left(\boldsymbol{\theta}^{cur} | \boldsymbol{y}\right) \right\}_i - \lambda \sum_{j=1}^{d} \left\{ \boldsymbol{M}\left(\boldsymbol{\theta}^{cur}\right) \frac{\partial \boldsymbol{M}^{-1}\left(\boldsymbol{\theta}^{cur}\right)}{\partial \boldsymbol{\theta}_j} \boldsymbol{M}\left(\boldsymbol{\theta}^{cur}\right) \right\}_{ij} \\ &+ \quad \frac{\lambda}{2} \sum_{j=1}^{d} \left\{ \boldsymbol{M}\left(\boldsymbol{\theta}^{cur}\right) \right\}_{ij} \text{tr} \left\{ \boldsymbol{M}\left(\boldsymbol{\theta}^{cur}\right) \frac{\partial \boldsymbol{M}^{-1}\left(\boldsymbol{\theta}^{cur}\right)}{\partial \boldsymbol{\theta}_j} \right\}. \end{aligned} \tag{2.1.10}$$

In that way, according to the authors, the diffusion is defined on a Riemann manifold and the use of $\boldsymbol{M}(\boldsymbol{\theta}^{cur})$ is justified as it can be viewed as the metric tensor describing the curvature of the manifold. In the case where the elements of $\boldsymbol{\Delta}(\boldsymbol{\theta}_t)$ are 0, the resulting proposal reduces to a preconditioned MALA with the position dependent preconditioning matrix $\boldsymbol{M}(\boldsymbol{\theta}^{cur})$. As we see from expression (2.1.10), the MMALA can be computationally expensive since it requires the calculation of third derivatives whereas the typical gain in efficiency over sMMALA can be small.

2.1.7 Independence sampler (MHIS)

Another type of MH algorithm is the Independence Sampler, Tierney (1994), for which the proposal does not depend on the current value of the chain, $\boldsymbol{\theta}^{cur}$. The next value in the chain is not actually independent of the current one since the usual accept reject scheme is used. Since the proposal distribution, q, does not depend on the current value the acceptance probability becomes

$$\alpha(\boldsymbol{\theta}^{cur}, \boldsymbol{\theta}^{prop}) = \min\left(1, \frac{\pi(\boldsymbol{\theta}^{prop})q(\boldsymbol{\theta}^{cur})}{\pi(\boldsymbol{\theta}^{cur})q(\boldsymbol{\theta}^{prop})}\right). \tag{2.1.11}$$

In the case of the MHIS the need of a proposal that mimics the target is very important. One useful strategy is to choose a proposal with mode, and curvature at the mode, matching these of the target distribution. Moreover, in order to be sure that the whole target is explored we want the proposal to have heavier tails than the target distribution; this is known as the heavy tail rule. If the proposal has lighter tails than the target then it is highly likely that the proposed values will be within the main body of the distribution and the tails will not be well explored. However, when eventually the chain does move to the tail, the probability that subsequent proposals will be accepted is very small and so the chain mixes very poorly. For instance consider that θ is one dimensional and the proposal has lighter tails than the taregt as illustrated in Figure 2.1 and a relatively constant ratio $\pi(\theta)/q(\theta)$ for values of θ in the main body of the distribution. Assume that the current value of θ is θ_1 and a move to θ_2 is proposed. In this case, $\pi(\theta_2)/q(\theta_2)$ will be very high and ince $\pi(\theta_1)/q(\theta_1)$ is relatively constant, the value of the acceptance ratio, in 2.1.11,

$$\frac{\pi(\theta_2)q(\theta_1)}{\pi(\theta_1)q(\theta_2)} = \frac{\pi(\theta_2)/q(\theta_2)}{\pi(\theta_1)/q(\theta_1)}.$$

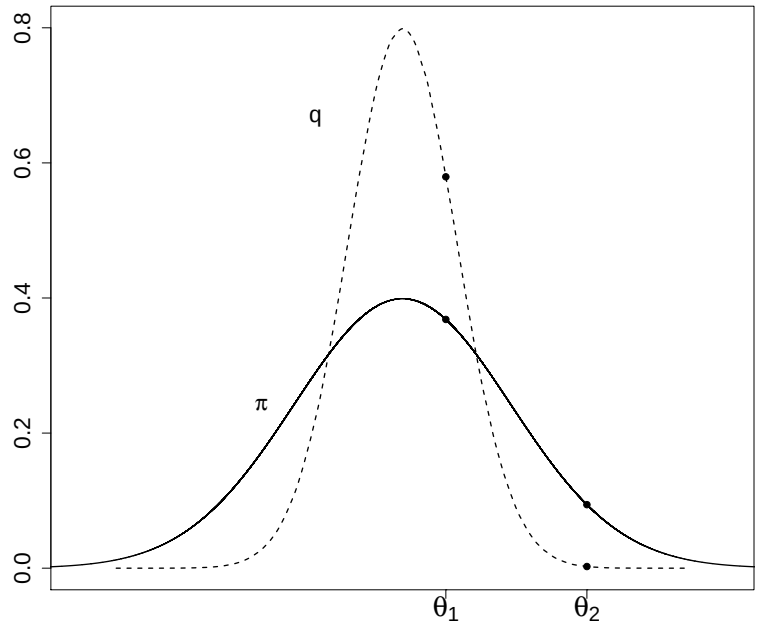

Figure 2.1: Density of the proposal distribution, $q(\theta)$ (dashed line) and of the target distribution $\pi(\theta)$ (solid line).

will be large. Therefore, such a proposed value as θ_2 will be accepted. If however we consider the opposite scenario of currently being to θ_2 and proposing a move to θ_1, the ratio $\pi(\theta_1)/q(\theta_1)$ is almost constant and $\pi(\theta_2)/q(\theta_2)$ is very high. Therefore, the acceptace ratio

$$\frac{\pi(\theta_1)q(\theta_2)}{\pi(\theta_2)q(\theta_1)} = \frac{\pi(\theta_1)/q(\theta_1)}{\pi(\theta_2)/q(\theta_2)}$$

will be very small and such moves will be always rejected making the return back to the main body of the target difficult .

As a final note we would like to mention the effect of choosing as a proposal the prior distribution. This might seem quite tempting since the acceptance ratio simplifies further to just the ratio of the likelihoods. However, if the likelihood is informative, so that the posterior and prior are dissimilar, then we might end up with very low acceptance rates. In particular, we will be proposing values that have high probability according to the

prior but may not correspond to a large likelihood value and having, as a result, to reject these moves. See Gamerman & Lopes (2006) for a detailed discussion.

2.1.8 Adaptive MCMC

As already mentioned, an appropriately shaped and also optimally tuned proposal distribution is crucial for constructing a well-mixing MCMC scheme. However, in complicated, high-dimensional targets this can be extremely difficult especially when a good estimate of the posterior covariance matrix is not available. But even in low-dimensional targets defining an optimal value for the tuning parameter λ can be painful since, usually in practice, this has to be done through trial an error.

Adaptive MCMC was created to provide a solution to such problems and minimise, as much as possible, the user intervention. The idea behind adaptive MCMC schemes is to use the information that becomes available as the sampler runs. This information is used in order to automatically update the variance of the proposal distribution, using estimates obtained from the empirical distribution of the chain so far, according to a pre-defined updating rule.

Although in practice such schemes have by now become straightforward to implement there are certain issues to be considered. First of all, if the chain starts away from the main body of the target distribution, i.e., in the tails, then there is a chance for the chain to stay there for a long time and, in the time available, not explore areas with high posterior probability. This is because the algorithm learns about this insignificant area and automatically adjusts the proposal distribution so that it can efficiently explore that specific part of the support. Therefore, even if the sampler is ergodic, given that it is run for a finite number of iterations, the posterior estimates obtained might not represent

the truth since it may take a long time to obtain an adequate sample. For that reason, in practice, the proposal distribution will sometimes use a mixture of an adaptive and a non-adaptive kernel in order to minimise the chance of being trapped in such areas (see for instance Sherlock et al. (2013) and Fearnhead et al. (2014)).

Another issue is by how much and how often should the proposal variance change during the MCMC scheme. For instance, it is sensible to initially let the sampler run with a fixed proposal and once a certain number of accepted moves has been achieved then start adapting the proposal distribution. This is to ensure that the chain has moved sufficiently so that the covariance matrix is not singular.

However, since the transition kernel keeps changing for as long as the sampler runs, convergence to the stationary distribution is not anymore guaranteed and hence nor is the ergodicity of the chain. There has been a lot of research on the ergodicity and convergence properties of adaptive MCMC schemes and two important concepts that have arisen are the diminishing adaptation and the containment condition. Let $(\boldsymbol{\theta}_n, \gamma_n)$ be the position and transition kernel at the n-th iteration under the adaptive scheme. Given the starting value and initial kernel $(\boldsymbol{\theta}_0, \gamma_0)$, the containment condition states that if the chain were to start at $\boldsymbol{\theta}_n$ with a non-adaptive fixed kernel γ_n then, irrespective of n, the chain will have nearly converged to $\pi(\cdot)$ after N iterations, for large enough N. For nearly all possible $(\boldsymbol{\theta}_n, \gamma_n)$. As mentioned in Brooks et al. (2011), Chapter 4, the containment condition will usually hold for most adaptive schemes given that a reasonable adaptation rule is used and therefore focus is placed on the notion of diminishing adaptation.

Diminishing adaptation suggests that changes in the proposal should become negligible as the chain evolves. This is to ensure that after a large number of iterations the successive transition kernels are similar and therefore reach an equilibrium. However, these changes

should also be large enough to reflect necessary changes in the covariance matrix.

For a thorough review and theoretical justifications on convergence and ergodicity results of adaptive MCMC we refer the reader to Andrieu & Thoms (2008) and Roberts & Rosenthal (2009) and the references therein.

2.2 Model based geostatistics

As mentioned in the Introduction of the thesis, Geostatistics concerns the study of a continuous spatial phenomenon. This phenomenon is usually modelled through a stationary Gaussian process, $\{S(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^2\}$. By stationary we mean that the expectation and variance of the process is the same for all $\boldsymbol{x}$ and the correlation between $S(\boldsymbol{x}_i)$ and $S(\boldsymbol{x}_j)$ only depends on the distance between $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$. Additionally, the Gaussianity of the process $S(\boldsymbol{x})$ implies that $S(\boldsymbol{x}_1), ..., S(\boldsymbol{x}_d)$ are jointly normally distributed for any set of locations $x_1, ..., x_d$, . This process however, is not directly observed. Instead, there is available only a finite sample of observations of a random variable, $\boldsymbol{Y}$, at specific sampling locations, $\boldsymbol{x}_i$, $i = 1, 2, ..., d$, over the area of interest. These observations, $\boldsymbol{y}$, are usually assumed to be either identical to or a noisy version of the underlying true process or of a function of it. Interest usually lies in predicting the realisation of the process, or a functional of it, at unsampled locations or making inference about some parameters of the model.

The term 'model-based geostastics' was introduced in the seminal paper of Diggle et al. (1998) to describe the unified modelling and inferential framework provided by the authors.

In this section we describe the components and the formulation of the simple linear

spatial model and show how this is extended to the generalised linear spatial model just like the simple linear model extends to a GLM. The LSM and the GLSM can be viewed as a linear or generalised linear mixed-effects model (Breslow & Clayton 1993) respectively where the random effects form a Gaussian random field.

In Section 2.2.2 we outline the simple LSM along with the inferential procedure usually used. Section 2.2.3 presents the most widely used covariance functions used to model the spatial dependence and the characteristics that each one bestows on the underlying process. Finally, in Section 2.2.6 we describe the GLSM and review some of the current MCMC schemes used for inference in Section 2.2.7. Unless otherwise stated, the two main sources of information for this Section are Diggle et al. (2007) and Diggle et al. (2003).

2.2.1 The Gaussian process

As it constitutes a key component of the linear spatial model this section focuses on the definition of the Gaussian process and the notion of weak and strong stationarity.

Definition 2.2.1. *A stochastic process, or random field, S with parameter space T is a collection of random variables $\{S(\boldsymbol{x}) : \boldsymbol{x} \in T\}$. The dimension of T is N and the random variables $S(\boldsymbol{x})$ are vectors of dimension n then the random field S is said to be an (N, n) random field.*

In our setting $\boldsymbol{x}$ represents the spatial coordinates of a sampling point and therefore the set T is of dimension $N = 2$. Also, at each $\boldsymbol{x}$, $S(\boldsymbol{x})$ is one dimensional and therefore gives rise to an $(2, 1)$ dimensional random field. For every stochastic process we can define the mean and covariance function given by

$$\mu(\boldsymbol{x}) = \mathbb{E}\left[(\boldsymbol{x})\right], \quad c(\boldsymbol{x}_i,\ \boldsymbol{x}_j) = \mathrm{Cov}\left[S(\boldsymbol{x}_i),\ S(\boldsymbol{x}_j)\right]$$

respectively. If the mean function of the process S is constant for every $\boldsymbol{x}$ and the covatriance function, $c(\boldsymbol{x}_i, \boldsymbol{x}_j)$, only depends on the difference $\boldsymbol{x}_i - \boldsymbol{x}_j$, i.e.,

$$\mu(\boldsymbol{x} + \boldsymbol{t}) = \mu(\boldsymbol{x}) \quad \text{and} \quad c(\boldsymbol{x}_i + \boldsymbol{h},\ \boldsymbol{x}_j + \boldsymbol{h}) = c(\boldsymbol{x}_i,\ \boldsymbol{x}_j)$$

then the process S is said to be weakly stationary. A stronger form of statioanrity is that of strong stationarity. The stochastic process S is said to be (strongly) stationary if, its finite-dimensional distributions are invariant under the operation $(+)$, i.e., if the joint distribution of $(S(\boldsymbol{x}_1 + \boldsymbol{h}), ..., S(\boldsymbol{x}_d + \boldsymbol{h}))$ is independent of $\boldsymbol{h}$ for all $\boldsymbol{x}_j \in \mathbb{R}^2$ and any $d \geq 1$.

Definition 2.2.2. *The Gaussian process S, or a Gaussian random field, is a random field for which the joint distribution of $(S(\boldsymbol{x}_1), ..., S(\boldsymbol{x}_d))$ is multivariate Gaussian for any finite $d \geq 1$ and every $(\boldsymbol{x}_1, ..., \boldsymbol{x}_d)$.*

In the case of a Gaussian process weak stationarity implies strong stationarity. For that reason in the following sections we do not distinct the two and in general refer to a stationary Gaussian process without clarifying whether the process is weakly or strongly stationary. In general though this does not hold. If a stochastic process is strongly stationary then it is also weakly stationary but the opposite does not hold. For a thorough study of Gaussian processes and in general random fields we refer the reader to Adler & Taylor (2007).

2.2.2 The linear spatial model (LSM)

Let $\{S(\boldsymbol{x}) : \boldsymbol{x} \in \mathbb{R}^2\}$ be the underlying process of interest and $\boldsymbol{y} = (y_1, ..., y_d)^{'}$ be a realisation of the observable random variable $\boldsymbol{Y} = (Y_1, ..., Y_d)^{'}$ at sampling locations $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_d\}$. Also let us assume that $\boldsymbol{f}_i$, $i = 1, ..., d$, is a column vector of available

explanatory variables measured at the sampling locations $\boldsymbol{x}_i$. In practice, $S(\boldsymbol{x}_i)$, $i = 1, ..., d$, is assumed to be normally distributed with mean 0 and marginal variance σ^2. The correlation structure of the process $S(\boldsymbol{x})$ will be discussed later. For $i = 1, ..., d$, conditionally on $S(\boldsymbol{x})$, Y_i are assumed to be independent following a normal distribution with variance τ^2 and mean linearly related to $\boldsymbol{f}_i$ and $S(\boldsymbol{x}_i)$. The equivalent mathematical formulation of the model is given by,

$$Y_i = \boldsymbol{f}_i^{'}\boldsymbol{\beta} + S(\boldsymbol{x}_i) + Z_i, \qquad i = 1, ..., d \tag{2.2.1}$$

where $Z_i \sim \text{normal}\left(0, \tau^2\right)$ and are mutually independent. At a specific location, even if the true value of the underlying process were known there would be some variability between consecutive measurements. Such variations are depicted by the conditional variance, τ^2, of $Y_i|S(\boldsymbol{x}_i), \boldsymbol{\beta}$ which is either interpreted as measurement error or small scale variation.

It is intuitive to assume that nearby locations would give rise to similar measurements while the correlation between two locations fades away as their distance increases. The alternative, where the correlation increases with distance, would not lead to a positive definite covariance matrix. Let $\boldsymbol{S} := \left(S(\boldsymbol{x}_1), ..., S(\boldsymbol{x}_d)\right)^{'}$ and $\text{Cor}(S(\boldsymbol{x}_i), S(\boldsymbol{x}_j)) = \rho(u_{ij}; \phi)$ where $\rho(\cdot\,; \phi)$ denotes a correlation function parametrised over some correlation parameter ϕ. Then it follows that,

$$\boldsymbol{S} \sim \text{MVN}\left(\boldsymbol{0}, \sigma^2\boldsymbol{R}(\phi)\right) \tag{2.2.2}$$

where $\boldsymbol{R}(\phi)$ denotes the correlation matrix with elements $\boldsymbol{R}_{ij} = \rho(u_{ij}; \phi)$ and conse-

quently

$$\boldsymbol{Y} \sim \text{MVN}\left(\boldsymbol{F\beta}, \sigma^2\boldsymbol{R}(\phi) + \tau^2\boldsymbol{I}\right) \tag{2.2.3}$$

with $\boldsymbol{F}$ representing the design matrix, with rows $\boldsymbol{f}_i^{'}$, and $\boldsymbol{I}$ being the $d \times d$ identity matrix.

A widely used equivalent formulation considers an underlying Gaussian process which does not have a constant zero mean. In this case the covariate information is incorporated in to the mean of the process. If we define $\boldsymbol{\eta} := \boldsymbol{F\beta} + \boldsymbol{S}$, then,

$$\boldsymbol{Y} = \boldsymbol{\eta} + \boldsymbol{Z}, \tag{2.2.4}$$

where $\boldsymbol{\eta} = (\eta(\boldsymbol{x}_1), ..., \eta(\boldsymbol{x}_d)) \sim \text{MVN}\left(\boldsymbol{F\beta}, \sigma^2\boldsymbol{R}(\phi)\right)$, $\boldsymbol{Z} \sim \text{normal}\left(\boldsymbol{0}, \tau^2 I\right)$. Equivalently,

$$Y_i = \boldsymbol{f}_i^{'}\boldsymbol{\beta} + S(\boldsymbol{x}_i) + Z_i = \ \eta_i + Z_i, \quad \text{for} \quad i = 1, ..., d \tag{2.2.5}$$

where marginally, $\eta_i \sim \text{Normal}\left(\boldsymbol{f}_i'\boldsymbol{\beta}, \sigma^2\right)$. Note that the process $\{\eta(\boldsymbol{x}) : \boldsymbol{x} \in \boldsymbol{R}^2\}$ is no longer stationary as defined in Section 2.2.1 but, as mentioned in Diggle et al. (2007), it is covariance stationary. In the absence of explanatory variables, where $\boldsymbol{f}_i^{'}\boldsymbol{\beta}$ is replaced by a constant and fixed mean effect $\boldsymbol{\beta}$, then both $\boldsymbol{\eta}(\boldsymbol{x})$ and $\boldsymbol{S}(\boldsymbol{x})$ are stationary Gaussian processes.

Although the working framework of the normal model is well established, the assumption of a linear relationship between the response variable Y and the signal process appears to be quite unrealistic in real life phenomena. Consider for instance applications where the observable variable Y concerns counts or in general has an asymmetric distribution. In Section 2.2.6 we talk about the generalised linear spatial model which deals with non

Gaussian data. Historically, before the introduction of GLSMs, approximate normality of the response variable was achieved through a transformation of the data and then inference was carried out under the Gaussian framework. A widely used family of such transformations is the Box-Cox (Box & Cox 1964) which however can be applied to strictly positive-valued data and comes at the additional cost of estimating an additional parameter. Additionally, after such transformations, the model parameters, i.e., $\boldsymbol{\beta}$, might not have a sensible natural interpretation. For a more detailed discussion and implementation of Box-Cox transformations on the geostatistical model see Christensen, Diggle & Ribeiro (2001).

To simplify notation for the rest of this thesis we will suppress $S(\boldsymbol{x}_i)$ to S_i and $\eta(\boldsymbol{x}_i)$ to η_i.

2.2.3 Models for the correlation structure

In the previous section we briefly discussed some of the assumptions made regarding the correlation structure of the latent process. For instance, the process is often assumed to be stationary, isotropic and also the correlation between any two points should decrease as the distance increases. In addition, the correlation function used should be positive definite.

A flexible family of correlation functions satisfying these properties is the Matérn family as introduced by Matérn (1960) given by,

$$\rho(u;\phi,\kappa) = \frac{(u/\phi)^{\kappa} K_{\kappa}(u/\phi)}{2^{\kappa-1}\Gamma(\kappa)}, \tag{2.2.6}$$

where $\Gamma(\cdot)$ is the Gamma function and $K_{\kappa}(\cdot)$ corresponds to the modified Bessel function of order κ. The parameter $\phi > 0$ is a scale parameter which gives the rate of decay of the

correlation as the distance u increases. Given any two points u units apart, the larger ϕ is, the higher the correlation between these two points will be.

The Matérn family gains its flexibility from the shape parameter κ since it defines the differentiability of the correlation function at the origin or equivalently the smoothness of the stochastic process $S(\cdot)$. In particular, κ reflects the short distance dependence of the random field. As κ increases the correlation remains at higher levels for longer and the latent process becomes smoother.

A particular property that describes the smoothness of a stochastic process is the mean-square differentiability.

Definition 2.2.3. *Mean Square Differentiability*

A stochastic process $S(\boldsymbol{x})$ with finite second moments is mean-square differentiable with mean-square derivative $S^{'}(\boldsymbol{x})$ if as $\|\epsilon\| \to 0$,

$$\mathbb{E}\left[\left\{\frac{S(\boldsymbol{x}+\boldsymbol{\epsilon})-S(\boldsymbol{x})}{\|\boldsymbol{\epsilon}\|}-S^{'}(\boldsymbol{x})\right\}^{2}\right]\to 0. \tag{2.2.7}$$

Higher order derivatives can be obtained in a similar way. A very helpful result that provides links between the differentiability of the correlation function at the origin with the mean-square differentiability of the stochastic process is the following. If the correlation structure of the latent process $S(\boldsymbol{x})$ is modelled using the Matérn family of correlation of order κ then $S(\boldsymbol{x})$ is $\lceil\kappa - 1$ times mean-square differentiable, where $\lceil\kappa$ denotes the smallest integer that is not greater than κ. The more times mean-square differentiable a process is the smoother it will be and therefore the stronger the correlation near the origin, i.e., for distances very close to 0.

To illustrate the effect of mean-square differentiability we consider two well-known cases

of the Matérn family; the exponential and Gaussian correlation functions. In particular, as $\kappa \to \infty$, $\rho(u;\phi) \to \exp\{-(u/\phi)^2\}$ corresponding to the Gaussian correlation function, which should not be confused with the normal distribution, and for $\kappa = 0.5$ we obtain the exponential correlation function given by $\rho(u;\phi) = \exp(-u/\phi)$.

Figure 2.2 shows the exponential and Gaussian correlation functions. Since the parameters ϕ and κ are not orthogonal the values of ϕ have been matched so that in both cases $\rho(u) = 0.05$ at the same distance u. For the exponential correlation $\phi = 1$ and for the Gaussian correlation function $\phi \approx 1.73$. A process arising form the exponential correlation function is not mean-square differentiable whereas process arising form the Gaussian correlation function is infinitely mean-square differentiable. As we see, in the case of the exponential correlation function the correlation drops quickly near the origin whereas in the case of the Gaussian the correlation stays near 1 for distances up to 0.5. Other valid correlation functions outside the Matérn family, are the powered exponential

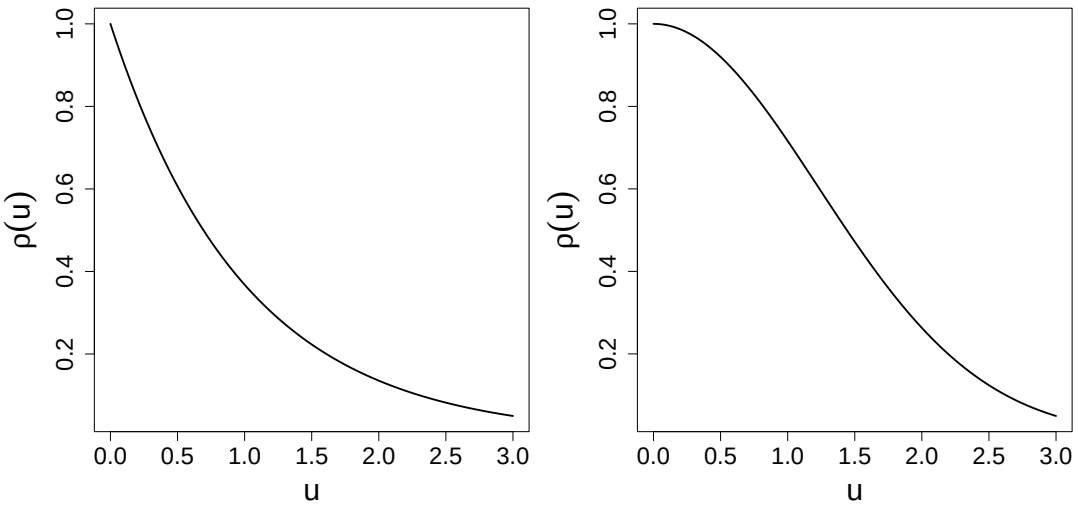

Figure 2.2: Correlation against distance. Left: exponential correlation function with $\phi = 1$, Right: Gaussian correlation function with $\phi = 1.73$. The parameter ϕ has been matched so that in both cases $\rho(u) = 0.05$ at the same distance u.

and the spherical correlation function. The powered exponential function which is given

by,

$$\rho(u;\phi,a) = \exp\left\{-\left(\frac{u}{\phi}\right)^a\right\}, \quad 0 < a \leq 2$$

embodies both the exponential, for $a = 1$, and the Gaussian correlation for $a = 2$. However, the powered exponential family is not as flexible as the Matérn family since the underlying process $S(\boldsymbol{x})$ will not be mean-square differentiable for $a < 2$.

The spherical correlation function,

$$\rho(u;\phi) = \begin{cases} 1 - \frac{3}{2}\frac{u}{\phi} + \frac{1}{2}\left(\frac{u}{\phi}\right)^3, & 0 \leq u \leq \phi \\ 0, & u > \phi, \end{cases}$$

is not mean-square differentiable but it is even more restrictive than the powered exponential since it assumes that at distance equal to ϕ the correlation becomes exactly zero and therefore has a finite range. As illustrated in Warnes & Ripley (1987), Mardia & Watkins (1989) and further discussed in Stein (1999), the spherical correlation function can usually give rise to a multimodal log-likelihood and therefore maximum likelihood techniques can be problematic when it comes to parameter estimation.

For simplicity, throughout this thesis we will usually denote the correlation matrix $\boldsymbol{R}(\phi,\kappa)$ simply by $\boldsymbol{R}$. If we want though to stress that this matrix is a function of ϕ,κ we will use $\boldsymbol{R}(\phi,\kappa)$.

For a more thorough and technical investigation of correlation functions we refer the reader to Chapter 2 of Stein (1999).

For the LSM inference can be carried out both under the classical and Bayesian framework. The likelihood is tractable since the spatial process can be integrated out and maximum likelihood estimates of the parameters can be obtained. In a Bayesian setting

on the other hand conjugate priors can be used for the parameters leading to exact sampling from the full posterior distribution.

2.2.4 Classical Inference and prediction for the LSM

Maximum likelihood estimation

In theory, the parameters of the model to be estimated are $\boldsymbol{\beta}, \sigma^2, \phi, \kappa$. However, the parameter κ is in practice poorly identified. Ideally the choice of κ should be based on some scientific knowledge on the smoothness of the spatial process and the correlation function to be used. Since this is not always the case, in practice, κ is ussually either assumed to be fixed to an arbitrary value or the log-likelihood is maximised with respect to κ over a discrete set of values, i.e., $\kappa \in \{0.5, 1.5, 2.5, 3.5\}$. Estimation of $\boldsymbol{\beta}, \sigma^2$ and ϕ can be carried out assuming κ is fixed. The likelihood of the LSM is given by,

$$\begin{aligned} l\left(\boldsymbol{\beta}, \sigma^2, \phi, \nu^2\right) = & - \frac{1}{2} d \log\left(2\pi\right) - \frac{1}{2} \log\left|\sigma^2 \boldsymbol{R}(\phi) + \tau^2 \boldsymbol{I}\right| \\ & - \frac{1}{2\sigma^2} \left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right)' \left(\sigma^2 \boldsymbol{R}(\phi) + \tau^2 \boldsymbol{I}\right)^{-1} \left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right). \end{aligned}$$

In order to obtain the maximum likelihood estimates, $\hat{\boldsymbol{\beta}}, \hat{\sigma}^2$, $\hat{\phi}$, we proceed with the following reparametrisation. Let $\nu^2 := \frac{\tau^2}{\sigma^2}$ then the correlation matrix for $\boldsymbol{S}$ becomes $\boldsymbol{C} = \boldsymbol{R} + \nu^2 \boldsymbol{I}$ and the log-likelihood is now given by,

$$l\left(\boldsymbol{\beta}, \sigma^2, \phi, \nu^2\right) = -\frac{1}{2} \left\{ d \log\left(2\pi\right) + \log\left|\sigma^2 \boldsymbol{C}\right| + \frac{1}{\sigma^2} \left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right)' \boldsymbol{C}^{-1} \left(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}\right) \right\}. \quad (2.2.8)$$

Treating the correlation matrix $\boldsymbol{C}$ as being fixed, i.e. for a fixed pair (ϕ, ν), maximisation of the above likelihood yields the maximum likelihood estimates,

$$\begin{aligned}\hat{\boldsymbol{\beta}}\left(\phi,\nu^2\right) &= \left(\boldsymbol{F}'\boldsymbol{C}^{-1}\boldsymbol{F}\right)^{-1}\boldsymbol{F}'\boldsymbol{C}^{-1}\boldsymbol{y},\\ \hat{\sigma}^2\left(\phi,\nu^2\right) &= \frac{1}{d}\left(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}\left(\boldsymbol{C}\right)\right)'\boldsymbol{C}^{-1}\left(\boldsymbol{y}-\boldsymbol{F}\hat{\boldsymbol{\beta}}\left(\boldsymbol{C}\right)\right),\end{aligned}$$

and substitution of the above estimates into (2.2.8) gives,

$$l\left(\hat{\boldsymbol{\beta}}(\phi,\nu),\hat{\sigma}^2(\phi,\nu),\phi,\nu\right) = -\frac{1}{2}\left\{d\log(2\pi)+\log|\hat{\sigma}^2\boldsymbol{C}|+d\right\}. \tag{2.2.9}$$

Therefore, for any given pair (ϕ, ν) we find $\hat{\boldsymbol{\beta}}(\phi, \nu)$ and hence $\hat{\sigma}^2(\phi, \nu)$ which is substituted into 2.2.9 to give the value of the profile log-likelihood for that combination of (ϕ, ν). This function of (ϕ, ν) is then maximised numerically using an iterative procedure such as the Nelder-Mead algorithm.

Prediction in the classical setting

As briefly mentioned in Section 2.2.3, in geostatistics it is of interest to predict the realisation of the underlying process S at unsampled locations. Under the classical framework estimation of model parameters and prediction constitute two different steps, with the latter having the former as a prerequisite.

The approach used for prediction is to estimate the minimum-mean-square-error predictor which in the case of the LSM coincides with the Kriging predictor. For instance let T be the random variable, that is a linear function of S, that we want to predict. The estimate $\hat{T} = \mathbb{E}[T|\boldsymbol{Y}]$ is the predictor of T that minimises the mean square error $\mathbb{E}\left[\left(T-\hat{T}\right)^2\right]$ and has prediction variance $\mathbb{V}\text{ar}[T|\boldsymbol{Y}]$.

For instance, let $\boldsymbol{x}^*$ denote a location at which we have not sampled and we wish to predict the signal at this location, i.e., $T = S^* = S(\boldsymbol{x}^*)$, based on the information provided from the data $\boldsymbol{y}$ at the sampled locations $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_d\}$. If we denote by $\boldsymbol{r}^{*'}$ the vector with elements $r_i^* = \rho(\|\boldsymbol{x}^* - \boldsymbol{x}_i\|; \phi)$ for $i = 1, ..., d$, then the joint distribution of $(T, \boldsymbol{Y})$ is a multivariate normal as shown below,

$$\begin{bmatrix} T \\ \boldsymbol{Y} \end{bmatrix} \sim \text{MVN}\left(\begin{bmatrix} 0 \\ \boldsymbol{F\beta} \end{bmatrix}, \sigma^2 \begin{bmatrix} 1 & \boldsymbol{r}^{*'} \\ \boldsymbol{r}^* & \boldsymbol{C} \end{bmatrix} \right).$$

Therefore, the distribution of $T|\boldsymbol{Y}$ will be normal with mean and variance given by

$$\mathbb{E}\left[T|\boldsymbol{Y}\right] = \boldsymbol{r}^{*'}\boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{F\beta}) = \hat{T} \quad \text{and} \quad \mathbb{V}\text{ar}\left[T|\boldsymbol{Y}\right] = \sigma^2\left(1 - \boldsymbol{r}^{*'}\boldsymbol{C}^{-1}\boldsymbol{r}^*\right).$$

Equivalently, if $T = \eta(\boldsymbol{x}^*) = \boldsymbol{f}_*'\boldsymbol{\beta} + S(\boldsymbol{x}^*)$, where $\boldsymbol{f}_*'$ denotes the vector of covariates for the location $\boldsymbol{x}^*$, then the estimate of T is given by,

$$\begin{aligned} \hat{T} &= \boldsymbol{f}_*'\boldsymbol{\beta} + \boldsymbol{r}^{*'}\boldsymbol{C}^{-1}(\boldsymbol{y} - \boldsymbol{F\beta}) \\ &= \boldsymbol{f}_*'\boldsymbol{\beta} + \sum_{i=1}^{d} w_i(\boldsymbol{x}^*)(y_i - \boldsymbol{f}_i'\boldsymbol{\beta}) \end{aligned} \tag{2.2.10}$$

In the geostatistics' literature this method is known as Kriging and is attributed to Krige (1951). As we see, from (2.2.10) the Kriging predictor is actually compromise between the unconditional mean at location $\boldsymbol{x}^*$ and the deviations of the observed data from their means. This compromise depends on the sampling design, the model parameters the prediction location $\boldsymbol{x}^*$ and the observed data. Whereas, the prediction variance does not depend on the data $\boldsymbol{y}$. Since in practice the model parameters $\boldsymbol{\beta}, \sigma^2$, ϕ and τ^2 will be unknown their estimates, i.e., maximum likelihood estimates, would be used in the

above expressions. In cases where interest lies in predicting non linear functionals of S simple Monte Carlo is usually used where we iterate between sampling from $S^*|\boldsymbol{y}$ and computing the functional of interest, resulting in a sample from the predictive of the functional of interest.

For a thorough overview of the underlying theory of Kriging see Stein (1999) and Chilès & Delfiner (1999).

2.2.5 Bayesian inference and prediction for the LSM

Under the Bayesian framework, all parameters of the model are treated as random variables and prior distributions are assigned to each parameter. In order to make inference about the parameters of the model we have to sample from the posterior distribution $\pi\left(\boldsymbol{\beta}, \sigma^2, \phi, \tau^2|\boldsymbol{y}\right)$. Under the use of certain prior distributions for the parameters we can simulate exactly from the above posterior distribution without the need for MCMC schemes. In this Section we briefly describe this approach as used in Diggle et al. (2003) and Diggle et al. (2007) and try to keep the notation consistent with the notation therein. Equations provided without proof are taken directly from Diggle et al. (2003)

Often, κ is chosen from a discrete set of positive values according to scientist's beliefs and therefore, in this section, for simplicity of presentation it is assumed to be fixed. Additionally, we assume that $\tau^2 = 0$; for details see Diggle et al. (2007). In the following we use $p(\cdot)$ to denote prior distributions and $\pi(\cdot)$ for the posteriors. The conjugate prior of $(\boldsymbol{\beta}, \sigma^2|\phi)$, is the normal-Scaled-Inverse χ^2. More explicitly, we assign the following priors to the parameters,

$$[\boldsymbol{\beta}|\sigma^2, \phi] \sim \text{MVN}\ (\boldsymbol{m}_\beta, \sigma^2 \boldsymbol{V}_\beta) \quad \text{and} \quad [\sigma^2|\phi] \sim \chi^2_{\text{SCI}}\ (n_\sigma, S^2_\sigma), \tag{2.2.11}$$

where $\chi^2_{\text{SCI}}\,(n_\sigma, S^2_\sigma)$ denotes the Scaled-Inverse χ^2 distribution with n_σ degrees of freedom and scale parameter S^2_σ. with probability density function given by,

$$p(\sigma^2|\phi) \propto \frac{1}{(\sigma^2)^{(1+n_\sigma/2)}} \exp\left(\frac{-n_\sigma S^2_\sigma}{2\sigma^2}\right).$$

Expression (2.2.11) can also be written as $(\boldsymbol{\beta}, \sigma^2|\phi) \sim \text{N}\chi^2_{\text{SCI}}\,(\boldsymbol{m}_\beta, \boldsymbol{V}_\beta, n_\sigma, S^2_\sigma)$. Using Bayes' theorem,

$$\pi(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, \phi) \propto L(\boldsymbol{\beta}, \tau^2, \sigma^2, \phi, \kappa) \times p(\boldsymbol{\beta}|\sigma^2, \phi) \times p(\sigma^2|\phi),$$

where $p(\boldsymbol{\beta}|\sigma^2, \phi)$, $p(\sigma^2|\phi)$ are the prior distributions as given in (2.2.11). It follows that the posterior distribution of $(\boldsymbol{\beta}, \sigma^2)$, $\pi(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, \phi)$, is a normal-Scaled-Inverse χ^2,

$$(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, \phi) \sim \text{N}\chi^2_{\text{SCI}}(\tilde{\boldsymbol{\beta}}, \boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}, n_\sigma + d, D^2),$$

with

$$\tilde{\boldsymbol{\beta}} = \boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}(\boldsymbol{V}_\beta^{-1}\boldsymbol{m}_\beta + \boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{y}) \quad , \quad \boldsymbol{V}_{\tilde{\boldsymbol{\beta}}} = (\boldsymbol{V}_\beta^{-1} + \boldsymbol{F}'\boldsymbol{R}^{-1}\boldsymbol{F})^{-1},$$

and

$$D^2 = \frac{n_\sigma S^2_\sigma + \boldsymbol{m}'_\beta \boldsymbol{V}_\beta^{-1}\boldsymbol{m}_\beta + \boldsymbol{y}' R^{-1}\boldsymbol{y} - \tilde{\boldsymbol{\beta}}' \boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}^{-1}\tilde{\boldsymbol{\beta}}}{n_\sigma + d}.$$

In principle the prior distribution for ϕ, $p(\phi)$, could be continuous, however, choosing it to be discrete allows us to sample from the posterior $\pi(\phi|\boldsymbol{y})$ exactly. Therefore, the joint posterior distribution of the parameters is $\pi(\phi, \sigma^2, \boldsymbol{\beta}|\boldsymbol{y}) = \pi(\boldsymbol{\beta}, \sigma^2|\phi, \boldsymbol{y}) \times \pi(\phi|\boldsymbol{y})$ with the posterior distribution of ϕ being

$$\pi(\phi|\boldsymbol{y}) \propto p(\phi)|\boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}|^{1/2}|\boldsymbol{R}|^{-1/2}(D^2)^{\frac{-(n_\sigma+d)}{2}}.$$

In order to sample from the posterior distribution of $(\boldsymbol{\beta}, \sigma^2, \phi)$ we begin by calculating the posterior probabilities $\pi(\phi|\boldsymbol{y})$ and hence sampling ϕ from $\pi(\phi|\boldsymbol{y})$, simulate σ^2 from $\pi(\sigma^2|\phi, \boldsymbol{y})$ and finally simulate $\boldsymbol{\beta}$ from $\pi(\boldsymbol{\beta}|\sigma^2, \phi, \boldsymbol{y})$.

Now, let $\boldsymbol{\eta}^* = (\eta(\boldsymbol{x}_1^*), ..., \eta(\boldsymbol{x}_m^*))$ be a vector with the values of the signal process that we want to predict at m unobserved locations $(\boldsymbol{x}_1^*, ..., \boldsymbol{x}_m^*)$. According to our model setting, $(\boldsymbol{Y}, \boldsymbol{\eta}^*)$ have a multivariate normal distribution. Therefore,

$$(\boldsymbol{\eta^*}|\boldsymbol{Y}, \sigma^2, \boldsymbol{\beta}, \phi) \sim \text{MVN}\left(\boldsymbol{F}^*\boldsymbol{\beta} + \boldsymbol{Q}'\boldsymbol{R}^{-1}(\boldsymbol{y} - \boldsymbol{F}\boldsymbol{\beta}), \sigma^2(\boldsymbol{R}^* - \boldsymbol{Q}'\boldsymbol{R}^{-1}\boldsymbol{Q})\right), \qquad (2.2.12)$$

where $\boldsymbol{F}^*$, $\boldsymbol{F}$ correspond to the design matrices regarding the unobserved and observed locations respectively. $\boldsymbol{Q}$ is a $d \times m$ matrix with elements, $\boldsymbol{Q}_{ij} = \text{Cor}(\eta(\boldsymbol{x}_i), \eta(\boldsymbol{x}_j^*))$ and $\boldsymbol{R}^*$ is an $m \times m$ matrix with elements $\boldsymbol{R}_{ij}^* = \text{Cor}(\eta(\boldsymbol{x}_i^*), \eta(\boldsymbol{x}_j^*))$.

The predictive distribution of $\boldsymbol{\eta}^*$, assuming a given value of ϕ is obtained by integrating out $\boldsymbol{\beta}, \sigma^2$ from the joint posterior distribution of $\boldsymbol{\eta}^*, \boldsymbol{\beta}, \sigma^2$ as shown below

$$\pi(\boldsymbol{\eta}^*|\boldsymbol{y}, \phi) = \int\int \pi(\boldsymbol{\eta}^*|\boldsymbol{y}, \boldsymbol{\beta}, \sigma^2, \phi) \times \pi(\boldsymbol{\beta}, \sigma^2|\boldsymbol{y}, \phi)\ d\boldsymbol{\beta}d\sigma^2,$$

resulting in an m-dimensional multivariate t-distribution the mean and variance of which is,

$$\text{E}[\boldsymbol{\eta}^*|\phi, \boldsymbol{y}] = (\boldsymbol{F^*} - \boldsymbol{Q}'\boldsymbol{R}^{-1}\boldsymbol{F})\boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}\boldsymbol{V}_{\beta}^{-1}\boldsymbol{m}_b + \left(\boldsymbol{Q}'\boldsymbol{R}^{-1} + (\boldsymbol{F^*} - \boldsymbol{Q}'\boldsymbol{R}^{-1}\boldsymbol{F})\boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}\boldsymbol{F}'\boldsymbol{R}^{-1}\right)\boldsymbol{y},$$

$$\text{Var}[\boldsymbol{\eta}^*|\phi, \boldsymbol{y}] = \sigma^2\left(\boldsymbol{R}^* - \boldsymbol{Q}'\boldsymbol{R}^{-1}\boldsymbol{Q}\right) + (\boldsymbol{F^*} - \boldsymbol{Q}\boldsymbol{R}^{-1}\boldsymbol{F})(\boldsymbol{V}_{\beta}^{-1} + \boldsymbol{V}_{\tilde{\boldsymbol{\beta}}}^{-1})^{-1}(\boldsymbol{F^*} - \boldsymbol{Q}'\boldsymbol{R}^{-1}\boldsymbol{F})',$$

respectively. Finally, the predictive distribution of $\eta^* = \eta(\boldsymbol{x}^*)$ at an arbitrary location

$\boldsymbol{x}^*$is given by

$$\pi(\eta^*|\boldsymbol{y}) = \int \pi(\eta^*|\phi, \boldsymbol{y})\pi(\phi|\boldsymbol{y})d\phi.$$

In order to simulate from this predictive distribution, we sample ϕ from $\pi(\phi|\boldsymbol{y})$ and then simulate $\boldsymbol{\eta^*}$ from $\pi(\boldsymbol{\eta^*}|\phi, \boldsymbol{y})$.

2.2.6 The generalised linear spatial model (GLSM)

Diggle et al. (1998), were the first to introduce the use of generalised linear spatial models (GLSMs) in the geostatistical setting and predict non-linear functionals of S under the Bayesian framework using Markov Chain Monte Carlo methods. This is achieved by incorporating the signal process S within the linear predictor of a Generalised Linear Model (GLM) (McCullagh & Nelder (1989)).

The assumptions underlying the GLSM are similar to those in the case of the LSM with two fundamental differences. First of all, $Y_i|S(\boldsymbol{x}_i)$ is not Gaussian and also the mean of the response variable Y is not linearly related with the process S. As in the LSM we assume that $Y_i|S(\boldsymbol{x}_i)$ are mutually independent and we denote the conditional expectations by $\mu_i = E[Y_i|s(\boldsymbol{x}_i)]$. However, now

$$h(\mu_i) = \boldsymbol{f}_i^{'}\boldsymbol{\beta} + S(\boldsymbol{x}_i) = \eta_i \tag{2.2.13}$$

where $h(\cdot)$ is the link function, $\boldsymbol{f}_i$ are the explanatory variables associated with location $\boldsymbol{x}_i$. Using the same notation as before $\boldsymbol{S} = (S(\boldsymbol{x}_1), ..., S(\boldsymbol{x}_d))$ for the signal at sampling locations $\boldsymbol{x}_i$ for $i = 1, ..., d$, $\boldsymbol{S} \sim \text{MVN}\left(\boldsymbol{0}, \sigma^2\boldsymbol{R}\right)$ with $\boldsymbol{R}$ being the correlation matrix as defined in Section 2.2.3. Therefore, the marginal distribution of $S_i = S(\boldsymbol{x}_i)$ is a normal distribution with mean zero and variance σ^2.

For instance, if we consider the case where the response variable Y concerns counts then a sensible model may be the Poisson. In this case, the canonical link function is the logarithm and a general Poisson GLSM would be as follows,

$$\begin{aligned} Y_i|S_i &\sim \text{Poisson}\left(\mu_i\right), \\ \log(\mu_i) &= \boldsymbol{f}_i'\boldsymbol{\beta} + S_i = \eta_i. \end{aligned}$$

In the case of a Binomial GLSM using the logistic link function we would have,

$$\begin{aligned} Y_i|S_i &\sim \text{Binomial}\left(n_i, p_i\right), \\ \text{logit}(p_i) &= \log\left(\frac{p_i}{1-p_i}\right) = \boldsymbol{f}_i'\boldsymbol{\beta} + S_i = \eta_i. \end{aligned}$$

with n_i being the number of independent trials at location $\boldsymbol{x}_i$.

The likelihood for the GLSM is given by the d-fold integral

$$\begin{aligned} L(\boldsymbol{\beta}, \sigma^2, \phi) &= \int \prod_{i=1}^{d} f(y_i|s_i, \boldsymbol{\beta}) p(s_i|\sigma^2, \phi) d\boldsymbol{s} \\ &= \int \prod_{i=1}^{d} f(y_i|\eta_i) p(\eta_i|\boldsymbol{\beta}, \sigma^2, \phi) d\boldsymbol{\eta} \end{aligned} \tag{2.2.14}$$

As we see, the above likelihood can not be expressed analytically and the dependence between the components of $\boldsymbol{s}$ does not allow the likelihood to be expressed as the product of one-dimensional integrals. Under the Bayesian framework MCMC methods are used in order to make inference for the latent process and the parameters eliminating the need to evaluate integrals such as (2.2.14). In addition, the Bayesian approach provides a unified framework for estimation and prediction and naturally incorporates parameter uncertainty.

As a final note we want to stress a potential drawback of the model. Diggle et al. (1998), point out that the regression parameters should always be interpreted conditional on the process $\boldsymbol{S}$ since there can be confounding between the deterministic trend modelled by the regression parameters and the underlying $\boldsymbol{S}$. Reich et al. (2006) proved collinearity between the fixed and random effects of the conditional autoregressive model using arguments that are directly transferable to the GLSM and provided an alternative reparametrisation. This issue is also further studied in Hughes & Haran (2013) where additional reparametrisations of the model are suggested and achieving also reduction of the dimensionality of the random effects.

2.2.7 MCMC algorithms for inference on the GLSM

Under the Bayesian framework prior distributions will be assigned to each model parameter and interest will lie in sampling from the joint posterior distribution of $\boldsymbol{S}, \boldsymbol{\beta}, \sigma^2, \phi$. In the following we will denote the prior distributions by $p(\cdot)$ and the posterior distributions by $\pi(\cdot|\boldsymbol{y})$ therefore the joint posterior of the latent process and the parameters will be $\pi\left(\boldsymbol{S}, \boldsymbol{\beta}, \sigma^2, \phi|\boldsymbol{y}\right)$. Usually, it might be hard to elicit informative priors and researchers have resorted to the use of improper flat priors. Caution is needed in such cases since this can lead to an improper posterior distribution. It is known for example that an improper prior on ϕ will lead to an improper posterior $\pi(\phi|\boldsymbol{y})$ leading to invalid inferences. For a detailed discussion on this issue see Christensen et al. (2000) and references therein.

As we have discussed in Section 1.1 the main difficulties in constructing an efficient MCMC algorithm in order to sample from the joint posterior of the latent process and the parameters are the posterior dependence between the components of the latent process and also the dependence of the latent process and the model parameters. Papaspiliopoulos et al. (2003) elaborate on the issues of dependence between parameters

and the informativeness of the data for a variety of hierarchical models and show that the use of certain parametrisations appear to improve the mixing of MCMC schemes.

In the following, we present some of the existing MCMC algorithms that consider sampling from the joint posterior of the parameters and the latent process. Most of these schemes attempt to tackle some of the aforementioned problems by either tailoring the proposal distributions to match the shape of the posterior or employ various reparametrisations, mainly of $\boldsymbol{\beta}$ and $\boldsymbol{S}$, that aim to remove some of the prior or posterior dependence.

Diggle et al. (1998) assume uniform priors on $\boldsymbol{\theta} = \left(\sigma^2, \phi\right), \boldsymbol{\beta}$ and making use of the conditional independence structure of the parameters in the provide an MCMC scheme which updates $\boldsymbol{\theta}, \boldsymbol{\beta}$ and $\boldsymbol{S}$ using MH proposals. The constructed MCMC algorithm consists of the following three steps. Update $\boldsymbol{\theta}|\boldsymbol{S}$ using as a proposal distribution the prior distribution of $\boldsymbol{\theta}$, update all components of $\boldsymbol{S}$ through d univariate updates on $S_i|\boldsymbol{S}_{-i}, \boldsymbol{\theta}, \boldsymbol{\beta}, \boldsymbol{y}$ proposing values form their univariate prior normal distributions $p(S_i|\boldsymbol{S}_{-i}, \theta)$ and finally update $\boldsymbol{\beta}|\boldsymbol{S}, \boldsymbol{y}$. A practical drawback of this scheme is the computational cost. At each iteration of the algorithm the d univariate updates of $\boldsymbol{S}$ require the inversion of the $(d-1) \times (d-1)$ covariance matrix of $S_i|\boldsymbol{S}_{-i}$. Also, in cases where the components of $\boldsymbol{S}$ have strong posterior correlation, such updates might hinder the mixing of the MCMC.

Christensen, Moller & Waagepetersen (2001) study the property of geometric ergodicity of RWM and MALA updates on $\boldsymbol{S}$ for the Poisson GLSM and show that truncated MALA updates on a reparametrisation of $\boldsymbol{S}$ are more efficient. In particular they express the latent process as $\boldsymbol{S} = \boldsymbol{Q}\boldsymbol{\Gamma}$ where $\boldsymbol{Q}$ is the Cholesky square root of the prior covariance matrix of $\boldsymbol{S}$ and $\boldsymbol{\Gamma} \sim \text{MVN}(\boldsymbol{0}, \boldsymbol{I})$. Considering that all other parameters in the model

are fixed the gradient of the log-posterior with respect to $\boldsymbol{\gamma}$ is given by

$$\nabla(\boldsymbol{\gamma}) = \frac{\partial}{\partial \boldsymbol{\gamma}} \log\left[\pi(\boldsymbol{\gamma}|\boldsymbol{y})\right] = -\boldsymbol{\gamma} + \boldsymbol{Q}^{'}(\boldsymbol{y} - \boldsymbol{\mu}),$$

where $\boldsymbol{\mu} = \exp\{\boldsymbol{F\beta} + \boldsymbol{S}\} = \exp\{\boldsymbol{F\beta} + \boldsymbol{Q\gamma}\}$. Since the gradient above grows exponentially with $\boldsymbol{\gamma}$, through $\boldsymbol{\mu}$, to secure the geometric ergodicity they use the truncated gradient, $\nabla_t(\boldsymbol{\gamma})$,

$$\nabla_t(\boldsymbol{\gamma}) = -\boldsymbol{\gamma} + \boldsymbol{Q}^{'}\left(\boldsymbol{y} - \{\boldsymbol{\mu} \wedge H\}\right),$$

where H is the truncation constant and the minimum, $\wedge$, is applied component wise to each element of $\boldsymbol{\mu}$. In that way the authors avoid very extreme proposed jumps of $\boldsymbol{\gamma}$, as we have discussed in Section 2.1.4. As noted by the authors, we want the two gradients to be equal for most values of $\boldsymbol{\gamma}$ in the main body of the target distribution therefore they choose H to be at least two times bigger than the maximum observed count y. Therefore they update $\boldsymbol{\Gamma}$, which is effectively $\boldsymbol{S}$ with the prior dependence removed, using the proposal

$$\boldsymbol{\gamma}^{prop} \sim \text{MVN}\left(\boldsymbol{\gamma}^{cur} + \frac{\lambda}{2}\nabla_t(\boldsymbol{\gamma}^{cur}), \lambda \boldsymbol{I}\right)$$

and then transform back to $\boldsymbol{s} = \boldsymbol{Q\gamma}$. This parametrisation of $\boldsymbol{S}$ is actually a non-centered parametrisation in the context of Papaspiliopoulos et al. (2003) and Papaspiliopoulos et al. (2007) and is expected to perform better when the data are weak and the main contribution comes from the prior distribution of $\boldsymbol{S}|\boldsymbol{\beta}, \sigma^2, \phi$. Finally, the authors indicate that the property of geometric ergodicity can still be preserved when $\boldsymbol{\beta}$ is updated through RWM or MALA updates as long as it has a multivariate normal prior and the parameters σ^2 and ϕ are fixed.

This result is further used in Christensen & Waagepetersen (2002) where a full MCMC

scheme on $\boldsymbol{S}, \boldsymbol{\beta}, \sigma^2$ and ϕ is considered using independent informative priors for the model parameters obtained using previous analyses. In the constructed MCMC scheme one-dimensional RWM updates are implemented on $\log(\sigma)$ and $\log(\phi)$ whereas $\boldsymbol{S}$ is updated in one block using the truncated MALA proposal described above. With the only difference being that $\boldsymbol{Q}$ is the Cholesky square root of the prior correlation matrix of $\boldsymbol{S}$. As far as $\boldsymbol{\beta}$ is concerned they follow the same strategy as for $\boldsymbol{S}$ and standardise $\boldsymbol{\beta}$ with respect to its prior covariance matrix. In particular, the prior distribution for $\boldsymbol{\beta}$ is a multivariate normal with mean $\boldsymbol{m}_\beta$ and covariance matrix $\boldsymbol{\Sigma}_\beta$ and therefore $\boldsymbol{\beta}$ can be written as,

$$\boldsymbol{\beta} = \boldsymbol{m}_\beta + \boldsymbol{K}\boldsymbol{b} \tag{2.2.15}$$

where $\boldsymbol{K}$ is the Cholesky square root of $\boldsymbol{\Sigma}_\beta$ and $\boldsymbol{b} \sim \text{MVN}(\boldsymbol{0}, \boldsymbol{I})$. Hence the proposal of updating $\boldsymbol{b}$ is,

$$\boldsymbol{b}^{prop} \sim \text{MVN}\left(\boldsymbol{b}^{cur} + \frac{\lambda_b}{2}\nabla_t(\boldsymbol{b}^{cur}), \lambda_b \boldsymbol{I}\right)$$

where $\nabla_t(\boldsymbol{b})$ is the truncated version of $\nabla(\boldsymbol{b}) = \frac{\partial}{\partial \boldsymbol{b}} \log\left[\pi(\boldsymbol{b}|\boldsymbol{y})\right]$

$$\nabla_t(\boldsymbol{b}) = -\boldsymbol{b} + \boldsymbol{K}^{'}\boldsymbol{F}^{'}\left(\boldsymbol{y} - \{\boldsymbol{\mu} \wedge H\}\right).$$

Then, $\boldsymbol{\beta}$ is obtained using (2.2.15). The authors compare this MCMC scheme with an equivalent scheme where RWM updates are also used for $\boldsymbol{S}$ and $\boldsymbol{\beta}$ and argue in favour of the truncated MALA updates. A drawback of this scheme is the need to calculate the Cholesky decomposition of the prior covariance matrix of $\boldsymbol{S}$ each time that ϕ is updated.

Diggle et al. (2003) also adopt the above truncated MALA update for the latent process and combine it with the results on the LSM when using a conjugate prior for $\left(\boldsymbol{\beta}, \sigma^2\right)$. However they work in terms of $\boldsymbol{\eta}$ rather than $\boldsymbol{S}$ since in this case $\boldsymbol{Y}$ is conditionally

independent of $\boldsymbol{\beta}, \sigma^2, \phi$ given $\boldsymbol{\eta}$. More explicitly, using the same conjugate prior for $\left(\boldsymbol{\beta}, \sigma^2\right)$ as in (2.2.11) the marginal prior for $\boldsymbol{\eta}|\phi$,

$$p(\boldsymbol{\eta}|\phi) = \int p(\boldsymbol{\eta}|\boldsymbol{\beta}, \boldsymbol{\sigma^2}) p(\boldsymbol{y}|\boldsymbol{\eta}) d\boldsymbol{\beta} d\sigma^2 \tag{2.2.16}$$

results in the $d-$ dimensional t distribution, $t_{n_\sigma}\left(\boldsymbol{m}_\beta, S_\sigma^2\left(\boldsymbol{F}\boldsymbol{V}_\beta\boldsymbol{F}^{'} + \boldsymbol{R}\right)\right)$ and the conditional posterior of $\boldsymbol{\eta}|\phi$ is given by

$$\pi(\boldsymbol{\eta}|\boldsymbol{y}, \phi) \propto p(\boldsymbol{y}|\boldsymbol{\eta}) p(\boldsymbol{\eta}|\phi), \tag{2.2.17}$$

which, does not admit a closed form. Diggle et al. (2003) express $\boldsymbol{\eta}$ with respect to $\boldsymbol{\gamma}$ through $\boldsymbol{\eta} = \boldsymbol{F}^{'}\boldsymbol{m}_\beta + S_\sigma\left(\boldsymbol{R} + \boldsymbol{F}\boldsymbol{V}_\beta\boldsymbol{F}^{'}\right)^{1/2}\boldsymbol{\gamma}$, so that *á priori* $\boldsymbol{\gamma} \sim \mathrm{t}_{n+n_\sigma}(\boldsymbol{0}, I_n)$ and make use of the MALA update suggested by Christensen, Moller & Waagepetersen (2001) in order to update $\boldsymbol{\gamma}|\boldsymbol{y}, \phi$. The parameter ϕ is assigned a finite discrete prior, $p(\phi)$, in order to ease computations. Doing that, we are able to precompute the covariance matrix $\boldsymbol{F}\boldsymbol{V}_\beta\boldsymbol{F}^{'} + \boldsymbol{R}$ in advance, for all possible values of ϕ. The parameter ϕ is updated using a RWM with a normal proposal which is rounded to the nearest ϕ value in the discrete set of $p(\phi)$.

Now, the distribution of $(\boldsymbol{\beta}, \sigma^2|\boldsymbol{\eta}, \phi)$ is the normal Scaled Inverse χ^2 as given in (2.2.12) with the only difference that $\boldsymbol{\eta}$ is now substituted for $\boldsymbol{y}$. This is because we consider that $\tau^2 = 0$ and therefore the distribution of $\boldsymbol{Y}$ in the linear model is the same of that of $\boldsymbol{\eta}$ in the present case. Therefore, we can simulate directly from $\pi(\boldsymbol{\beta}, \sigma^2|\boldsymbol{\eta}, \boldsymbol{y})$ using the updated value of $\boldsymbol{\eta}|\boldsymbol{y}, \phi$. This framework gives flexibility by integrating out some of the model parameters and also ensures a proper posterior distribution. However, in cases where the correlation matrix is parametrised by more than one parameter the storage

requirements for precomputing the needed covariance matrices would be large.

A more sophisticated MCMC scheme is presented in Christensen et al. (2006) extending the ideas in Christensen & Waagepetersen (2002). The authors make a quadratic approximation of the posterior distribution $\pi(\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{\theta}, \boldsymbol{y})$ and remove much of the posterior dependence within $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ and also make $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ approximately orthogonal to σ^2 and ϕ. Effectively, their approach lies under the framework of partially non-centered parametrisations presented in Papaspiliopoulos et al. (2003) and Papaspiliopoulos et al. (2007).

To begin with, they work under the setting of $\boldsymbol{\eta} \sim \text{MVN}\ (\boldsymbol{F\beta}, \boldsymbol{\Sigma})$ where $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{R}$ and $\mathbb{E}[Y_i|\eta_i] = m_i h^{-1}(\eta_i)$, for $i = 1, ..., d$ with m_i being known scalars. We consider the case where the covariance parameters, $\boldsymbol{\theta} = (\phi, \sigma^2)$, are fixed since they do not affect the resulting transformations of $\boldsymbol{S}$ and $\boldsymbol{\beta}$. Let also $\boldsymbol{\beta} \sim \text{MVN}\,(\boldsymbol{m}_\beta, \Omega)$; then the log-posterior distribution of $(\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{\theta}, \boldsymbol{y})$ is given by,

$$\log\pi(\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{\theta}, \boldsymbol{y}) = \log\ f(\boldsymbol{y}|\boldsymbol{\eta}) + \log\ p(\boldsymbol{\eta}|\boldsymbol{\beta}, \boldsymbol{\theta}) + \log\ p(\boldsymbol{\beta}|\boldsymbol{\theta}) + \text{const.} \tag{2.2.18}$$

Using a Taylor expansion for $\log f(\boldsymbol{y}|\boldsymbol{\eta})$ around $\hat{\boldsymbol{\eta}}$ where $\hat{\eta}_i = \text{argmax} f(y_i|\eta_i)$ we obtain that,

$$\log f(\boldsymbol{y}|\boldsymbol{\eta}) \approx -\frac{1}{2}(\boldsymbol{\eta} - \hat{\boldsymbol{\eta}})' \Lambda(\hat{\boldsymbol{\eta}})(\boldsymbol{\eta} - \hat{\boldsymbol{\eta}}) + \text{constant}$$

where $\Lambda(\hat{\boldsymbol{\eta}})$ is a diagonal matrix with entries $-\frac{\partial^2}{\partial \eta_i^2} \log f(y_i|\eta_i)\big|_{\eta_i = \hat{\eta}_i}$. If we plug this approximation into (2.2.18) we can derive that the posterior of $\boldsymbol{\eta}$ is approximately multivariate normal with mean $\tilde{\boldsymbol{\Sigma}}\left(\Lambda\left(\hat{\boldsymbol{\eta}}\right)\hat{\boldsymbol{\eta}} + \boldsymbol{\Sigma}^{-1}\boldsymbol{F\beta}\right)$ and variance $\tilde{\boldsymbol{\Sigma}}$ where $\tilde{\boldsymbol{\Sigma}} = \left(\boldsymbol{\Sigma}^{-1} + \Lambda\left(\hat{\boldsymbol{\eta}}\right)\right)^{-1}$. Equivalently, we get that $\boldsymbol{\beta}$ is approximately normally distributed with mean given by $\tilde{\boldsymbol{\Omega}}\left(\boldsymbol{F}'\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{\Sigma}}\Lambda\left(\hat{\boldsymbol{\eta}}\right)\hat{\boldsymbol{\eta}} + \boldsymbol{\Omega}^{-1}\boldsymbol{b}\right)$ and variance $\tilde{\boldsymbol{\Omega}} = \left(\boldsymbol{\Omega}^{-1} + \boldsymbol{F}'\left(\Lambda\left(\hat{\boldsymbol{\eta}}\right)\boldsymbol{\Sigma} + I_n\right)^{-1}\Lambda\left(\hat{\boldsymbol{\eta}}\right)\boldsymbol{F}\right)^{-1}$.

This motivates the transformation of $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ to,

$$\tilde{\boldsymbol{S}} = \left(\tilde{\boldsymbol{\Sigma}}^{-1/2}\right)\left(\boldsymbol{\eta} - \tilde{\boldsymbol{\Sigma}}\left(\Lambda\left(\hat{\boldsymbol{\eta}}\right)\hat{\boldsymbol{\eta}} + \boldsymbol{\Sigma}^{-1}\boldsymbol{F}\boldsymbol{\beta}\right)\right) \tag{2.2.19}$$

$$\tilde{\boldsymbol{\beta}} = \left(\tilde{\boldsymbol{\Omega}}^{1/2}\right)^{-1}\left(\boldsymbol{\beta} - \tilde{\boldsymbol{\Omega}}\left(\boldsymbol{F}'\boldsymbol{\Sigma}^{-1}\tilde{\boldsymbol{\Sigma}}\Lambda\left(\hat{\boldsymbol{\eta}}\right)\hat{\boldsymbol{\eta}} + \boldsymbol{\Omega}^{-1}\boldsymbol{b}\right)\right) \tag{2.2.20}$$

so that $\tilde{\boldsymbol{S}}$ and $\tilde{\boldsymbol{\beta}}$ are multivariate standard normal variables consisting of uncorrelated components being also approximately uncorrelated with $\boldsymbol{\theta}$. The proposed algorithm for simulation from the joint posterior distribution $\pi(\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{\theta}, \boldsymbol{y})$ consists of two blocks using MALA proposals. The exact proposals are, $\tilde{\boldsymbol{\beta}}^{prop}|\tilde{\boldsymbol{S}}, \boldsymbol{\theta}, \boldsymbol{y} \sim \mathrm{N}\left(\tilde{\boldsymbol{\beta}}^{cur} + \frac{\delta_{\tilde{\boldsymbol{\beta}}}}{2}\xi(\tilde{\boldsymbol{\beta}}), \delta_{\tilde{\boldsymbol{\beta}}} I_n\right)$ and $\tilde{\boldsymbol{S}}^{prop}|\tilde{\boldsymbol{\beta}}, \boldsymbol{\theta}, \boldsymbol{y} \sim \mathrm{N}\left(\tilde{\boldsymbol{S}}^{cur} + \frac{\delta_{\tilde{\boldsymbol{S}}}}{2}\xi(\tilde{\boldsymbol{S}}^{cur}), \delta_{\tilde{\boldsymbol{S}}} I_n\right)$. Where $\xi(\tilde{\boldsymbol{S}}) = \tilde{\boldsymbol{S}} + \nabla \log \pi(\tilde{\boldsymbol{S}}|\tilde{\boldsymbol{\beta}}, \boldsymbol{\theta}, \boldsymbol{y})$ and $\xi(\tilde{\boldsymbol{\beta}}) = \tilde{\boldsymbol{\beta}} + \nabla \log \pi(\tilde{\boldsymbol{\beta}}|\boldsymbol{\theta}, \boldsymbol{y})$ After each update, the current values of $\boldsymbol{\eta}$ and $\boldsymbol{\beta}$ are obtained using (2.2.19) and (2.2.20).

In the case where the correlation parameters $\boldsymbol{\theta} = (\sigma, \phi)$ are unknown then one just has to add the logarithm of their prior distribution, $\log\pi(\theta|\boldsymbol{y})$, in (2.2.18) and the sampling algorithm would further include two one-dimensional RWM steps. Based on the fact that these two parameters usually exhibit posterior dependence and that their posterior distributions can be heavily skewed the authors suggest to update $\tilde{\theta}_1 = \log(\sigma)$ and $\tilde{\theta}_2 = 2\log(\sigma) - \log(\phi)$ instead.

The approach of Girolami & Calderhead (2011) has been briefly described in Section 2.1.6. In the following we provide the exact proposal distribution used for updating the latent process $\boldsymbol{\eta}$ conditionally on the parameters, focusing on the MMALA rather than the RMHMC. In their example, they consider the case $Y_i|\eta_i \sim \text{Poisson}\left(m\exp(\eta_i)\right)$ and $\boldsymbol{\eta} \sim \text{MVN}\left(\boldsymbol{\mu}_\eta, \boldsymbol{\Sigma}\left(\sigma^2, \phi\right)\right)$ where m is a known scalar and $\boldsymbol{\mu}_\eta = \mu_\eta \mathbf{1}_{d\times 1}$ with μ_η being a constant mean. Let $\boldsymbol{\theta} = \left(\boldsymbol{\mu}_\eta, \sigma^2, \phi\right)$. In order to construct the preconditioning matrix,

$\boldsymbol{M}$, the authors take the expectation of the second derivative of the log posterior over both the data and the latent process, i.e.,

$$\boldsymbol{M} = -\mathbb{E}_{y,\boldsymbol{\eta}|\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial\boldsymbol{\eta}\partial\boldsymbol{\eta}'}\log\left\{\pi(\boldsymbol{\eta}|\boldsymbol{\theta},\boldsymbol{y})\right\}\right]^{-1} = \left(\boldsymbol{\Lambda}+\boldsymbol{\Sigma}^{-1}\right)^{-1},$$

where $\boldsymbol{\Lambda}$ is a $d\times d$ diagonal matrix with diagonal elements $\mathbb{E}\left[\exp(\eta_i)\right] = \exp\{\mu_\eta + \frac{1}{2}\boldsymbol{\Sigma}_{ii}\}$ for $i = 1, ..., d$. In that way, the curvature of the random field is constant and the matrix $\boldsymbol{M}$ does not depend on $\boldsymbol{\eta}$. Therefore the full MMALA for the latent process reduces to a simple MALA update with a fixed preconditioning matrix $\boldsymbol{M}$. The exact proposal distribution is therefore,

$$\boldsymbol{\eta}^{prop} \sim \text{MVN}\left(\boldsymbol{\eta}^{cur} + \frac{\lambda}{2}\boldsymbol{M}\nabla\log\pi(\boldsymbol{\eta}^{cur}|\boldsymbol{\theta},\boldsymbol{y}), \lambda\boldsymbol{M}\right).$$

In Chapter 3 we will refer to this simple preconditioned MALA as pMMALA. However, based on what was introduced in Section 2.1.6 the preconditioning matrix should be constructed based on,

$$\boldsymbol{M}(\boldsymbol{\eta}) = -\mathbb{E}_{y|\boldsymbol{\eta},\boldsymbol{\theta}}\left[\frac{\partial^2}{\partial\boldsymbol{\eta}\partial\boldsymbol{\eta}'}\log\left\{\pi(\boldsymbol{\eta}|\boldsymbol{\theta},\boldsymbol{y})\right\}\right]^{-1} = \left(\boldsymbol{\Lambda}+\boldsymbol{\Sigma}^{-1}\right)^{-1},$$

where in this case $\boldsymbol{\Lambda}$ is a $d\times d$ diagonal matrix with diagonal elements $\exp\{\eta_i\}$ for $i = 1, ..., d$. In that way, the proposal would use a position specific preconditioning matrix since it would depend on $\boldsymbol{\eta}$ resulting in the sMMALA.

Haran & Tierney (2012) consider the construction of a simple MCMC scheme and in particular a MHIS in order to make inference on a model similar to the Poisson GLSM. The fundamental difference between their model and the GLSM is that in the former, the spatial dependence is modelled through a Gaussian Markov random field. Similar

to the motivation of Chapter 3 of this thesis, their suggestion is to employ a Gaussian approximation to the posterior of interest and use a heavier-tailed version of it as a proposal in a MHIS. Their algorithm shares similarities with the algorithm L1 that we propose in Section 3.1.2 including the exact transformation used on the data in order to achieve the Gaussian approximation. However, we believe that the information provided by the authors regrading the performance of the algorithm is limited and concrete conclusions can not be drawn.

Giorgi et al. (2015) share ideas from Christensen (2004) and construct an algorithm that combines Monte Carlo maximum likelihood techniques and MCMC sampling. For instance, estimates for the parameters of the model are obtained using Monte Carlo maximum likelihood and the latent process conditional on the parameters is updated using the parametrisation of Christensen et al. (2006).

CHAPTER 3

Single block MHIS proposals for the latent variables in a GLSM

Throughout this thesis we focus on the Poisson GLSM, a model with correlated Gaussian latent variables, and consider inference on the latent variables using the MH algorithm. In this Chapter we consider all latent variables being updated as a single block. We demonstrate the full derivation of our proposals and their performance in this particular case.

Throughout this Chapter it is assumed that the covariance parameters (σ^2, ϕ) from (2.2.2) are known and that we wish to perform inference for the Gaussian latent process and the mean parameters $\boldsymbol{\beta}$ from (2.2.13). In practice one usually wishes to perform inference on the joint distribution of all of the parameters and the Gaussian process. However, MCMC algorithms typically alternate an update of the Gaussian process given the covariance parameters with an update of the covariance parameters given the Gaussian process. The focus of this thesis is on improving the former step, and this is why

we assume that the covariance parameters are known.

In particular, we explore the idea of applying a Gaussian approximation to the density of a transformation of the data, conditional on the Gaussian process. This enables us to work under the framework of the linear Gaussian model, where the form of the posterior is tractable. In that way we are able to find an approximate posterior distribution for the, potentially transformed, latent variables given the transformed data, and use this as a proposal in our MCMC algorithm. We correct for the approximation with the usual MH accept-reject step. Additionally, non-linear terms of the distribution of the latent process are replaced with fixed values based on the data, resulting in an MHIS. The algorithms presented in this chapter require no tuning and they also automatically select initial values for the latent process.

In Section 3.1.1 we outline a general algorithm based on the link function which can be implemented on any GLSM, and show the exact form of the proposal for a Poisson GLSM in Section 3.1.3. Thereafter, the constructed proposals focus on the Poisson GLSM and in Section 3.1.4 we provide a further transformation which attempts to reduce the error in the approximations to the mean and variance of the transformed data. Section 3.2.1, illustrates the use of Anscombe's transformation on the data which results in a different proposal, and in Section 3.2.3 we present some further approximations for the expected value of the transformed data. Finally, in Section 3.3 we compare our suggested algorithms against those already existing in the literature.

3.1 The link function transformation

3.1.1 A general algorithm

Let the d-dimensional vector $\boldsymbol{Y}$ be the response variable arising from a distribution within the exponential family and suppose that conditional on the parameters and latent variables its mean is modelled as

$$h(\mathrm{E}[\boldsymbol{Y}|\boldsymbol{\eta}]) = h(\boldsymbol{\mu}) = \boldsymbol{\eta} = \boldsymbol{F}\boldsymbol{\beta} + \boldsymbol{S},$$

where $h()$ is the link function, $\boldsymbol{F}$ is the design matrix, $\boldsymbol{\beta}$ the vector of regression coefficients and $\boldsymbol{S}$ is the spatial process having the following priors,

$$\boldsymbol{\beta} \sim \mathrm{MVN}(\boldsymbol{\mu}_\beta, \sigma^2\boldsymbol{\Sigma}_\beta)$$

$$\boldsymbol{S} \sim \mathrm{MVN}(\boldsymbol{0}, \sigma^2\boldsymbol{R}),$$

where the correlation matrix, $\boldsymbol{R}$, depends on the parameter ϕ as introduced in Section 2.2.2.

Consider a random variable such as a Poisson random variable with a large mean, or a binomial random variable with a large number of trials and a success probability not close to 0 or 1. Such a random variable has a distribution which is close to normal and, moreover its standard deviation is much less than its mean. Hence, any suitably well-behaved transformation of it will also have a distribution which is approximately normal, and for a reasonably accurate description of this random variable it should be sufficient to obtain its mean and variance.

Consider now the transformed variable $\boldsymbol{Y}^l = h(\boldsymbol{Y})$. In order to obtain a normal ap-

proximation of the likelihood we use the Delta method, i.e., we first approximate the moments of $\boldsymbol{Y}^l|\boldsymbol{\eta}$ using a second order Taylor expansion about $\boldsymbol{\mu} := \mathbb{E}(\boldsymbol{Y})$, (see, for example, Casella & Berger (1990)). For ease of notation, since $Y_i|\eta_i$, $i = 1, ..., n$ are independent, we illustrate the Taylor expansion in the univariate case and drop the subscript i. Thus, denoting $\mathbb{E}[Y_i|\eta_i]$ by $\mu(\eta)$, we have,

$$\begin{aligned} Y^l|\eta &\approx h(\mu) + (Y-\mu)\frac{\partial h(\mu)}{\partial \mu} + \frac{1}{2}(Y-\mu)^2\frac{\partial^2 h(\mu)}{\partial \mu^2}| \\ &= \eta + (Y-\mu)\frac{\partial h(\mu)}{\partial \mu} + \frac{1}{2}(Y-\mu)^2\frac{\partial^2 h(\mu)}{\partial \mu^2}, \end{aligned} \tag{3.1.1}$$

evaluated at $\mu = \mu(\eta)$. In the above expression we have neglected the term $\frac{1}{6}(Y-\mu)^3\frac{\partial^3 h(\mu)}{\partial \mu^3}\Big|_{\mu=\mu(\eta)+t(Y-\mu(\eta))}$ for some $t(\eta, Y) \in [0,1]$.

It is crucial to the efficiency, but not to the accuracy, of our technique that each successive term in this Taylor expansion is small in comparison with the previous term. The probability that each term is negligible compared to the previous one, tends to one as the probability that an observation is, in some sense, closer to the mean tends to one. For the Poisson model this occurs as the mean, $\mu \to \infty$ whereas for a Binomial, $B(n,p)$, model this occurs as $n \to \infty$ provided $p \in (0,1)$.

The expected value of $Y_i^l|\eta_i$ is approximated by

$$\mathrm{E}[Y_i^l|\eta_i] = \eta_i + \frac{1}{2}\frac{\partial^2 h(\mu_i)}{\partial \mu_i^2}\mathrm{E}[(Y_i-\mu_i)^2] = \eta_i + \frac{1}{2}\frac{\partial^2 h(\mu_i)}{\partial \mu_i^2}\mathrm{Var}[Y_i|\eta_i] = \eta_i + \mu_i^*, \tag{3.1.2}$$

and taking into account only terms up to first order, the variance can be approximated

by

$$\text{Var}[Y_i^l|\eta_i] = \left[\frac{\partial h(\mu_i)}{\partial \mu_i}\right]^2 \text{Var}[Y_i|\eta_i] = \Sigma_{ii}^*. \tag{3.1.3}$$

Therefore, assuming for now that μ_i^*, Σ_{ii}^* are known, and assuming a Gaussian distribution for each Y_i^l, we have that approximately $Y_i^l|\eta_i \sim \text{N}(\eta_i + \mu_i^*, \Sigma_{ii}^*)$. Extending the above approximation to the multivariate case, we can obtain the following approximation for the joint distribution of $(\boldsymbol{Y}^l, \boldsymbol{\eta})$

$$\begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{Y}^l \end{bmatrix} \sim \text{MVN}\left(\begin{bmatrix} \boldsymbol{F\mu}_\beta \\ \boldsymbol{F\mu}_\beta + \boldsymbol{\mu}^* \end{bmatrix}, \begin{bmatrix} \sigma^2\left(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}\right) & \sigma^2\left(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}\right) \\ \sigma^2\left(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}\right)' & \sigma^2\left(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}\right) + \boldsymbol{\Sigma}^* \end{bmatrix}\right), \tag{3.1.4}$$

where $\boldsymbol{\mu}^*$ is a vector with elements μ_i^* and $\boldsymbol{\Sigma^*}$ is a diagonal matrix with diagonal elements Σ_{ii}^*.

If 3.1.4 were true, then the distribution of $\boldsymbol{\eta}|\boldsymbol{Y}^l$ would be

$$\text{MVN}\left(\boldsymbol{\mu}_{\eta|y^l}, \boldsymbol{\Sigma}_{\eta|y^l}\right) \tag{3.1.5}$$

(see, for example, Diggle et al. (2007)); where,

$$\boldsymbol{\mu}_{\eta|y^l} = \boldsymbol{F\mu}_\beta + \sigma^2(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R})\left[\sigma^2(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}) + \boldsymbol{\Sigma}^*\right]^{-1}(\boldsymbol{y}^l - \boldsymbol{F\mu}_\beta - \boldsymbol{\mu}^*), \tag{3.1.6}$$

$$\boldsymbol{\Sigma}_{\eta|y^l} = \sigma^2(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}) - \sigma^2(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R})\left[\sigma^2(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}) + \boldsymbol{\Sigma}^*\right]^{-1}\sigma^2(\boldsymbol{F\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}). \tag{3.1.7}$$

Our suggestion is to use this conditional distribution as a proposal for $\boldsymbol{\eta}$ in our MCMC algorithm in order to draw samples from the posterior distribution of $\boldsymbol{\eta}|\boldsymbol{y}$. However, $\boldsymbol{\mu}^*$ and $\boldsymbol{\Sigma}^*$ are functions of $\boldsymbol{\eta}$ and are unknown. Since, though, $\text{E}[\boldsymbol{Y}] = h^{-1}(\boldsymbol{\eta})$, we

approximate $\boldsymbol{\mu}^*(\boldsymbol{\eta})$, $\boldsymbol{\Sigma}^*(\boldsymbol{\eta})$ with $\boldsymbol{\mu}^*(h(\boldsymbol{Y}))$ and $\boldsymbol{\Sigma}^*(h(\boldsymbol{Y}))$.

Furthermore, as far as the regression parameter $\boldsymbol{\beta}$ is concerned we have that

$$\begin{bmatrix} \boldsymbol{\eta} \\ \boldsymbol{\beta} \end{bmatrix} \sim \text{MVN}\left(\begin{bmatrix} \boldsymbol{F}\boldsymbol{\mu}_\beta \\ \boldsymbol{\mu}_\beta \end{bmatrix}, \begin{bmatrix} \sigma^2\left(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R}\right) & \sigma^2\boldsymbol{F}\boldsymbol{\Sigma}_\beta \\ \sigma^2\boldsymbol{\Sigma}_\beta\boldsymbol{F}' & \sigma^2\boldsymbol{\Sigma}_\beta \end{bmatrix} \right). \tag{3.1.8}$$

Thus,

$$\boldsymbol{\beta}|\boldsymbol{\eta} \sim \text{MVN}(\boldsymbol{\mu}_{\beta|\eta}, \boldsymbol{\Sigma}_{\beta|\eta}), \tag{3.1.9}$$

where,

$$\boldsymbol{\mu}_{\beta|\eta} = \boldsymbol{\mu}_\beta + \sigma^2\boldsymbol{\Sigma}_\beta\boldsymbol{F}' \left[\sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R})\right]^{-1}(\boldsymbol{\eta} - \boldsymbol{F}\boldsymbol{\mu}_\beta)$$

$$\boldsymbol{\Sigma}_{\beta|\eta} = \sigma^2\boldsymbol{\Sigma}_\beta - \sigma^2\boldsymbol{\Sigma}_\beta\boldsymbol{F}' \left[\sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}' + \boldsymbol{R})\right]^{-1}\sigma^2\boldsymbol{F}\boldsymbol{\Sigma}_\beta \tag{3.1.10}$$

Hence, having sampled from $\boldsymbol{\eta}|\boldsymbol{y}$ we can use the updated value of $\boldsymbol{\eta}$ in order to sample exactly from the posterior distribution of $\boldsymbol{\beta}|\boldsymbol{\eta}, \boldsymbol{y}$.

3.1.2 The algorithm (L1)

We illustrate the resulting MHIS algorithm, L1, that draws samples from the posterior distribution of $\boldsymbol{\eta}, \boldsymbol{\beta}|\boldsymbol{y}$. Let, $\boldsymbol{\eta}^{prop}$, $\boldsymbol{\eta}^{cur}$ be the proposed and current value of the latent process $\boldsymbol{\eta}$. Since, $\boldsymbol{y}^l = h(\boldsymbol{y})$ is substituted for $\boldsymbol{\eta}$, the proposal distribution derived from (3.1.5) and (3.1.7) will not depend on $\boldsymbol{\eta}^{cur}$. Therefore, the resulting algorithm will be an MHIS and we denote the proposal density by $q^*(\boldsymbol{\eta}|\boldsymbol{y}^l)$. Then the algorithm reads,

- Set initial values $\boldsymbol{\eta}^{cur} = \boldsymbol{\eta}^{(0)}$, $i = 1$.
- Propose, $\boldsymbol{\eta}^{prop}$ according to (3.1.5) – (3.1.7)

- Accept with probability,

$$\alpha(\boldsymbol{\eta}^{cur}, \boldsymbol{\eta}^{prop}) = \min\left(1, \frac{\pi(\boldsymbol{\eta}^{prop}|\boldsymbol{y})q^*(\boldsymbol{\eta}^{cur}|\boldsymbol{y}^l)}{\pi(\boldsymbol{\eta}^{cur}|\boldsymbol{y})q^*(\boldsymbol{\eta}^{prop}|\boldsymbol{y}^l)}\right).$$

- If $\boldsymbol{\eta}^{prop}$ is accepted, set $\boldsymbol{\eta}^{cur} = \boldsymbol{\eta}^{prop}$.

- Store $\boldsymbol{\eta}^{(i)} = \boldsymbol{\eta}^{cur}$; set $i = i + 1$.

In practice, in order to ensure that the algorithm is geometrically ergodic, the multivariate normal proposal for $\boldsymbol{\eta}$ is replaced with a d-dimensional multivariate Student's t_ν-distribution with density given by

$$q^*(\boldsymbol{\eta}|\boldsymbol{y}^l) \propto \left(1 + \frac{1}{\nu}\left(\boldsymbol{\eta} - \boldsymbol{\mu}_{\eta|y^l}\right)' \boldsymbol{\Sigma}^{-1}_{\eta|y^l}\left(\boldsymbol{\eta} - \boldsymbol{\mu}_{\eta|y^l}\right)\right)^{-\frac{\nu+d}{2}} \tag{3.1.11}$$

After the MCMC algorithm has completed I iterations, a sample for $\boldsymbol{\beta}$ can be drawn, through the following step,

- Using each $\boldsymbol{\eta}^{(i)}$, simulate $\boldsymbol{\beta}^{(i)}$ using (3.1.9) – (3.1.10)

3.1.3 Example: Poisson GLSM

We will now consider a Poisson GLSM and illustrate the exact form of the proposal distribution derived in the previous section. More explicitly, we have that $Y_i|\eta_i \sim \text{Poisson}(\mu_i)$ where $\mu_i = e^{\eta_i}$. The canonical link function is,

$$h(\mu_i) = \log(\mu_i) = \eta_i = \boldsymbol{f}_i'\boldsymbol{\beta} + s_i \quad \text{for} \quad i = 1, ..., n. \tag{3.1.12}$$

Therefore,

$$\frac{\partial}{\partial \mu_i} h(\mu_i) = e^{-\eta_i}, \quad \frac{\partial^2}{\partial \mu_i^2} h(\mu_i) = -e^{-2\eta_i}.$$

Combining these expressions with (3.1.2) and (3.1.3) the mean and variance of $Y_i^l|\eta_i$ are

$$\mathrm{E}[Y_i^l|\eta_i] = \eta_i - \frac{1}{2}e^{-\eta_i}, \quad \mathrm{Var}[Y_i^l|\eta_i] = \mathbf{\Sigma}_{ii}^* = e^{-\eta_i}. \tag{3.1.13}$$

Thus, $\mu_i^* = -\frac{1}{2}e^{-\eta_i}$, and substituting $\eta_i = y_i^l = \log(y_i)$ for the non linear terms we obtain

$$\mu_i^* = -\frac{1}{2e^{y_i^l}} = -\frac{1}{2y_i}, \quad \mathbf{\Sigma}_{ii}^* = \frac{1}{e^{y_i^l}} = \frac{1}{y_i}. \tag{3.1.14}$$

A word of caution is needed for the Poisson GLSM when using the canonical link function since $\log(y_i) = -\infty$ if $y_i = 0$. In this case, we substitute $y_i = 0.5$ when taking the logarithm. This is supported by the general argument provided in Section 3.1.4 and is also supported by Haran & Tierney (2012).

As a final comment, when the above algorithm was implemented we found that ignoring the correction term μ_i^* did not noticeably alter the performance of the algorithm. Furthermore, a MH version of the algorithm was explored where η_i in the terms μ_i^* and $\mathbf{\Sigma}^*$ was replaced with the current value of the process rather than using the approximation $\eta_i \approx y_i^l$. However, the gain in efficiency was minor, yet the extra computational cost was very high since the mean and the covariance matrix of the proposal distribution had to be calculated at every iteration. Therefore, the results presented in Section 3.3 correspond to the algorithm illustrated above ignoring the correction term μ_i^*.

3.1.4 An alternative approximation for the Poisson GLSM (L2)

In Section 3.1.1 we outlined a general method for constructing the proposal distribution which applies to any GLSM. In this section we provide a further approximation for the case of a Poisson GLSM through which the leading error term in the Taylor expansion

for the expectation of the transformed data disappears.

Let $Y_i \sim \text{Poisson}(\mu_i)$, where in our case $\mu_i = e^{\eta_i}$, with η_i being the linear predictor as defined in 3.1.12. For ease of notation we ignore the subscript i and define the transformed variable $Y^p = \log(Y + \alpha)$. If we set,

$$T := \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \frac{(Y-\mu)^k}{(\mu+\alpha)^k}, \tag{3.1.15}$$

then under the assumption that $\left|\frac{Y-\mu}{\mu+\alpha}\right| < 1$, Y^p can be approximated through a Taylor expansion of Y about μ by,

$$\begin{aligned} Y^p &= \log(Y+\alpha) = \log(Y-\mu+\alpha+\mu) \\ &= \log(\mu+\alpha) - T \qquad (3.1.16) \\ &= \log(\mu) + \log\left(1+\frac{\alpha}{\mu}\right) - T \\ &\approx \log(\mu) + \left(\frac{\alpha}{\mu} - \frac{\alpha^2}{2\mu^2} + \frac{\alpha^3}{3\mu^3}\right) - T. \qquad (3.1.17) \end{aligned}$$

Now, consider the expectation of Y^p given by,

$$\mathbb{E}\left[Y^p\right] = \log(\mu+\alpha) - \mathbb{E}\left[T\right].$$

We want the accuracy of our approximation to be of order $\mathcal{O}\left(\mu^{-2}\right)$ and we therefore have to include all terms of T up to $k = 4$. To see this, recall that the first five central

moments of a Poisson random variable Y with mean μ are given by,

$$\mathbb{E}\left[(Y-\mu)^1\right] = 0, \quad \mathbb{E}\left[(Y-\mu)^2\right] = \mathbb{E}\left[(Y-\mu)^3\right] = \mu,$$

$$\mathbb{E}\left[(Y-\mu)^4\right] = 3\mu^2+\mu, \quad \mathbb{E}\left[(Y-\mu)^5\right] = 10\mu^2+\mathcal{O}(\mu).$$

Hence, the expectation of T is given by,

$$\begin{aligned}
\mathbb{E}[T] &= \sum_{k=1}^{\infty} \frac{(-1)^k}{k} \frac{\mathbb{E}\left[(Y-\mu)^k\right]}{(\mu+\alpha)^k} \\
&\approx \frac{\mu}{2(\mu+\alpha)^2} - \frac{\mu}{3(\mu+\alpha)^3} + \frac{3\mu^2+\mu}{4(\mu+\alpha)^4} + \mathcal{O}\left(\mu^{-3}\right) \\
&\approx \frac{1}{2\mu}\left(1-2\frac{\alpha}{\mu}\right) - \frac{1}{3\mu^2}\left(1-3\frac{\alpha}{\mu}\right) + \frac{1}{4\mu^3}\left(3\mu-12\alpha+1-\frac{4\alpha}{\mu}\right) + \mathcal{O}\left(\mu^{-3}\right) \\
&\approx \frac{1}{2\mu} + \frac{(5-12\alpha)}{12\mu^2} + \mathcal{O}\left(\mu^{-3}\right). \qquad (3.1.18)
\end{aligned}$$

Substituting now back to (3.1.17) we obtain the expectation of Y^p through,

$$\begin{aligned}
\mathbb{E}[Y^p] &\approx \log(\mu) + \left(\frac{\alpha}{\mu} - \frac{\alpha^2}{2\mu^2} + \frac{\alpha^3}{3\mu^3}\right) - \frac{1}{2\mu} - \frac{(5-12\alpha)}{12\mu^2} + \mathcal{O}\left(\mu^{-3}\right) \\
&\approx \eta + \frac{1}{\mu}\left(\alpha-\frac{1}{2}\right) + \frac{1}{\mu^2}\left(\alpha-\frac{\alpha^2}{2}-\frac{5}{12}\right) + \mathcal{O}\left(\mu^{-3}\right).
\end{aligned}$$

For the calculation of the variance of Y^p, consider the form of Y^p in (3.1.16) for simplicity

and notice that

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[Y^p\right] &= \mathbb{E}\left[\left(Y^p\right)^2\right] - \mathbb{E}^2\left[Y^p\right] \\ &= \mathbb{E}\left[T^2\right] - \mathbb{E}^2\left[T\right] = \mathbb{V}\mathrm{ar}\left[T\right]. \end{aligned} \tag{3.1.19}$$

Hence, we only need to calculate the expectation of T^2,

$$\mathbb{E}\left[T^2\right] = \sum_{k,l=1}^{\infty} \frac{(-1)^{k+l}}{kl} \frac{\mathbb{E}\left[(Y-\mu)^{k+l}\right]}{(\mu+\alpha)^{k+l}} \tag{3.1.20}$$

To account for all $\mathcal{O}\left(\mu^{-2}\right)$ terms we evaluate $\mathbb{E}\left[T^2\right]$ for all pairs (k,l) such that $k+l \leq 4$ and obtain that,

$$\begin{aligned}\mathbb{E}\left[T^2\right] &= \frac{\mathbb{E}\left[(Y-\mu)^2\right]}{(\mu+\alpha)^2} - \frac{\mathbb{E}\left[(Y-\mu)^3\right]}{(\mu+\alpha)^3} + \frac{11}{12}\frac{\mathbb{E}\left[(Y-\mu)^4\right]}{(\mu+\alpha)^4} + \mathcal{O}\left(\mu^{-3}\right) \\ &= \frac{1}{\mu}\left(1-2\frac{\alpha}{\mu}\right) - \frac{1}{\mu^2}\left(1-3\frac{\alpha}{\mu}\right) + \frac{11}{12\mu^4}\left(3\mu^2+\mu\right)\left(1-4\frac{\alpha}{\mu}\right) + \mathcal{O}\left(\mu^{-3}\right) \\ &= \frac{1}{\mu} + \frac{1}{4\mu^2}\left(7-8\alpha\right) + \mathcal{O}\left(\mu^{-3}\right)\end{aligned}$$

Combining this expression with (3.1.18) and (3.1.19)

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left[Y^p\right] &= \mathbb{E}\left[T^2\right] - \mathbb{E}^2\left[T\right] \\ &= \frac{1}{\mu} + \frac{1}{4\mu^2}\left(7-8\alpha\right) - \frac{1}{4\mu^2} + \mathcal{O}\left(\mu^{-3}\right) \\ &= \frac{1}{\mu} + \frac{1}{4\mu^2}\left(6-8\alpha\right) + \mathcal{O}\left(\mu^{-3}\right).\end{aligned}$$

Notice that when $\alpha = 0$,

$$\mathbb{E}\left[Y^p\right] \approx \eta - \frac{1}{2\mu} - \frac{5}{12\mu^2} + o\left(\mu^{-2}\right), \quad \mathbb{V}\mathrm{ar}\left[Y^p\right] \approx \frac{1}{\mu} + \frac{3}{2\mu^2} + \mathcal{O}(\mu^{-3}),$$

and when $\alpha = 0.5$,

$$\mathbb{E}\left[Y^p\right] \approx \eta - \frac{1}{24\mu^2} + o\left(\mu^{-2}\right), \quad \mathbb{V}\mathrm{ar}\left[Y^p\right] \approx \frac{1}{\mu} + \frac{1}{2\mu^2} + \mathcal{O}(\mu^{-3}). \tag{3.1.21}$$

Hence, by choosing $\alpha = 0.5$, the $\mathcal{O}(\mu^{-1})$ term in the expectation disappears and the $\mathcal{O}(\mu^{-2})$ reduces by an order of magnitude. Moreover, in the variance the $\mathcal{O}(\mu^{-2})$ term also becomes smaller. Therefore, the choice of $\alpha = 0.5$ leads to smaller errors in the approximations $\mathbb{E}[Y^p] \approx \eta$ and $\mathbb{V}\mathrm{ar}[Y^p] \approx \frac{1}{\mu}$.

Extending the approximation made in Section 3.1.1, we wish to approximate the variance of Y^p by a function of the form $\frac{1}{Y+\beta}$ and would like this to be unbiased to $o(\mu^{-2})$. Consider, therefore,

$$(Y+\beta)^{-1} = \frac{1}{\mu+\beta}\left(1 + \frac{Y-\mu}{\mu+\beta}\right)^{-1} = \frac{1}{\mu+\beta}\sum_{k=0}^{\infty}(-1)^k\frac{(Y-\mu)^k}{(\mu+\beta)^k}.$$

This has expectation given by,

$$\begin{aligned}
\mathbb{E}\left[(Y+\beta)^{-1}\right] &\approx \frac{1}{\mu+\beta}\left(1+\frac{\mu}{(\mu+\beta)^2}\right)+\mathcal{O}(\mu^{-3}) \\
&= \frac{1}{\mu+\beta}+\frac{\mu}{(\mu+\beta)^{-3}}+\mathcal{O}(\mu^{-3}) \\
&= \mu^{-1}\left(1+\frac{\beta}{\mu}\right)^{-1}+\mu^{-2}\left(1+\frac{\beta}{\mu}\right)^{-3}+\mathcal{O}(\mu^{-3}) \\
&= \frac{1}{\mu}+\frac{1-\beta}{\mu^2}+\mathcal{O}(\mu^{-3}).
\end{aligned}$$

Comparing now this with the variance given in (3.1.21) we see that,

$$\frac{1}{\mu}+\frac{1}{2\mu^2}=\frac{1}{\mu}+\frac{1-\beta}{\mu^2}\Leftrightarrow\beta=\frac{1}{2},$$

which indicates that the bias in our choice of $\frac{1}{Y+0.5}$ for the variance is of order $o(\mu^{-2})$.

Hence, we approximate the distribution of $Y_i^p|\eta_i$ by,

$$Y_i^p|\eta_i \sim \mathrm{N}\left(\eta_i, \quad \frac{1}{y_i+0.5}\right)$$

Extending to the multivariate case and taking into account the structure of the GLSM we can approximate the joint distribution of the vector random variables $(\boldsymbol{\eta}, \boldsymbol{Y}^p)$ through a multivariate normal distribution similar to the one obtained from (3.1.4). The only differences, now, are that firstly there is no first order correction term $\boldsymbol{\mu}^*$ and secondly, the diagonal elements of the covariance matrix $\boldsymbol{\Sigma}^*$ will be of the form $1/(y_i+0.5)$. The proposal for our MCMC scheme is therefore, $\boldsymbol{\eta}|\boldsymbol{Y}^p \sim \mathrm{MVN}\left(\boldsymbol{\mu}_{\eta|y^p}, \boldsymbol{\Sigma}_{\eta|y^p}\right)$ with mean

and variance given by

$$\boldsymbol{\mu}_{\eta|y^p} = \boldsymbol{F}\boldsymbol{\mu}_\beta + \sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^{'} + \boldsymbol{R})\left[\sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^{'} + \boldsymbol{R}) + \boldsymbol{\Sigma}^*\right]^{-1}(\boldsymbol{y}^p - \boldsymbol{F}\boldsymbol{\mu}_\beta)$$

$$\boldsymbol{\Sigma}_{\eta|y^p} = \sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^{'} + \boldsymbol{R}) - \sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^{'} + \boldsymbol{R})\left[\sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^{'} + \boldsymbol{R}) + \boldsymbol{\Sigma}^*\right]^{-1}\sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^{'} + \boldsymbol{R}).$$

3.2 Using Anscombe's transformation for the Poisson GLSM

The two main characteristics of the normal distribution are symmetry and independence between variance and mean. This is in contrast with the Poisson or Binomial distributions which are skewed and where mean and variance are related.

So far, in the case of a Poisson GLSM, we have employed the logarithmic transformation of the data in order to obtain an approximate Gaussian distribution. As discussed in the beginning of this Chapter, by the Central Limit Theorem a Poisson random variable, Y, with a large mean is approximately Gaussian in distribution. By the Delta method, the logarithm of Y is also approximately Gaussian. However, simple simulations with a mean in the range 20 to 100 show that the Gaussian approximation to Y is generally more accurate than that to to $\log(Y)$. In particular, $\log(Y)$ has a relatively heavy left-hand tail, destroying the approximate symmetry. Moreover, the logarithmic transformation does not provide us with a variance that is relatively independent of the mean. To achieve that we employ the transformation introduced by Anscombe (1948) that reduces the positive skewness of a Poisson random variable and also leaves the variance approximately independent of the mean. In the same article, Anscombe also introduces a transformation for binomial data which could serve as the basis for a proposal for a binomial GLSM.

3.2.1 The Anscombe transformation

Let $Y_i \sim \text{Poisson}(\mu_i = e^{\eta_i})$ and define $Y_i^A = \sqrt{Y_i + \frac{3}{8}}$. Anscombe (1948) showed that approximately,

$$\mathbb{E}\left[\sqrt{Y_i + \alpha}\right] \approx \sqrt{\mu_i + \alpha} - \frac{1}{8\mu_i^{1/2}} + \frac{24\alpha - 7}{128\mu_i^{3/2}}$$

$$\mathbb{V}\text{ar}\left[\sqrt{Y_i + \alpha}\right] \approx \frac{1}{4}\left\{1 + \frac{3 - 8\alpha}{8\mu_i} + \frac{32\alpha^2 - 52\alpha + 17}{32\mu_i^2}\right\}$$

Therefore, choosing $\alpha = \frac{3}{8}$ gives the Anscombe transformation that makes the first order term in the variance disappear resulting in a variance that is closer to a constant. Hence, we derive,

$$\text{E}[Y_i^A|\eta_i] = \text{E}\left[\sqrt{Y_i + \frac{3}{8}}\right] \approx \sqrt{\mu_i + \frac{3}{8}} - \frac{1}{8\mu_i^{1/2}} + \frac{1}{64\mu_i^{3/2}} \tag{3.2.1}$$

$$\text{Var}[Y_i^A|\eta_i] = \text{Var}\left[\sqrt{Y_i + \frac{3}{8}}\right] \approx \frac{1}{4}\left[1 + \frac{1}{16\mu_i^2}\right]. \tag{3.2.2}$$

However, we can further exploit this approximation. In particular, considering a second order Taylor expansion of $\sqrt{\mu_i + \frac{3}{8}}$ and for $\left|\frac{3}{8\mu}\right| < 1$ we can write,

$$\left(\mu + \frac{3}{8}\right)^{1/2} = \mu^{1/2}\left(1 + \frac{3}{8\mu}\right)^{1/2} \approx \sqrt{\mu} + \frac{3}{16\sqrt{\mu}} - \frac{9}{512\mu^{3/2}}.$$

Substituting now back to (3.2.1) we can derive an equivalent approximation to the mean of the transformed random variable Y_i^A which reads,

$$\text{E}[Y_i^A|\eta_i] = \text{E}\left[\sqrt{Y_i + \frac{3}{8}}\right] \approx \sqrt{\mu_i} + \frac{1}{16\sqrt{\mu_i}} - \frac{1}{512\mu_i^{3/2}}, \tag{3.2.3}$$

where $\mu_i = e^{\eta_i}$. Our suggestion is to use (3.2.2) and (3.2.3) results in order to define a good proposal for our MCMC scheme.

We work under the same setting as in Section 3.1.1 but we now define the transformed

data by $Y_i^A = \sqrt{Y_i + \frac{3}{8}}$. According to the result described above, we will use the following approximation,

$$Y_i^A|\eta_i \sim \text{normal}\left(e^{\eta_i/2} + \frac{1}{16}e^{-\eta_i/2}, \frac{1}{4}\right).$$

Our aim, is to have data that are approximately Gaussian and centred on the parameters which in turn have a Gaussian prior resulting in a Gaussian posterior. Therefore, let $\psi_i = e^{\eta_i/2}$ and set $\mu_i^* = \frac{e^{-\eta_i/2}}{16}$. We now introduce two different methods for obtaining a Gaussian approximation to $\boldsymbol{\psi}$: moment matching and linearisation via a Taylor approximation.

3.2.2 Using moment matching

Using the properties of log-normal distribution we can derive the prior mean, variance and covariance for $\boldsymbol{\psi}$ from the prior Gaussian distribution of $\boldsymbol{\eta}$. These are,

$$\mathrm{E}[\psi_i] = \exp\left\{\frac{1}{2}\boldsymbol{f}_i'\boldsymbol{\mu}_\beta + \frac{\sigma^2}{8}\left(\boldsymbol{f}_i'\boldsymbol{\Sigma}_\beta\boldsymbol{f}_i + \boldsymbol{R}_{ii}\right)\right\} = \mu_{\psi_i},$$

$$\mathrm{Var}[\psi_i] = \left(\exp\left\{\frac{\sigma^2}{4}\left(\boldsymbol{f}_i'\boldsymbol{\Sigma}_\beta\boldsymbol{f}_i + \boldsymbol{R}_{ii}\right)\right\} - 1\right)\exp\left\{\boldsymbol{f}_i'\boldsymbol{\mu}_\beta + \frac{\sigma^2}{4}\left(\boldsymbol{f}_i'\boldsymbol{\Sigma}_\beta\boldsymbol{f}_i + \boldsymbol{R}_{ii}\right)\right\} = \boldsymbol{V}_{ii},$$

$$\begin{aligned}\mathrm{Cov}[\psi_i, \psi_j] &= \left(\exp\left\{\frac{\sigma^2}{4}(\boldsymbol{f}_i'\boldsymbol{\Sigma}_\beta\boldsymbol{f}_j + \boldsymbol{R}_{ij})\right\} - 1\right) \\ &\times \exp\left\{\frac{1}{2}\left(\boldsymbol{f}_i' + \boldsymbol{f}_j'\right)\boldsymbol{\mu}_\beta + \frac{\sigma^2}{8}\left(\boldsymbol{f}_i'\boldsymbol{\Sigma}_\beta\boldsymbol{f}_i + \boldsymbol{R}_{ii} + \boldsymbol{f}_j'\boldsymbol{\Sigma}_\beta\boldsymbol{f}_j + \boldsymbol{R}_{jj}\right)\right\} = \boldsymbol{V}_{ij}.\end{aligned}$$

To summarise, $\boldsymbol{\psi}$ has, *a priori*, a multivariate log-normal distribution with mean $\boldsymbol{\mu}_\psi$ and covariance matrix $\boldsymbol{V}$ with elements μ_{ψ_i} and $\boldsymbol{V}_{ii}, \boldsymbol{V}_{ij}$ respectively. In order to work under the Gaussian framework we approximate this log-normal distribution by a normal

distribution with the same mean and variance. Therefore, we are now able to derive the approximate joint distribution of $\boldsymbol{\psi}, \boldsymbol{Y}^A$,

$$\begin{bmatrix} \boldsymbol{\psi} \\ \boldsymbol{Y}^A \end{bmatrix} \sim \text{MVN}\left(\begin{bmatrix} \boldsymbol{\mu}_{\boldsymbol{\psi}} \\ \boldsymbol{\mu}_{\boldsymbol{\psi}} + \boldsymbol{\mu}^* \end{bmatrix}, \begin{bmatrix} \boldsymbol{V} & \boldsymbol{V} \\ \boldsymbol{V} & (\boldsymbol{V} + \frac{1}{4}\boldsymbol{I}) \end{bmatrix} \right). \tag{3.2.4}$$

If (3.2.4) were true, then the conditional distribution of $\boldsymbol{\psi}|\boldsymbol{y}^A$ would be,

$$\boldsymbol{\psi}|\boldsymbol{y}^A \sim \text{MVN}\left(\boldsymbol{\mu}_{\psi|y^A}, \boldsymbol{V}_{\psi|y^A}\right), \tag{3.2.5}$$

with mean and variance given by,

$$\boldsymbol{\mu}_{\psi|y^A} = \boldsymbol{\mu}_{\boldsymbol{\psi}} + \boldsymbol{V}\left(\boldsymbol{V} + \frac{1}{4}\boldsymbol{I}\right)^{-1}(\boldsymbol{y}^A - \boldsymbol{\mu}_{\boldsymbol{\psi}} - \boldsymbol{\mu}^*) \tag{3.2.6}$$

$$\boldsymbol{V}_{\psi|y^A} = \boldsymbol{V} - \boldsymbol{V}\left(\boldsymbol{V} + \frac{1}{4}\boldsymbol{I}\right)^{-1}\boldsymbol{V}, \tag{3.2.7}$$

which is now our new proposal in the MCMC scheme to sample from the posterior distribution $\pi(\boldsymbol{\eta}|\boldsymbol{y})$. Once more, we make use of the transformed data for the correction term in the mean and replace $\boldsymbol{\mu}^*$ with $1/(16\boldsymbol{y}^A)$ leading to a MHIS. However, the components of $\boldsymbol{\psi}$ can only take positive values, whereas, the approximation of the log-normal by a normal distribution can give rise to negative values of $\boldsymbol{\psi}$. Therefore we impose the constraint that each component of a proposed value must be positive. In practice, we sample from this truncated distribution by rejection sampling.

Algorithm (A1)

Let $\boldsymbol{\psi}^{prop}$, $\boldsymbol{\psi}^{cur}$ be the proposed and current value of $\boldsymbol{\psi}$. Moreover, denote the proposal distribution derived in (3.2.5)-(3.2.7) by $q(\boldsymbol{\psi}|\boldsymbol{y}^A)$ and the target by $\pi(\boldsymbol{\psi}|\boldsymbol{y})$, where

$$\pi(\boldsymbol{\psi}|\boldsymbol{y}) = \Big|\frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\psi}}\Big|\pi(\boldsymbol{\eta}|\boldsymbol{y})\big|_{\boldsymbol{\eta}=2\mathbf{log}(\boldsymbol{\psi})}.$$

The MCMC scheme used in order to draw samples from $\pi(\boldsymbol{\eta}|\boldsymbol{y})$ is as follows,

- Set initial values $\boldsymbol{\psi}^{cur} = \boldsymbol{\psi}^0$.
- Propose, $\boldsymbol{\psi}^{prop}$ according to (3.2.5)-(3.2.7)
- If $\psi_i > 0$, for $i = 1, ..., d$,
  - Accept with probability,

$$\alpha(\boldsymbol{\psi}^{cur}, \boldsymbol{\psi}^{prop}) = \min\left(1, \frac{\pi(\boldsymbol{\psi}^{prop}|\boldsymbol{y})q(\boldsymbol{\psi}^{cur}|\boldsymbol{y}^A)}{\pi(\boldsymbol{\psi}^{cur}|\boldsymbol{y})q(\boldsymbol{\psi}^{prop}|\boldsymbol{y}^A)}\right)$$

  - If $\boldsymbol{\psi}^{prop}$ is accepted, set $\boldsymbol{\psi}^{cur} = \boldsymbol{\psi}^{prop}$
  - Obtain $\boldsymbol{\eta}^{cur} = 2\log(\boldsymbol{\psi}^{cur})$.

3.2.3 Linearisation of the transformed variable $\boldsymbol{\psi}$

As described in the previous section, using Anscombe's transformation the latent process $\boldsymbol{\eta}$ has to be transformed to $\boldsymbol{\psi}$. An important drawback of this approach is that although $\boldsymbol{\psi}$ has a log-normal distribution we consider it having a normal distribution, albeit with the same expectation and variance. In this section, in order to avoid this misspecification, we approximate $\boldsymbol{\psi}$ through a linear relationship with $\boldsymbol{\eta}$ while keeping Anscombe's transformation for the data. To do that, we Taylor expand $\boldsymbol{\psi}(\boldsymbol{\eta})$ about $\boldsymbol{\eta} = \boldsymbol{m}$, for

some 'central' value $\boldsymbol{m}$, such as an approximation to the posterior expectation of $\boldsymbol{\eta}$, and construct a proposal for $\boldsymbol{\eta}$.

In the following, for ease of notation we denote the prior mean of the process $\boldsymbol{\eta}$ by $\boldsymbol{\mu}_\eta$ and its prior covariance matrix by $\boldsymbol{V}_\eta$. Recall that $\boldsymbol{\psi} = e^{\frac{\boldsymbol{\eta}}{2}}$ and $\boldsymbol{y}^A = \sqrt{\boldsymbol{y} + \frac{3}{8}}$ and define $\boldsymbol{D}_m$ to be a diagonal matrix with diagonal elements equal to $e^{\frac{\boldsymbol{m}}{2}}$. Finally, let $\boldsymbol{c}_\eta$ be a column vector with elements the diagonal of $\boldsymbol{V}_\eta$. Using results of Section 3.2.1 we have that approximately,

$$\boldsymbol{Y}^A|\boldsymbol{\psi} \sim \text{MVN}(\boldsymbol{\psi}, 0.25\boldsymbol{I}). \tag{3.2.8}$$

A first order Taylor expansion for $\boldsymbol{\psi}$ gives,

$$\begin{aligned} \boldsymbol{\psi} &= e^{\boldsymbol{\eta}/2} \\ &= e^{\frac{1}{2}\boldsymbol{m}+\frac{1}{2}(\boldsymbol{\eta}-\boldsymbol{m})} \\ &\approx e^{\frac{1}{2}\boldsymbol{m}}\left(\mathbf{1} + \frac{1}{2}(\boldsymbol{\eta}-\boldsymbol{m})\right), \end{aligned} \tag{3.2.9}$$

The expected value and variance of $\boldsymbol{\psi}$ can be approximated by,

$$\begin{aligned} \boldsymbol{\mu}_\psi &= \mathbb{E}[\boldsymbol{\psi}] \approx \boldsymbol{D}_m\left(\mathbf{1} + \frac{1}{2}(\boldsymbol{\mu}_\eta - \boldsymbol{m})\right), \\ \boldsymbol{V}_{\boldsymbol{\psi}} &= \mathbb{V}\text{ar}[\boldsymbol{\psi}] \approx \frac{1}{4}\boldsymbol{D}_m\boldsymbol{V}_\eta\boldsymbol{D}_m, \end{aligned} \tag{3.2.10}$$

Consequently, we can approximate the distribution of $\boldsymbol{\psi}$ through a multivariate normal distribution with mean and covariance matrix given by (3.2.10) so that $\boldsymbol{\psi} \sim \text{MVN}(\boldsymbol{\mu}_\psi, \boldsymbol{V}_\psi)$.

Combining this with the normal likelihood of $\boldsymbol{y}^A|\boldsymbol{\psi}$ as shown in, (3.2.8), we obtain that,

$$\boldsymbol{\psi}|\boldsymbol{y}^A \sim \text{MVN}(\boldsymbol{\mu}_{\psi|y^A}, \boldsymbol{V}_{\psi|y^A})$$

where,

$$\begin{aligned} \boldsymbol{\mu}_{\psi|y^A} &= \boldsymbol{\mu}_\psi + \boldsymbol{V}_\psi \left(\boldsymbol{V}_\psi + \frac{1}{4}\boldsymbol{I}\right)^{-1} \left(\boldsymbol{y}^A - \boldsymbol{\mu}_\psi\right) \\ \boldsymbol{V}_{\psi|y^A} &= \boldsymbol{V}_\psi - \boldsymbol{V}_\psi \left(\boldsymbol{V}_\psi + \frac{1}{4}\boldsymbol{I}\right)^{-1} \boldsymbol{V}_\psi. \end{aligned} \tag{3.2.11}$$

In order to obtain our proposal for $\boldsymbol{\eta}$ we rearrange expression (3.2.9) such that,

$$\boldsymbol{\eta} = 2\left(\boldsymbol{D}_m^{-1}\boldsymbol{\psi} - \mathbf{1} + \frac{1}{2}\boldsymbol{m}\right).$$

Using now the conditional mean of $\boldsymbol{\psi}$ as given in (3.2.11), our approximation is that

$$\boldsymbol{\eta}|\boldsymbol{y}^A \sim \text{MVN}(\boldsymbol{\mu}_{\eta|y^A}, \boldsymbol{V}_{\eta|y^A}),$$

where the mean and variance read,

$$\boldsymbol{\mu}_{\eta|y^A} = 2\left(\boldsymbol{D}_m^{-1}\boldsymbol{\mu}_{\psi|y^A} - \mathbf{1} + \frac{1}{2}\boldsymbol{m}\right),$$

$$\boldsymbol{V}_{\eta|y^A} = 4\boldsymbol{D}_m^{-1}\boldsymbol{V}_{\psi|y^A}\boldsymbol{D}_m^{-1} \tag{3.2.12}$$

respectively. As illustrated in previous sections both $\mathbb{E}\left[\boldsymbol{Y}^l|\boldsymbol{\eta}\right]$ and $\mathbb{E}\left[\boldsymbol{Y}^p|\boldsymbol{\eta}\right]$ are approximately equal to $\boldsymbol{\eta}$. Therefore, the above algorithm could be implemented either using, $\boldsymbol{m} = \log(\boldsymbol{y})$ or $\boldsymbol{m} = \log(\boldsymbol{y} + \frac{1}{2})$. We choose to use set $\boldsymbol{m} = \log(\boldsymbol{y})$ and denote this

algorithm by RA. Additionally, we try an iterative scheme in order to obtain a value for $\boldsymbol{\mu}_{\eta|y^A}$ closer to the the true posterior expectation $\boldsymbol{\mu}_{\eta|y}$. More explicitly, we give the initial value $\boldsymbol{m} = \log(\boldsymbol{y})$ and calculate $\boldsymbol{\mu}^{(1)}_{\eta|y^A}$ using (3.2.11) and and the conditional mean of (3.2.12); we then set $\boldsymbol{m} = \boldsymbol{\mu}^{(1)}_{\eta|y^A}$ and repeat the calculations to obtain $\boldsymbol{\mu}^{(2)}_{\eta|y^A}, \boldsymbol{\mu}^{(3)}_{\eta|y^A}$. The intuition behind this iterative scheme is that after several iterations $\boldsymbol{m}$ should more closely approximate $\mathbb{E}[\boldsymbol{\eta}|\boldsymbol{y}]$. This iterative version of RA with 3 iterations will be referred to as iRA.

If η_i is small, i.e., $m_i \approx 0$, then the Anscombe approximation will be poor but the likelihood in expression (3.2.8) is relatively uninformative compared to the prior variance in expression (3.2.10). If the likelihood now was completely uninformative then $\boldsymbol{\mu}_\psi = \boldsymbol{\mu}_{\psi|y^A}$ and $\boldsymbol{V}_\psi = \boldsymbol{V}_{\psi|y^A}$ and the back transformation given in (3.2.12) would be exact reducing to

$$\boldsymbol{\eta}|\boldsymbol{y}^A \sim \text{MVN}\left(\boldsymbol{\mu}_\eta, \boldsymbol{V}_\eta\right).$$

Our algorithm would therefore be exact in the case of a completely uninformative likelihood.

On the other hand, if η_i is large then the Anscombe transformation will be accurate and the likelihood in expression (3.2.8) would be very informative compared to the prior. Therefore, with an appropriate choice choice of $\boldsymbol{m}$ the Taylor expansion should be accurate resulting in an accurate posterior for $\boldsymbol{\eta}$.

3.3 Simulation study and results

3.3.1 Simulation study

In this section we assess the performance of our proposals. We compare our algorithms against the algorithm of Christensen et al. (2006) and the pMMALA, the further simplified version of sMMALA as described in Section 2.2.7, suggested by Girolami & Calderhead (2011). The comparison is made on a simple scenario of a Poisson GLSM where all parameters describing the covariance structure and the mean are fixed to their true values. Therefore inference concerns only the latent process $\boldsymbol{\eta}$. As we have discussed in Section 3.1.1, given a sample $\boldsymbol{\eta}$, it is straightforward to produce a sample from the posterior of $(\boldsymbol{\beta}, \boldsymbol{S})$. The constructed normal proposals of the previous sections have been replaced by the Student's $t-$distribution with 10 degrees of freedom; see (3.1.11).

Design of simulation study

We assess the performance of each algorithm under different scenarios of parameter values in three different dimensions. In particular, we explore the performance of the algorithms when the dimension of the process $\boldsymbol{\eta}$, is equal to $d = 25, 49, 100$. The observations are sampled on a regular grid in the square $\left\{1, 2, ..., \sqrt{d}\right\}^2$. For the mean of the process, we consider $\boldsymbol{\mu}_\eta \in \{\log(1), \log(10), \log(100)\}$ and for the variance $\sigma^2 \in \left\{\frac{1}{3}, 1, 3\right\}$. As far as the correlation structure is concerned, we use the exponential correlation function with $\phi \in \{1, 10, 100\}$. We set the default parameters to $\left(\boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10\right)$ and test the effect of the parameter values by making a single change from this combination each time, resulting in seven different scenarios of parameter values within a given dimension. Finally, three different datasets, namely $\mathrm{d}_1, \mathrm{d}_2, \mathrm{d}_3$, were simulated for each of the seven scenarios for all three different dimensions.

Data simulation and standardisation

We treat all parameters as known and fixed to their true prior values. However, in a full MCMC scheme we would also sample these parameters from their posterior distribution. This could give rise to parameter values which were typically quite different to the truth and more representative of their posterior distribution given the particular dataset. Given the large correlation between the latent variables in some of our simulations ($\phi = 10$ and $\phi = 100$) we expect, in particular, considerable variability in the sample mean, $\bar{\boldsymbol{\eta}}$, compared with the variance, σ^2/d, that would arise if the latent variables were iid. At the same time, given their high correlation, the variability of the individual values about $\bar{\boldsymbol{\eta}}$ will be small compared with σ^2. We therefore expect that often the posterior for $\boldsymbol{\mu}_\eta$ will be centered at a value that is both quite different to the true mean $\boldsymbol{\mu}_\eta$, and more appropriate for use in generating samples from the latent variables using an independence sampler. Since we are not performing inference on the parameters, we simply ensure that the true mean will be sensible for use in our independence sampler by sampling the latent variables subject to the constraint that $\bar{\boldsymbol{\eta}} = \boldsymbol{\mu}_\eta$. In practice, we achieve this by sampling the latent variables $\boldsymbol{\eta}^{true}$ by its true distribution and then setting

$$\boldsymbol{\eta} = \boldsymbol{\eta}^{true} - \bar{\boldsymbol{\eta}}^{true} + \boldsymbol{\mu}_\eta.$$

Choice of proposal distribution

In order to define a proposal distribution for our MCMC schemes we considered cases where the chain starts at the tails of target and assessed its convergence and mixing behaviour. Therefore, using the posterior samples drawn from the algorithm of Christensen et al. (2006), a sample vector of $\boldsymbol{\eta}$, lying in the tails of the target was identified. This was subsequently used as initial values in our algorithm that was run for 10^6 iterations

using different proposal distributions such as t_4, t_{10}, t_20 and normal.

In Figure 3.1 we provide some indicative boxplots of the effective sample sizes achieved from algorithm L1 at a single run for different scenarios of parameter values using a t_4, t_{10} and t_{20} proposal distribution for dimension $d = 25$. As we see the t_4 proposal always achieves the lowest ESS. In most cases t_{20} appears to perform better than the t_{10} proposal in terms of ESS but we have found cases such as the one in the top left plot where the t_{10} proposal outperforms the t_{20}. In practice we have seen cases were a t_{20} proposal presented extended periods of rejections either in the beginning of the chain or even after 4×10^6 iterations.

A normal proposal would be highly inefficient due to its very light tails. The resulting sampler would not be geometrically ergodic and therefore a central limit theorem would not hold (Roberts and Rosenthal, 2008). In our experiments we encountered cases where very few, if any, proposed moves were accepted. Therefore such a choice was rejected. On the other hand with a t_4 proposal a central limit theorem would hold for every function h with finite second moment. However, such a proposal would again lead to an inefficient sampler due to its wrong shape. This was evident in our experiments through the presence of very low acceptance rates and hence very low ESS that were achieved. Since as $\lim_{df\to\infty} T_{df}(x) = \Phi(x)$, the optimal value for df will lie between these two extremes. However, the optimal value for df will be different for each scenario of parameter values, dimension and dataset. We are in favour of avoiding extended periods of rejections and slow mixing rather than attempting to achieve an extremely accurate shape for our proposal. For that reason we chose to use a t_{10} proposal.

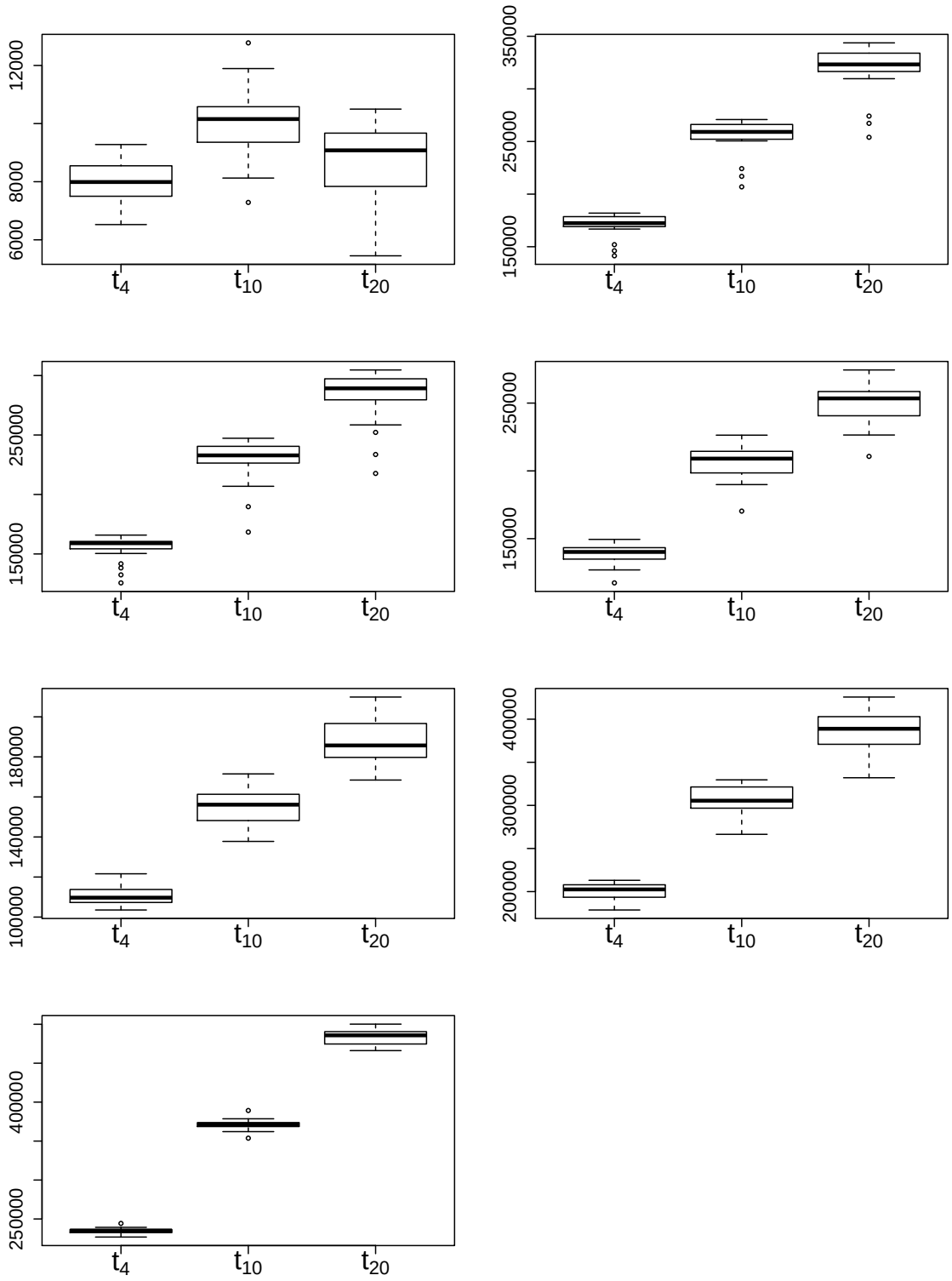

Figure 3.1: Boxplots of ESS obtained form algorithm L1 for seven different scenarios of parameter values and dimension $d = 25$.

MCMC implementation

Initial runs of the algorithm of Christensen et al. (2006) were implemented for all scenarios for each dataset. A sample point from the posterior with marginal components always lying within the 0.01 and 0.99 quantiles of their distributions was chosen, by rejection sampling, and was used as the starting value for each of the algorithms in our simulation study for that particular dataset. The algorithms were run for 10^6 iterations and the first 2×10^5 iterations were discarded as burn in. All results are based on 8×10^5 samples drawn from the posterior distribution of $\boldsymbol{\eta}|\boldsymbol{y}$.

We monitor acceptance rates (α), effective sample sizes (ESS) and the CPU time that was needed for 8×10^5 samples to be drawn. Finally, we divide the ESS by the CPU time in order to get a measure of efficiency for each algorithm, the adjusted ESS.

Assessment of convergence

In this thesis we measure the efficiency of a MCMC sampler in terms of the adjusted ESS. However, before proceeding to the comparison of ESS of two different algorithms it is crucial to ensure that both samplers have adequately explored the target distribution. This is because a MCMC algorithm can successfully explore the main body of the target distribution and achieve high ESS while failing to explore the tails.

In Section 2.1.2 we mentioned some widely used convergence diagnostics which deal with assessing whether the distribution of either, parts of the same chain or two different chains are similar or not. However, such a conclusion is reached by comparing either the first two moments or a certain set of quantiles of the empirical distribution of the chains. As mentioned in Brooks & Gelman (1998*b*) such convergence diagnostics are not appropriate when inference relies on distributional summaries other than the first

two moments. Boone et al. (2014) for instance, illustrate a simple example where two distributions can have the same mean, variance, 2.5%, 97.5% quantlis but have markedly differet densities. In such cases, the diagnostics mentioned above would fail to detect the dissimilarity of the two distributions.

For that reason, we feel we have to protect the assessment of convergence from such cases. Sharing similar motivation with Brooks et al. (2003) and Boone et al. (2014), we employ the two-sample Kolmogorov-Smirnov statistics and construct a non-parametric diagnostic that assesses the shape of the whole distribution.

The two sample K–S test (Kolmogorov 1933) is a non-parametric test used to assess whether the equality of two distribution functions. In practice one has two samples and wishes to assess whether they arise from the same distribution. This is achieved by comparing their empirical distribution functions (edf) using as test statistic the maximum difference between the two edfs. More explicitly, we have two independent random samples $\boldsymbol{X} = (X_1, X_2, ..., X_{n_1})$ and $\boldsymbol{Y} = (Y_1, Y_2, ..., Y_{n_2})$ with cumulative distribution functions F_1, F_2 respectively and we are interested in testing, $H_0 : F_1 = F_2$ vs. $H_1 : F_1 \neq F_2$. Let the empirical distribution functions (edf) of $\boldsymbol{X}$, $\boldsymbol{Y}$ to be,

$$F_{n_1}(x) = \frac{1}{n_1} \sum_{j=1}^{n_1} I\left(X_j \leq x\right) \quad \text{and} \quad F_{n_2}(x) = \frac{1}{n_2} \sum_{j=1}^{n_2} I\left(Y_j \leq x\right),$$

respectively, then the KS statistic is defined as,

$$T := \sup_x \left| F_{n_1}(x) - F_{n_2}(x) \right|.$$

The null hypothesis H_0 is rejected at significance level α if the statistic T is greater than the critical value $c(\alpha)$. Tables with the critical values of the distribution of T are

available such as in Conover (1999).

Henceforth we refer to the algorithm of Christensen et al. (2006) as CRS. Consider any algorithm and let $\boldsymbol{s}_i$, $(i = 1, \ldots, d)$ be the posterior sample for the $i-$th component of the latent variable, S_i, after burn-in has been discarded. We first apply a uniform (across components) thinning, to the sample as follows. For each component, i, we repeatedly thin the sample by a factor of 2 until the estimated lag-1 autocorrelations of S_i and of S_i^2 using the thinned sample are both not significantly different from zero, at the 5% level and under the null hypothesis of Gaussian noise with no correlation; let t_i be the number of thinnings required and let $t_{max} = \max_{\{i=1,\ldots,n\}} t_i$. Each component, i, is then thinned by a factor of two a further $(t_{max} - t_i)$ times. By thinning all components equally, we preserve the correlation structure between the components of $\boldsymbol{S}$ and at the same time obtain close to zero correlation within each marginal sample.

For each data set and for each pair of algorithms, CRS and some other algorithm, G, K-S tests were conducted on each of the d marginal components of the pair of thinned samples, leading to d KS statistics, $KS_1, \ldots, KS_d$. Since the components of $\boldsymbol{S}$ are positively correlated the marginal KS statistics might be related and so they cannot be treated as being independent tests of convergence. In order to account for this correlation and incorporate it into our null hypothesis we conduct a permutation test (see Davison & Hinkley (1997)). A single test statistic $K = \sum_{i=1}^{d} KS_i$ is created from the marginal KS statistics, and a further $M = 1000$ pseudo test statistics $K^m = \sum_{i=1}^{d} KS_i^m$, $m = 1, \ldots, M$ are created from $1000 \times d$ marginal pseudo KS statistics. The marginal test statistics KS_i^m, $(i = 1, \ldots, d)$ are generated together as follows. Suppose that the thinned sample sizes are n_{CRS} and n_G so that laying one beneath the other leads to an $(n_{CRS} + n_G) \times d$ matrix. A random permutation on the numbers $1, \ldots, (n_{CRS} + n_G)$ was generated

and applied simultaneously to all d columns of the matrix. KS_i^m is then generated by conducting a K-S test on the first n_{CRS} elements of column i against the last n_G elements. The sample $K^1, \ldots, K^M$ is then a sample from the distribution of K under the null hypothesis of,

$$H_0 \quad : \quad \text{The thinned samples from } G \text{ and from } CRS \text{ both represent independent identically distributed draws from the same joint distribution for } \boldsymbol{S} \qquad (3.3.1)$$

3.3.2 Results

Tables 3.3.1-3.3.5 display summaries of the performance of L1, L2, A, RA and iRA respectively for all three dimensions and scenarios of parameter values whereas Tables 3.3.6 and 3.3.7 illustrate the same summaries for the algorithm of Christensen et al. (2006) and pMMALA respectively. Both algorithms were tuned to have acceptance rates between 58% and 60%, which is close to the approximately optimal 57% for MALA algorithms. As already mentioned we record acceptance rates, relative ESS, i.e., ESS/8×10^5, CPU timings and adjusted ESS for each algorithm. The relative ESS and adjusted ESS are summarised in terms of their minimum, median and maximum values. The three different columns within each scenario of parameter values correspond to the different simulated datasets.

We illustrate with grey colour the cases for which, according the permutation tests, we failed to accept the null hypothesis (3.3.1). Additionally, we use $(*)$ to denote the cases where the thinning process resulted in sample sizes that were less than 50 and therefore the permutation test was not conducted for that chain since it would have little power to detect any discrepancies.

The most immediate pattern in Tables 3.3.1-3.3.5 is that both acceptance rates and ESS decrease as dimension increases for all of our algorithms whereas, according to Tables 3.3.6 and 3.3.7, the performance of Christensen et al. (2006) and pMMALA appears to be more stable across dimensions. For our algorithms this is due to a single global approximation to the posterior. Even when $\phi = 1$, which corresponds to the lowest correlation between the components of $\boldsymbol{\eta}$, our algorithms perform poorly. To give an intuition behind this, suppose that the posterior distribution of $\boldsymbol{\eta}$ is $\pi(\boldsymbol{\eta}) \approx \prod_{i=1}^{d} \pi_i(\eta_i)$ and our proposal is $q(\boldsymbol{\eta}) = \prod_{i=1}^{d} q_i(\eta_i)$. For simplicity, let us assume that the approximation to each component is of similar accuracy in the sense that $\inf_\eta \frac{\pi_i(\eta)}{q_i(\eta)} = \delta$, $\forall i$. Consequently, $\inf_{\boldsymbol{\eta}} \frac{\pi(\boldsymbol{\eta})}{q(\boldsymbol{\eta})} = \delta^d$. For such an MHIS, it is well known (e.g. Liu (1996), Murray (2004)) that an upper bound on the total variation distance between the target and the distribution of the state after n iterations is proportional to $(1 - \delta^d)^n$. Therefore, the rate of convergence decays exponentially with dimension. Considering the poor performance when the components of the process are approximately independent in each case, either the Gaussian approximation of the data or the Gaussian approximation of the prior for $\boldsymbol{\psi}$ is not sufficiently accurate.

Algorithm L1

We now look at each algorithm separately starting from L1 and Table 3.3.1. Keeping σ^2 and ϕ fixed, the performance of the algorithm improves as the prior mean $\boldsymbol{\mu}_\eta$ increases. In this case, the normal approximation to the distribution of $\boldsymbol{Y}^l$ is more accurate since each successive term in (3.1.1) is negligible compared with the previous term and in combination with the normal prior the overall normal approximation is better. On the other hand, with the prior mean fixed, the performance of the algorithm deteriorates as σ^2 increases. This could be due to the fact that, while assessing the effect of σ^2 on

the performance of the algorithm we keep the mean of the likelihood fixed and close to 10 which is not high enough for our Gaussian approximation for Y^l to be good. At the same time, as σ^2 increases the normal prior becomes less informative and the likelihood contributes relatively more to our proposal. Since the likelihood approximation is not especially accurate this results into lower acceptance rates and ESS. However, we should always keep in mind that in the standardisation of $\boldsymbol{\eta}$ we have not accounted for the variance. Therefore, we cannot draw clear conclusions about its true effect on the performance of the algorithms. Finally, as the prior correlation increases, i.e. as ϕ increases, the shape of the posterior is closer to normal (see Figure 3.4) and therefore our proposal matches its shape reasonably well leading to better results. This is because, increasing correlation leads to $(d-1)$ small principal components and one considerably large. In this case, all small $(d-1)$ principal components will have a very small variance, and there will be a lot of variability only along one component. Therefore, the prior becomes a lot more informative on these small principal components while the likelihood, and any approximation made to it, does not play an important role. These result in a MHIS where the inaccuracy in our approximation to the likelihood only has a real impact on one of the d principal components. We therefore expect to avoid the curse of dimensionality.

Algorithm L2

In Section 3.1.4 we showed that the transformation $\log(Y+0.5)$ should be more accurate and therefore provide a more efficient proposal for algorithm L2. In Table 3.3.2 we see that although in some cases L2 might provide better acceptance rates than L1; it always performs worse than L1 in terms of minimum ESS. This small increase in acceptance rates does give an indication of the improved accuracy, but acceptance rates cannot be

used to compare efficiency. When the transformation $\log(\boldsymbol{y} + 0.5)$ is used, the diagonal elements of $\boldsymbol{\Sigma}^*$ become larger resulting in a smaller proposal variance proposal. This, or the altered mean or even the combination of both appears to actually reduce the accuracy of the proposal.

Algorithm A1

Table 3.3.3 illustrates the results for algorithm A1. For this algorithm there are two things to consider, the normal approximation to the likelihood and the normal approximation of the log-normal prior of the transformed parameter $\boldsymbol{\psi}$. For increasing $\boldsymbol{\mu}_\eta$, Anscombe's approximation for the Poisson distribution becomes more accurate. Although the informativeness of the data increases as the true mean increases, that of the prior remains unaffected. Hence, more weight is given to the improving approximation of the likelihood. Increasing σ^2 increases the skewness of the log-normal distribution so that it cannot be effectively approximated by the normal distribution. Consequently, our normal approximation to the target fails with increasing prior variance and this is demonstrated in the results of Table 3.3.3. In terms of ESS, this can only be seen for dimension $d = 25$ since in higher dimensions we cannot make any statement regarding convergence for the corresponding scenarios of increasing σ^2. However, the fact that a sufficiently large iid sample could not be obtained through our thinning process, indicates that the proposal is poor. Due to variability between the different datasets no obvious pattern can be seen for the correlation parameter ϕ.

Algorithms RA and iRA

Table 3.3.4 and Table 3.3.5 correspond to algorithms RA and iRA respectively. Both algorithms are expected to present the same pattern of performance since both use the

same proposal distribution. The only difference lies in the iterative scheme we employ in order to obtain the mean of the proposal for iRA, and through which we expect it to perform better. Indeed, we find that iRA always performs at least as well as RA and it often performs better by a factor of two or three. As in algorithms L1 and A1, increasing $\boldsymbol{\mu}_\eta$ leads to better performance of the algorithms. This is a by-product of the likelihood and prior approximations. For the likelihood approximation we have used Anscombe's transformation which does perform better with increasing mean of the Poisson. Let us now consider the effect of increasing $\boldsymbol{\mu}_\eta$ to the linear approximation of $\boldsymbol{\psi}$ by $\boldsymbol{\eta}$. The Taylor expansion of the exponential function, e^x, holds for small x so that each successive term $x^2, x^3, x^4, ...$ is small compared to the previous one. In our case $x = (\eta - m)$ where $m = \log(y)$. As $\boldsymbol{\mu}_\eta$ increases we have already justified that $\mathbb{E}\left[\log\left(Y\right)|\eta\right] \approx \eta$ and therefore the approximate linear relation between $\boldsymbol{\psi}$ and $\boldsymbol{\eta}$ becomes more accurate leading to a more efficient normal approximation (see Equation 3.2.9). Furthermore, as the prior variance σ^2 increases, while $\boldsymbol{\eta}$ and ϕ remain fixed, the accuracy of the approximation decreases since $\boldsymbol{\eta}$ can be very different from its expectation. With increasing prior variance approximating $\boldsymbol{\eta}$ via $\log(\boldsymbol{y})$ fails as mentioned earlier. Once more, no immediate effect of ϕ is apparent in the data.

Algorithm of Christensen et al. (2006) and pMMALA

Table 3.3.6 and Table 3.3.7 illustrate the results of the algorithm of Christensen et al. (2006) and pMMALA respectively. As we see, the latter always provides better minimum and median ESS whereas the former tends to achieve higher maximum ESS. The algorithm of Christensen et al. (2006) provides more stable results due to the standardisation of the latent process with respect to the posterior mode and variance.

Table 3.3.7 illustrates that the range of ESS, as defined by the difference between max-

imum and minimum, is quite wide compared to that provided by the remaining algorithms. In order to construct the preconditioning matrix used in the proposal of pMMALA, Girolami & Calderhead (2011) choose to take the expectation over both the latent process and observations rather than using the current values of the latent process or the data. Consider, for instance, an ideal situation where the shape of the proposal matches exactly the shape of the posterior. In this case, the proposed jumps will tend to be larger in directions where the target is wider and smaller in directions where the target is narrower. However, if the shape of the proposal and target do not match then to obtain the same overall acceptance rate, the size of proposed jumps will be limited by the narrow target directions so that the movement in the wide target directions will be relatively slow. Therefore, we could expect such a pattern in the ESS of pMMALA, since the chain will mix well for components with variance similar to the expected one whereas it will mix poorly for those having a considerably different variance. This is also supported by the fact that this pattern is more obvious for scenarios with large prior variability between the components of $\boldsymbol{\eta}$ (i.e., large σ^2 or low ϕ). On the other hand,, Christensen et al. (2006) use the observed information matrix and thay make use of the data by using the maximum likelihood estimate of η_i. Therefore the proposal used by the algorithm of Christensen et al. (2006) would be more appropriate for a particualr dataset.

The best performing of our proposed algorithms are L1 and iRA since they provide better acceptance rates, ESS and adjusted ESS. It also appears that L1 performs better than iRA when the $\boldsymbol{\mu}_\eta = \log(100)$ irrespective of dimension. The algorithm of Christensen et al. (2006) appears to have a more robust performance than our algorithms, both within and between dimensions. However, irrespective of the dimension, the algorithm

L1 always achieves better ESS and adjusted ESS than the algorithm of Christensen et al. (2006) when $\boldsymbol{\mu}_\eta = \log(100)$. Whereas, the algorithm iRA always performs better than the algorithm of Christensen et al. (2006) when $\phi = 100$. As far as the adjusted ESS is concerned, under this simple scenario where the parameters of the model are assumed to be fixed, our algorithm has an advantage over pMMALA and the algorithm of Christensen et al. (2006). Since the latter are both MALA algorithms have in general higher computational cost. Therefore, in cases where the performance of our algorithms, in terms of ESS, is comparable with that of Christensen et al. (2006) and pMMALA, our algorithms tend to provide higher adjusted ESS, especially when $d = 25$. However, in the case of a full MCMC where σ^2 and $\boldsymbol{\phi}$ are also updated at each iteration, the computational cost of L1 would also increase since it would require additional matrix multiplications. Given this, it might well be sensible to have a large number of updates to $\boldsymbol{S}$ before updating ϕ and σ^2.

Table 3.3.1: Algorithm L1. Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. Grey color indicates that the permutation test does not support convergence. The $*$ indicates that the thinned sample size was less than 50 and permutation test was not conducted.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 87.8$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
α	(.091, .065, .089)	(.497, .365, .420)	(.468, .386, .400)	(.443, .286, .371)	(.387, .136, .286)	(.541, .293, .376)	(.578, .572, .570)
Min. ESS	(.006, .006, .007)	(.208, .086, .149)	(.174, .131, .131)	(.174, .070, .119)	(.139, .032, .085)	(.274, .058, .106)	(.354, .361, .355)
Med. ESS	(.009, .007, .011)	(.261, .131, .190)	(.234, .163, .180)	(.210, .104, .155)	(.157, .040, .105)	(.308, .078, .138)	(.371, .372, .369)
Max. ESS	(.012, .008, .014)	(.273, .147, .201)	(.247, .181, .194)	(.226, .114, .172)	(.171, .046, .118)	(.328, .096, .153)	(.389, .384, .385)
Adj. Min. ESS	(54, 54, 63)	(1893, 782, 1356)	(1583, 1192, 1192)	(1583, 637, 1083)	(1265, 291, 773)	(2494, 527, 964)	(3222, 3286, 3231)
Adj. Med. ESS	(81, 63, 100)	(2375, 1192, 1729)	(2130, 1483, 1638)	(1911, 946, 1410)	(1429, 364, 955)	(2803, 710, 1256)	(3377, 3386, 3359)
Adj. Max. ESS	(109, 72, 127)	(2485, 1338, 1829)	(2248, 1647, 1766)	(2057, 1037, 1565)	(1556, 418, 1074)	(2985, 873, 1392)	(3541, 3495, 3504)
$d = 49$ $\overline{CPU} = 192.6$							
α	(.022*, .008*, .024)	(.165, .128, .184)	(.161, .091, .127)	(.167, .050, .104)	(.160, .070, .165)	(.127, .024, .177)	(.425, .378, .422)
Min. ESS	(.002*, .000*, .002)	(.027, .013, .029)	(.029, .014, .019)	(.028, .008, .016)	(.031, .014, .033)	(.014, .002, .027)	(.219, .159, .215)
Med. ESS	(.003*, .001*, .003)	(.033, .022, .042)	(.036, .018, .027)	(.039, .010, .022)	(.037, .014, .040)	(.020, .003, .036)	(.233, .190, .230)
Max. ESS	(.004*, .002*, .004)	(.037, .026, .046)	(.041, .020, .029)	(.044, .011, .025)	(.044, .019, .047)	(.024, .004, .041)	(.245, .204, .243)
Adj. Min. ESS	(8*, 2*, 8)	(112, 53, 120)	(120, 58, 78)	(116, 33, 66)	(128, 58, 137)	(58, 8, 112)	(909, 660, 892)
Adj. Med. ESS	(12*, 4*, 12)	(137, 91, 174)	(149, 74, 112)	(161, 41, 91)	(153, 58, 166)	(83, 12, 149)	(967, 789, 955)
Adj. Max. ESS	(16*, 8*, 16)	(153, 107, 191)	(170,83, 120)	(182, 45, 103)	(182, 78, 195)	(99, 16, 170)	(1017, 847, 1009)
$d = 100$ $\overline{CPU} = 278.3$							
α	(.001*, .002*, .001*)	(.133, .051, .038)	(.049, .039, .031)	(.043, .024, .006*)	(.034, .007*, .009)	(.179, .150, .063)	(.288, .256, .280)
Min. ESS	(.000*, .000*, .000*)	(.024, .007, .005)	(.007, .005, .004)	(.006, .003, .001*)	(.005, .001*, .002)	(.039, .023, .008)	(.122, .096, .116)
Med. ESS	(.000*, .000*, .000*)	(.028, .009, .006)	(.009, .006, .006)	(.008, .004, .002*)	(.006, .002*, .002)	(.044, .032, .010)	(.132, .107, .126)
Max. ESS	(.001*, .001*, .001*)	(.033, .010, .007)	(.011, .008, .006)	(.009, .005, .002*)	(.007, .002*, .003)	(.050, .036, .011)	(.143, .114, .136)
Adj. Min. ESS	(0.00*, 0.00*, 0.00*)	(68, 20, 14)	(20, 14, 11)	(17, 8, 2*)	(14, 2*, 5)	(112, 66, 22)	(350. 275, 333)
Adj. Med. ESS	(0.00*, 1*, 0.00*)	(80, 25, 17)	(25, 17, 17)	(22, 11, 5*)	(17, 5*, 5)	(126, 91, 28)	(379, 307, 362)
Adj. Max. ESS	(2*, 3*, 2*)	(94, 28, 20)	(31, 22, 17)	(25, 14, 5*)	(20, 5*, 8)	(143, 103, 31)	(410, 327, 390)

Table 3.3.2: Algorithm L2. Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. Grey color indicates that the permutation test does not support convergence. The $*$ indicates that the thinned sample size was less than 50 and permutation test was not conducted.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 88.1$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
α	(.015*, .007*, .015*)	(.286, .206, .233)	(.258, .221, .210)	(.228, .165, .172)	(.196, .081, .154)	(.318, .160, .199)	(.511, .511, .506)
Min. ESS	(.001*, .000*, .001*)	(.058, .029, .046)	(.049, .042, .035)	(.041, .026, .021)	(.035, .014, .023)	(.072, .018, .028)	(.277, .280, .273)
Med. ESS	(.002*, .001*, .002*)	(.071, .040, .051)	(.062, .051, .044)	(.050, .034, .031)	(.040, .017, .030)	(.0.80, .023, .034)	(.289, .292, .282)
Max. ESS	(.002*, .002*, .003*)	(.077, .046, .057)	(.072, .056, .054)	(.059, .041, .036)	(.045, .018, .035)	(.090, .028, .040)	(.302, .300, .301)
Adj. Min. ESS	(10*, 4*, 12*)	(525, 265, 414)	(449, 383, 316)	(371, 232, 191)	(316, 129, 207)	(658, 165, 255)	(2519, 2542, 2482)
Adj. Med. ESS	(15*, 8*, 16*)	(643, 367, 463)	(562, 463, 403)	(452, 308, 280)	(360, 151, 276)	(725, 209, 308)	(2621, 2650, 2565)
Adj. Max. ESS	(22*, 16*, 27*)	(700, 415, 515)	(650, 511, 490)	(538, 374, 328)	(410, 168, 317)	(820, 257, 367)	(2741, 2725, 2732)
$d = 49$ $\overline{CPU} = 195.2$							
α	(.000*, 0, .001*)	(.052*, .044, .068)	(.053*, .033*, .049)	(.053*, .019, .041)	(.050, .026, .052)	(.036, .065, .058)	(.351, .304, .347)
Min. ESS	(.000*, 0, .000)	(.004*, .004, .007)	(.005*, .003*, .006)	(.005*, .002, .005)	(.006, .004, .006)	(.003, .007, .006)	(.146, .004, .137)
Med. ESS	(.000*, 0, .000*)	(.006*, .006, .009)	(.007*, .005*, .007)	(.007*, .003, .007)	(.007, .005, .008)	(.004, .008, .007)	(.154, .006, .151)
Max. ESS	(.000*, 0, .001*)	(.008*, .007, .011)	(.009*, .006*, .009)	(.009*, .004, .008)	(.008, .006, .009)	(.006, .009, .009)	(.161, .007, .161)
Adj. Min. ESS	(0*, 0, 0*)	(17*, 18, 29)	(19*, 14*, 23)	(19*, 10, 21)	(23, 16, 21)	(11, 28, 23)	(597, 400, 560)
Adj. Med. ESS	(0*, 0, 1*)	(26*, 25, 38)	(29*, 20*, 30)	(30*, 14, 27)	(30, 20, 27)	(17, 32, 29)	(631, 489, 617)
Adj. Max. ESS	(1*, 0, 3*)	(32*, 28, 44)	(35*, 24*, 32)	(35*, 16, 36)	(34, 24, 31)	(24, 39, 35)	(659, 516, 659)
$d = 100$ $\overline{CPU} = 282.8$							
α	(0, 0, 0)	(.024*, .009*, .005*)	(.009*, .005*, .005*)	(.006*, .002*, .001*)	(.006*, .001*, .001*)	(.034, .028*, .008*)	(.209, .174, .199)
Min. ESS	(0, 0, 0)	(.002*, .001*, .001*)	(.001*, .001*, .001*)	(.001*, .000*, .000*)	(.001*, .000*, 0*)	(.004, .002*, .001*)	(.058, .040, .055)
Med. ESS	(0, 0, 0)	(.004*, .002*, .002*)	(.002*, .001*, .001*)	(.001*, .001*, .000*)	(.001*, .000*, 0*)	(.005, .004*, .002*)	(.065, .047, .061)
Max. ESS	(0, 0, 0)	(.004*, .002*, .002*)	(.002*, .001*, .002*)	(.002*, .001*, .001*)	(.002*, .001*, 0*)	(.006, .005*, .002*)	(.072, .052, .066)
Adj. Min. ESS	(0, 0, 0)	(7*, 3*, 1*)	(4*, 1*, 2*)	(2*, 0*, 0*)	(2*, 0*, 1*)	(11, 5*, 3*)	(163, 113, 156)
Adj. Med. ESS	(0, 0, 0)	(9*, 5*, 3*)	(5*, 2*, 3*)	(3*, 1*, 1*)	(3*, 1*, 1*)	(14, 11*, 4*)	(185, 132, 173)
Adj. Max. ESS	(0, 0, 0)	(12*, 6*, 5*)	(6*, 4*, 4*)	(5*, 3*, 2*)	(5*, 2*, 1*)	(18, 15*, 6*)	(203, 147, 188)

Table 3.3.3: Algorithm A1. Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. Grey color indicates that the permutation test does not support convergence. The $*$ indicates that the thinned sample size was less than 50 and permutation test was not conducted.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 129.6$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
α	(.065*, .107, .084)	(.422, .415, .260)	(.304, .240, .114)	(.216, .047, .072)	(.345, .103, .171)	(.259, .283, .156)	(.508, .477, .435)
Min. ESS	(.004, .009, .009)	(.161, .077, .035)	(.083, .041, .017)	(.047, .009, .012)	(.091, .017, .023)	(.062, .085, .034)	(.281, .235, .187)
Med. ESS	(.006*, .013, .011)	(.182, .159, .064)	(.104, .066, .023)	(.056, .011, .014)	(.122, .025, .038)	(.075, .103, .040)	(.296, .256, .214)
Max. ESS	(.009*, .016, .013)	(.208, .169, .081)	(.115, .072, .028)	(.061, .012, .017)	(.131, .027, .044)	(.087, .114, .047)	(.306, .270, .225)
Adj. Min. ESS	(24*, 55, 55)	(993, 475, 216)	(512, 253, 104)	(290, 55, 74)	(561, 104, 141)	(382, 524, 209)	(1734, 1450, 1154)
Adj. Med. ESS	(37*, 80, 67)	(1123, 981, 395)	(641, 407, 141)	(345, 67, 86)	(753, 154, 234)	(462, 635, 246)	(1827, 1580, 1320)
Adj. Max. ESS	(55*, 98, 80)	(1283, 1043, 499)	(709, 444, 172)	(376, 74, 104)	(808, 166, 271)	(536, 703, 290)	(1888, 1666, 1388)
$d = 49$ $\overline{CPU} = 243.5$							
α	(.064, .021*, .055*)	(.179, .143, .211)	(.040, .007*, .032)	(.007*, .000*, .002*)	(.142, .041, .081)	(.082, .080, .091)	(.277, .080, .212)
Min. ESS	(.006, .001*, .003*)	(.028, .023, .033)	(.006, .001*, .004)	(.001*, 0*, .000*)	(.029, .008, .009)	(.014, .030, .016)	(.090, .012, .047)
Med. ESS	(.008, .002*, .007*)	(.041, .032, .054)	(.008, .002*, .005)	(.002*, 0*, .001*)	(.032, .009, .012)	(.016, .037, .018)	(.098, .016, .058)
Max. ESS	(.010, .004*, .009*)	(.046, .036, .063)	(.009, .002*, .006)	(.002*, 0*, .001*)	(.038, .010, .014)	(.018, .043, .020)	(.106, .018, .062)
Adj. Min. ESS	(19, 3*, 9*)	(91, 75, 108)	(19, 3*, 13)	(3*, 0.00*, 0.00*)	(95, 26, 29)	(45, 98, 52)	(295, 39, 154)
Adj. Med. ESS	(26, 6*, 22*)	(134, 105, 177)	(26, 6*, 16)	(6*, 0.000*, 3*)	(105, 29, 39)	(52, 121, 59)	(321, 52, 190)
Adj. Max. ESS	(32, 13*, 29*)	(151, 118, 206)	(29, 6*, 19)	(6*, 0.000*, 3*)	(124, 32, 45)	(59, 141, 65)	(348, 59, 203)
$d = 100$ $\overline{CPU} = 453.2$							
α	(.005*, .001*, .008*)	(.117, .010, .052)	(.004*, 0, .001*)	(.000*, 0*, 0)	(.030, .002*, .003*)	(.064, .046, .013*)	(.034, .021, .086)
Min. ESS	(.001*, .000*, .001*)	(.014, .001, .007)	(.001*, 0, .000*)	(0*, 0*, 0)	(.004, .001*, .000*)	(.010, .006, .001*)	(.034, .003, .013)
Med. ESS	(.001*, .001*, .002*)	(.022, .002, .009)	(.001*, 0, .000*)	(0*, 0*, 0)	(.005, .001*, .001*)	(.012, .008, .003*)	(.040, .004, .016)
Max. ESS	(.002*, .001*, .002*)	(.024, .003, .011)	(.002*, 0, .001*)	(0*, 0*, 0)	(.006, .001*, .001*)	(.013, .010, .004*)	(.046, .005, .018)
Adj. Min. ESS	(1*, 0*, 1*)	(24, 1, 12)	(1*, 0.0, 0.0*)	(0.0*, 0.0*, 0.0)	(7, 1*, 0.00*)	(17, 10, 1*)	(60, 5, 22)
Adj. Med. ESS	(1*, 1*, 3*)	(38, 3, 15)	(1*, 0.00, 0.00*)	(0.00*, 0.00*, 0.00)	(8, 1*, 1*)	(21, 14, 5*)	(70, 7, 28)
Adj. Max. ESS	(3*, 2*, 3*)	(42, 5, 19)	(3*, 0.00, 1*)	(0.00*, 0.00*, 0.00)	(10, 1*, 1*)	(22, 17, 7*)	(81, 8, 31)

Table 3.3.4: Algorithm RA. Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. Grey color indicates that the permutation test does not support convergence. The $*$ indicates that the thinned sample size was less than 50 and permutation test was not conducted.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 105.5$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
α	(.047, .032, .052)	(.330, .227, .253)	(.299, .239, .225)	(.270, .015, .186)	(.226, .084, .134)	(.365, .173, .217)	(.528, .527, .523)
Min. ESS	(.004, .003, .004)	(.084, .036, .058)	(.073, .054, .044)	(.066, .025, .032)	(.049, .016, .025)	(.110, .022, .035)	(.300, .301, .294)
Med. ESS	(.005, .003, .006)	(.101, .051, .066)	(.089, .064, .058)	(.075, .033, .042)	(.055, .019, .030)	(.118, .029, .043)	(.311, .311, .305)
Max. ESS	(.006, .004, .007)	(.115, .057, .076)	(.099, .070, .067)	(.083, .038, .050)	(.060, .021, .033)	(.132, .034, .048)	(.326, .320, .321)
Adj. Min. ESS	(30, 22, 30)	(636, 272, 439)	(553, 409, 333)	(500, 189, 242)	(371, 121, 189)	(833, 166, 265)	(2273, 2281, 2228)
Adj. Med. ESS	(37, 22, 45)	(765, 386, 500)	(674, 485, 439)	(568, 250, 318)	(416, 144, 227)	(894, 219, 325)	(357, 2357, 2311)
Adj. Max. ESS	(45, 30, 53)	(871, 432, 576)	(750, 530, 507)	(629, 288, 378)	(454, 159, 250)	(1000, 257, 363)	(2471, 2425, 2425)
$d = 49$ $\overline{CPU} = 205.2$							
α	(.004*, .002*, .004*)	(.060, .045, .076)	(.057, .034, .047)	(.054, .018*, .029)	(.050, .031, .049)	(.040*, .077, .067)	(.370, .315, .366)
Min. ESS	(.000*, .000*, .000*)	(.007, .004, .009)	(.007, .004, .006)	(.007, .003*, .005)	(.006, .005, .007)	(.003*, .008, .007)	(.164, .107, .154)
Med. ESS	(.001*, .000*, .001*)	(.008, .006, .011)	(.009, .006, .008)	(.009, .004*, .005)	(.008, .006, .008)	(.005*, .010, .009)	(.173, .130, .170)
Max. ESS	(.001*, .001*, .001*)	(.010, .008, .012)	(.010, .006, .009)	(.010, .004*, .006)	(.009, .007, .009)	(.007*, .012, .011)	(.184, .138, .180)
Adj. Min. ESS	(0.0*, 0.0*, 0.0*)	(27, 15, 35)	(27, 15, 23)	(27, 11*, 19)	(23, 19, 27)	(11*, 31, 27)	(639, 417, 600)
Adj. Med. ESS	(3*, .000*, 3*)	(31, 23, 42)	(35, 23, 31)	(35, 15*, 19)	(31, 23, 31)	(19*, 38, 35)	(674, 506, 662)
Adj. Max. ESS	(3*, 3*, 3*)	(38, 31, 46)	(38, 23, 35)	(38, 15*, 23)	(35, 27, 35)	(27*, 46, 42)	(717, 537, 701)
$d = 100$ $\overline{CPU} = 332.3$							
α	(.000*, .000*, .000*)	(035, .008*, .004*)	(.011, .006*, .003*)	(.007*, .004*, .001*)	(.005*, .002*, .001*)	(.005, .041*, .010*)	(.228, .189, .217)
Min. ESS	(0*, 0*, 0*)	(.004, .001*, .001*)	(.002, .001*, .000*)	(.001*, .000*, .000*)	(.000*, .000*, .000*)	(.006, .004*, .001*)	(.072, .050, .067)
Med. ESS	(0*, 0*, 0*)	(.005, .002*, .001*)	(.002, .001*, .001*)	(.001*, .001*, .000*)	(.001*, .001*, .000*)	(.007, .006*, .002*)	(.079, .059, .074)
Max. ESS	(0*, 0*, 0*)	(.007, .002*, .002*)	(.003, .002*, .001*)	(.002*, .001*, .001*)	(.002*, .001*, .001*)	(.009, .007*, .003*)	(.089, .064, .080)
Adj. Min. ESS	(0.0*, 0.0*, 0.0*)	(9, 2*, 2*)	(4, 2*, 0.00*)	(2*, 0.00*, 0.00*)	(0.00*, 0.00*, 0.00*)	(14, 9*, 2*)	(173, 120, 161)
Adj. Med. ESS	(0.0*, 0.0*, 0.0*)	(12, 4*, 2*)	(4, 2*, 2*)	(2*, 2*, 0.0*)	(2*, 2*, 0.0*)	(16, 14*, 4*)	(190, 142, 178)
Adj. Max. ESS	(0.0*, 0.0*, 0.0*)	(16, 4*, 4*)	(7, 4*, 2*)	(4*, 2*, 2*)	(4*, 2*, 2*)	(21, 16*, 7*)	(214, 154, 192)

Table 3.3.5: Algorithm iRA. Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. Grey color indicates that the permutation test does not support convergence. The $*$ indicates that the thinned sample size was less than 50 and permutation test was not conducted.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 105.3$	($\sigma^2 = 1, \phi = 10$)	($\sigma^2 = 0.3, \phi = 10$)	($\sigma^2 = 1, \phi = 10$)	($\sigma^2 = 3, \phi = 10$)	($\sigma^2 = 1, \phi = 1$)	($\sigma^2 = 1, \phi = 100$)	($\sigma^2 = 1, \phi = 10$)
α	(.153, .211, .192)	(.451, .501, .467)	(.375, .411, .379)	(.297, .338, .288)	(.257, .281, .268)	(.481, .548, .528)	(.536, .538, .536)
Min. ESS	(.012, .020, .020)	(.190, .247, .195)	(.126, .161, .125)	(.069, .106, .0629)	(.055, .062, .055)	(.204, .296, .255)	(.316, .317, .311)
Med. ESS	(.015, .025, .024)	(.199, .261, .217)	(.139, .170, .138)	(.084, .115, .078)	(.062, .074, .066)	(.219, .321, .284)	(.321, .324, .320)
Max. ESS	(.022, .034, .029)	(.213, .282, .235)	(.147, .179, .155)	(.094, .128, .08)	(.074, .094, .081)	(.235, .343, .310)	(.327, .330, .331)
Adj. Min. ESS	(92, 154, 154)	(1471, 1913, 1510)	(976, 1247, 968)	(534, 821, 487)	(426, 480, 426)	(1580, 2292, 1975)	(2447, 2455, 2409)
Adj. Med. ESS	(116, 193, 185)	(1541, 2021, 1680)	(1076, 1316, 1068)	(650, 890, 604)	(480, 573, 511)	(1696, 2486, 2199)	(2486, 2509, 2478)
Adj. Max. ESS	(170, 263, 224)	(1649, 2184, 1820)	(1138, 1386, 1200)	(728, 991, 619)	(573, 728, 627)	(1820, 2656, 2401)	(2532, 2556, 2563)
$d = 49$ $\overline{CPU} = 204.8$							
α	(.092, .100, .087)	(.356, .356, .341)	(.254, .278, .262)	(.163, .191, .185)	(.116, .134, .116)	(.417, .416, .405)	(.390, .390, .391)
Min. ESS	(.006, .008, .006)	(.140, .145, .126)	(.063, .078, .069)	(.024, .032, .027)	(.016, .019, .014)	(.196, .203, .187)	(.186, .180, .183)
Med. ESS	(.008, .011, .009)	(.150, .152, .136)	(.070, .088, .077)	(.029, .039, .037)	(.018, .022, .017)	(.217, .215, .201)	(.192, .191, .193)
Max. ESS	(.012, .013, .011)	(.160, .162, .143)	(.079, .096, .082)	(.033, .046, .044)	(.021, .025, .021)	(.229, .227, .212)	(.198, .198, .199)
Adj. Min. ESS	(23, 31, 23)	(546, 566, 492)	(246, 304, 269)	(93, 124, 105)	(62, 74, 54)	(765, 792, 730)	(726, 702, 714)
Adj. Med. ESS	(31, 42, 35)	(585, 593, 531)	(273, 343, 300)	(113, 152, 144)	(70, 85, 66)	(847, 839, 784)	(749, 745, 753)
Adj. Max. ESS	(46, 50, 42)	(624, 632, 558)	(308, 374, 320)	(128, 179, 171)	(82, 97, 82)	(894, 886, 827)	(773, 773, 777)
$d = 100$ $\overline{CPU} = 322.7$							
α	(.015*, .013*, .023)	(.178, .198, .229)	(.107, .116, .127)	(.052, .049, .067)	(.030*, .034, .027*)	(.226, .229, .275)	(.248, .242, .247)
Min. ESS	(.002*, .000*, .003)	(.036, .047, .069)	(.016, .017, .023)	(.006, .005, .008)	(.002*, .003, .002*)	(.065, .067, .105)	(.086, .082, .089)
Med. ESS	(.002*, .002*, .003)	(.043, .055, .076)	(.019, .020, .026)	(.007, .007, .010)	(.004*, .005, .004*)	(.071, .073, .113)	(.095, .090, .094)
Max. ESS	(.003*, .003*, .004)	(.048, .063, .084)	(.021, .023, .028)	(.009, .008, .012)	(.005*, .006, .005*)	(.077, .082, .120)	(.101, .097, .102)
Adj. Min. ESS	(4*, 0.0*, 7)	(89, 116, 171)	(39, 42, 57)	(14, 12, 19)	(4*, 7, 4*)	(161, 166, 260)	(213, 203, 220)
Adj. Med. ESS	(4*, 4*, 7)	(106, 136, 188)	(47, 49, 64)	(17, 17, 24)	(9*, 12, 9*)	(175, 180, 280)	(235, 223, 232)
Adj. Max. ESS	(7*, 7*, 9)	(118, 156, 208)	(52, 57, 69)	(22, 19, 29)	(12*, 14, 12*)	(190, 203, 297)	(250, 240, 252)

Table 3.3.6: Algorithm Christensen et al. (2006). Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 199.8$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min. ESS	(.120, .115, .115)	(.153, .139, .146)	(.140, .138, .141)	(.141, .121, .138)	(.140, .113, .121)	(.162, .156, .154)	(.173, .170, .171)
Med. ESS	(.179, .168, .184)	(.184, .178, .184)	(.179, .166, .175)	(.169, .160, .164)	(.151, .127, .142)	(.185, .183, .183)	(.179, .182, .180)
Max. ESS	(.258, .236, .238)	(.199, .212, .222)	(.210, .232, .232)	(.189, .270, .233)	(.201, .431, .239)	(.196, .224, .217)	(.200, .195, .201)
Adj. Min. ESS	(480, 460, 460)	(612, 556, 584)	(560, 552, 564)	(564, 484, 552)	(560, 452, 484)	(648, 624, 616)	(692, 680, 684)
Adj. Med. ESS	(716, 672, 736)	(736, 712, 736)	(716, 664, 700)	(676, 640, 656)	(604, 508, 568)	(740, 732, 732)	(716, 728, 720)
Adj. Max. ESS	(1032, 944, 952)	(796, 848, 888)	(840, 928, 928)	(756, 1080, 932)	(804, 1725, 956)	(784, 896, 868)	(800, 780, 804)
$d = 49$ $\overline{CPU} = 453.3$							
Min. ESS	(.086, .084, .087)	(.110, .103, .107)	(.101, .102, .095)	(.097, .086, .084)	(.098, .088, .100)	(.120, .110, .124)	(.128, .120, .126)
Med. ESS	(.135, .137, .133)	(.137, .136, .139)	(.128, .129, .131)	(.119, .112, .116)	(.110, .100, .108)	(.137, .139, .138)	(.136, .135, .135)
Max. ESS	(.175, .175, .201)	(.174, .173, .167)	(.194, .195, .212)	(.184, .258, .222)	(.196, .267, .178)	(.154, .158, .159)	(.150, .187, .135)
Adj. Min. ESS	(151, 148, 153)	(194, 181, 188)	(178, 179, 167)	(171, 151, 148)	(172, 155, 176)	(211, 194, 218)	(225, 211, 222)
Adj. Med. ESS	(238, 241, 234)	(241, 239, 245)	(225, 227, 231)	(209, 197, 204)	(194, 176, 190)	(241, 245, 243)	(239, 238, 238)
Adj. Max. ESS	(308, 308, 354)	(307, 305, 294)	(342, 344, 374)	(324, 455, 391)	(345, 471, 314)	(271, 278, 280)	(264, 329, 238)
$d = 100$ $\overline{CPU} = 911.4$							
Min. ESS	(.072, .083, .064)	(.086, .083, .080)	(.075, .078, .070)	(.070, .064, .058)	(.071, .063, .065)	(.092, .090, .087)	(.095, .089, .092)
Med. ESS	(.101, .101, .102)	(.103, .102, .103)	(.096, .098, .098)	(.091, .090, .086)	(.079, .072, .073)	(.103, .104, .104)	(.101, .100, .101)
Max. ESS	(.145, .130, .131)	(.128, .132, .127)	(.165, .143, .138)	(.187, .153, .157)	(.154, .218, .257)	(.114, .117, .120)	(.115, .137, .116)
Adj. Min. ESS	(63, 72, 56)	(75, 72, 70)	(65, 68, 61)	(61, 56, 50)	(62, 55, 57)	(80, 78, 76)	(83, 78, 80)
Adj. Med. ESS	(88, 88, 89)	(90, 89, 90)	(84, 86, 86)	(79 78, 75)	(69, 63, 64)	(90, 91, 91)	(88, 87, 88)
Adj. Max. ESS	(127, 114, 114)	(112, 115, 111)	(144, 125, 121)	(164, 134, 137)	(135, 191, 225)	(100, 102, 105)	(100, 120, 101)

Table 3.3.7: Algorithm pMMALA. Acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
$\overline{CPU} = 165.1$							
Min. ESS	(.090, .113, .102)	(.145, .159, .120)	(.101, .102, .058)	(.053, .035, .014)	(.037, .022, .010)	(.124, .143, .116)	(.103, .073, .048)
Med. ESS	(.122, .134, .128)	(.176, .175, .164)	(.151, .141, .113)	(.122, .086, .035)	(.069, .075, .027)	(.142, .153, .140)	(.154, .126, .078)
Max. ESS	(.199, .152, .230)	(.210, .229, .256)	(.285, .357, .408)	(.366, .585, .652)	(.686, .571, .711)	(.161, .168, .166)	(.340, .465, .590)
Adj. Min. ESS	(436, 547, 494)	(70, 770, 581)	(489, 494, 281)	(256, 169, 67)	(179, 106, 48)	(600, 693, 562)	(499, 353, 232)
Adj. Med. ESS	(591, 649, 620)	(852, 848, 794)	(731, 683, 547)	(591, 416, 169)	(334, 363, 130)	(688, 741, 678)	(746, 610, 378)
Adj. Max. ESS	(964, 736, 1114)	(1017, 1109, 1240)	(1381, 1730, 1977)	(1773, 2835, 3159)	(3324, 2767, 3445)	(780, 814, 804)	(1647, 2253, 2859)
$d = 49$							
$\overline{CPU} = 380.6$							
Min. ESS	(.093, .075, .089)	(.097, .089, .103)	(.056, .047, .060)	(.021, .010, .016)	(.019, .011, .010)	(.098, .099, .096)	(.046, .016, .038)
Med. ESS	(.115, .101, .119)	(.128, .131, .136)	(.109, .095, .112)	(.062, .036, .048)	(.060, .048, .024)	(.111, .119, .116)	(.101, .094, .097)
Max. ESS	(.172, .211, .149)	(.183, .178, .176)	(.214, .424, .358)	(.370, .673, .066)	(.349, .550, .543)	(.125, .142, .124)	(.305, .052, .508)
Adj. Min. ESS	(195, 157, 187)	(203, 187, 216)	(117, 98, 126)	(44, 21, 33)	(39, 23, 21)	(205, 208, 201)	(96, 33, 79)
Adj. Med. ESS	(241, 212, 250)	(268, 275, 285)	(229, 199, 235)	(130, 75, 100)	(126, 100, 50)	(233, 250, 243)	(212, 197, 203)
Adj. Max. ESS	(361, 443, 313)	(384, 374, 369)	(449, 891, 752)	(777, 1414, 138)	(733, 1155, 1141)	(262, 298, 260)	(640, 109, 1067)
$d = 100$							
$\overline{CPU} = 831.5$							
Min. ESS	(.055, .046, .059)	(.083, .061, .073)	(.050, .022, .040)	(.019, .005, .012)	(.010, .008, .008)	(.076, .066, .070)	(.032, .012, .024)
Med. ESS	(.084, .088, .089)	(.099, .102, .100)	(.087, .077, .083)	(.054, .035, .044)	(.034, .030, .028)	(.088, .093, .082)	(.078, .061, .074)
Max. ESS	(.118, .157, .129)	(.131, .139, .127)	(.162, .266, .227)	(.329, .464, .476)	(.476, .530, .393)	(.102, .107, .098)	(.229, .355, .327)
Adj. Min. ESS	(52, 44, 56)	(79, 58, 70)	(48, 21, 38)	(18, 4, 11)	(9, 7, 7)	(73, 63, 67)	(30, 11, 23)
Adj. Med. ESS	(80, 84, 85)	(95, 98, 96)	(83, 74, 79)	(51, 33, 42)	(32, 28, 26)	(84, 89, 78)	(75, 58, 71)
Adj. Max. ESS	(113, 151, 124)	(126, 133, 122)	(155, 255, 218)	(316, 446, 457)	(457, 509, 378)	(98, 102, 94)	(220, 341, 314)

Investigation of the accuracy of our proposals

To visualise the constructed proposals and investigate how accurately they approximate the target distribution we assess the contours of our proposals. We consider a bivariate example where all parameters' values are fixed apart from the correlation parameter ϕ. The prior mean of the process is set equal to $\boldsymbol{\mu}_\eta = (\log(10), \log(10))$ and the prior variance $\sigma^2 = 1$. Regarding the correlation structure, we use the exponential function, as in the simulation study, we set the distance between the two points to be equal to 1 and use three different values for $\phi \in \{1, 10, 100\}$. Finally, the Poisson observations are set to be $\boldsymbol{y} = (10, 10)$.

Figures 3.2-3.4 display the contours of the log target, in black, with the contours of the log proposals superimposed in red. Figure 3.2 corresponds to the case $\phi = 1$, Figure 3.3 to $\phi = 10$ and Figure 3.4 to the case $\phi = 100$. The top line of each Figure shows the proposals of the algorithms L1, L2, the middle line the proposals of RA and iRA on the $\boldsymbol{\eta}$ scale and the bottom line shows the proposals of A both on $\boldsymbol{\psi}$ and $\boldsymbol{\eta}$ scale.

As we can see, neither the posterior of $\boldsymbol{\eta}$ nor that of $\boldsymbol{\psi}$ is close to Gaussian. However, the contours of $\boldsymbol{\eta}$ are closer to ellipses than the contours of $\boldsymbol{\psi}$, especially as the dependence increases. The effect of using a Student's t_{10} proposal is clearly illustrated in these graphs as all the proposals have much heavier tails than the target. At least for this two dimensional example, the first four algorithms, namely, L1, L2, RA and iRA appear to represent the posterior better than A. All proposals seem to approximate the target reasonably well around the mode but fail to capture its shape as we move into the tails.

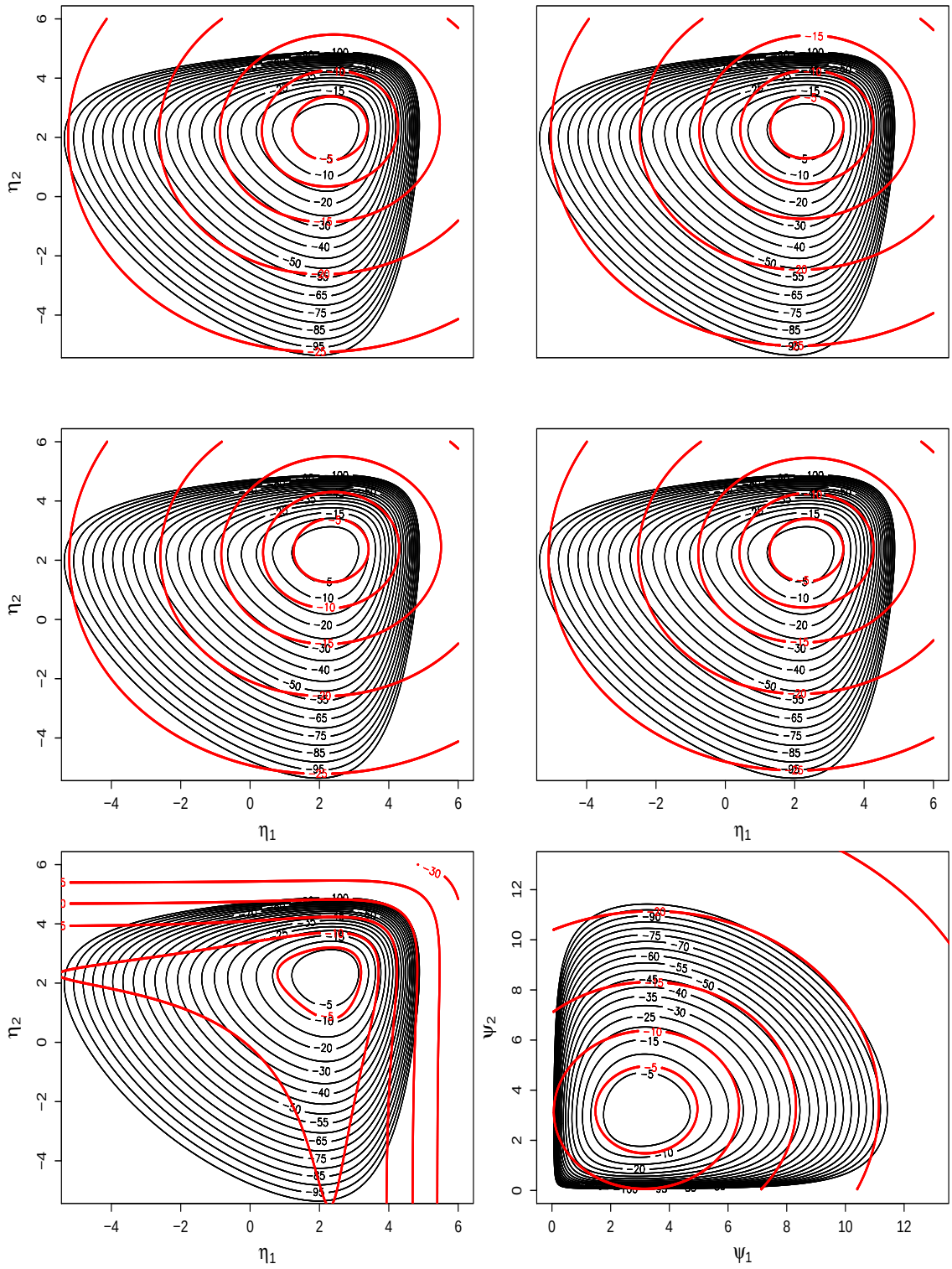

Figure 3.2: Contours of bivariate log-target (Black lines) and log-proposals (Red lines) distribution. Top row: Proposals of the L1 (left) and L2 (right) algorithms. Middle row: Proposals of the RA (left) and iRA (right) proposals. Bottom row: Proposal of A algorithm on $\boldsymbol{\eta}$ scale (left) and $\boldsymbol{\psi}$ scale (right). Parameters' values fixed to be $\boldsymbol{y} = (10, 10)$, $\boldsymbol{\mu}_\eta = (\log(10), \log(10))$, $\sigma^2 = 1$, $\phi = 1$ and distance between points set equal to 1.

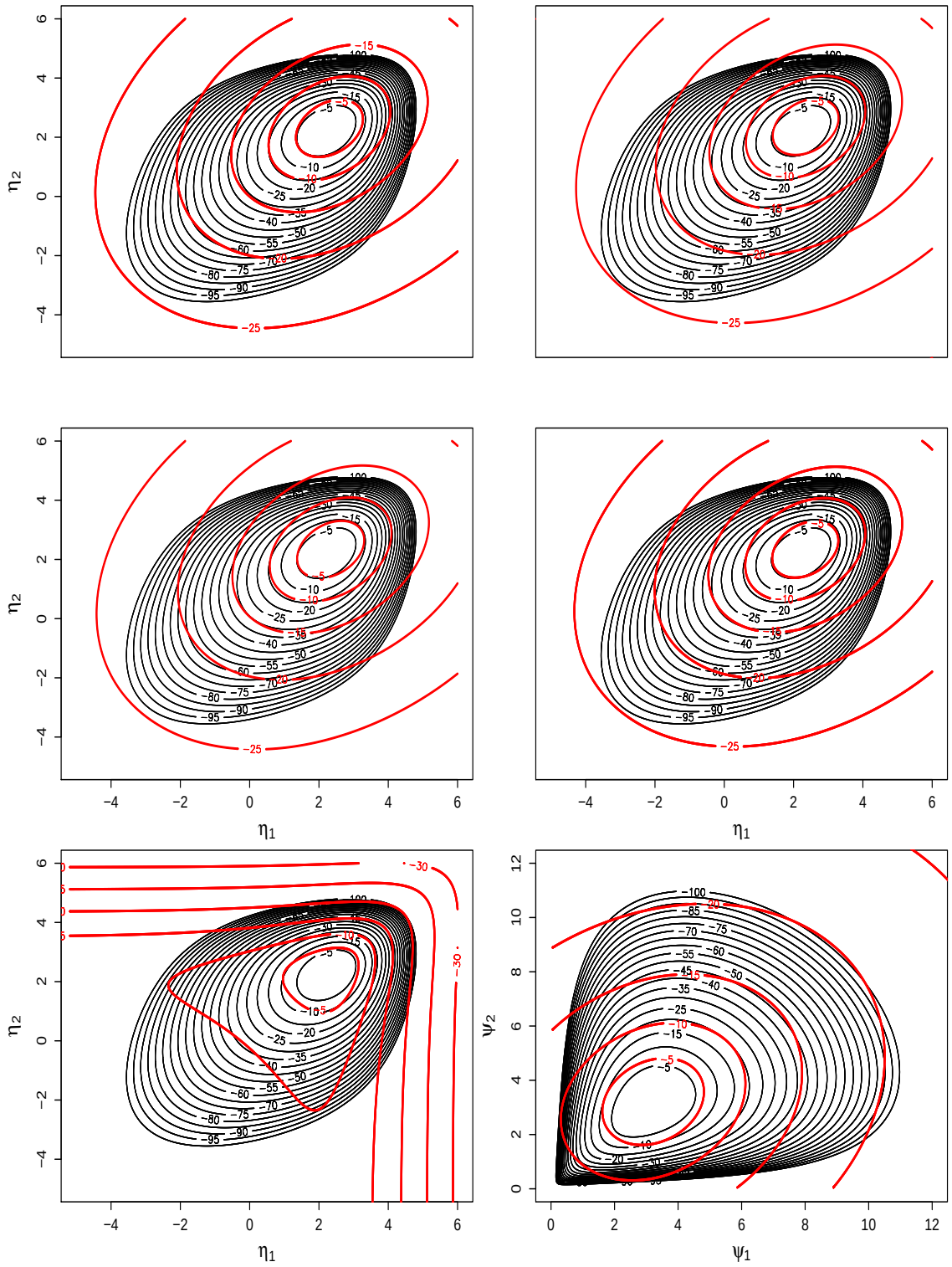

Figure 3.3: Contours of bivariate log-target (Black lines) and log-proposals (Red lines) distribution. Top row: Proposals of the L1 (left) and L2 (right) algorithms. Middle row: Proposals of the RA (left) and iRA (right) proposals. Bottom row: Proposal of A algorithm on $\boldsymbol{\eta}$ scale (left) and $\boldsymbol{\psi}$ scale (right). Parameters' values fixed to be $\boldsymbol{y} = (10, 10)$, $\boldsymbol{\mu}_\eta = (\log(10), \log(10))$, $\sigma^2 = 1$, $\phi = 10$ and distance between points set equal to 1.

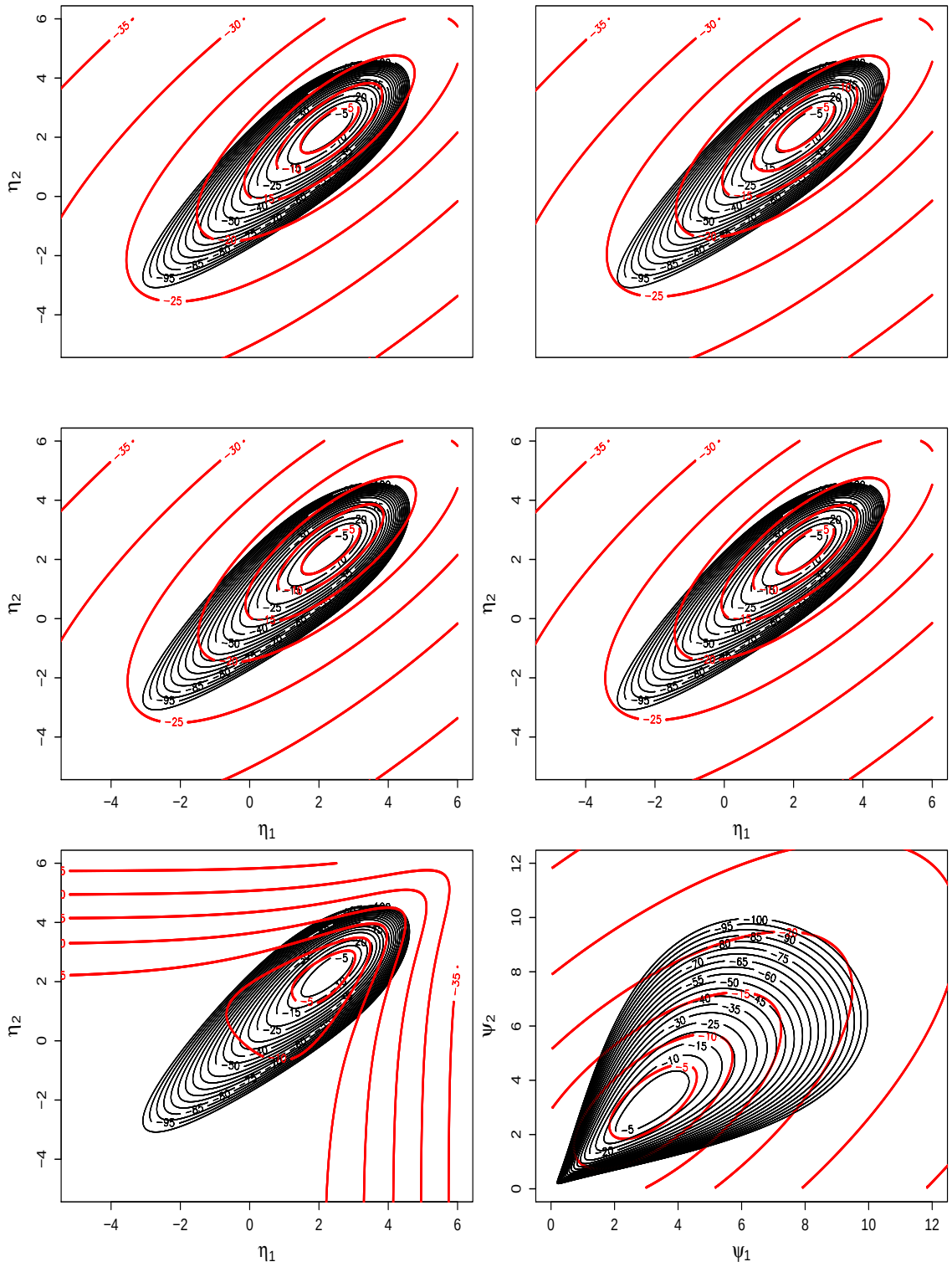

Figure 3.4: Contours of bivariate log-target (Black lines) and log-proposals (Red lines) distribution. Top row: Proposals of the L1 (left) and L2 (right) algorithms. Middle row: Proposals of the A1 (left) and A2 (right) algorithms on $\boldsymbol{\eta}$ scale. Bottom row: Proposals of the A1 (left) and A2 (right) algorithms on $\boldsymbol{\psi}$ scale. Parameters' values fixed to be $\boldsymbol{y} = (10, 10)$, $\boldsymbol{\mu}_\eta = (\log(10), \log(10))$, $\sigma^2 = 1$, $\phi = 100$ and distance between points set equal to 1.

3.4 Discussion

In this Chapter, we have presented five MHIS proposals which propose a joint update on all latent variables using a Gaussian approximation to the posterior distribution. The first two algorithms, L1 and L2, employ a transformation of the data using the link function. The remaining algorithms apply the Anscombe transformation to the data and also utilise a transformation of the parameters in order to create an approximate Gaussian posterior.

In Section 3.3 we compared our algorithms against each other and against the algorithms existing in the literature and assessed their performance on a simple Poisson GLSM over a variety of prior parameters' values.

We have found that a single global Gaussian approximation to the posterior does not provide an efficient MHIS proposal, mainly due to the non-Gaussian shape of the target of interest. The initial approach of exploiting the structure of the Poisson GLSM and use the link function to construct appropriate transformations of the data proved to be the best performing (algorithm L1). In theory algorithm L2 should perform better than L1 since the transformation used leads to smaller approximation errors. However, we found that the resulting smaller variance and the altered mean of the proposal had a negative effect on the algorithm leading to a very poor performance.

As far as Anscombe's transformation is concerned, linearising $\boldsymbol{\psi}$ should overcome the problem of approximating the log-normal prior of $\boldsymbol{\psi}$ by moment matching. These two algorithms, namely RA and iRA, indeed provided better results than A but still did not outperform L1.

Overall, the higher the prior mean of $\boldsymbol{\eta}$ is, the better our algorithms perform, with L1

performing even better than the algorithm of Christensen et al. (2006). Provided that σ^2 is small, if $\boldsymbol{\mu}_\eta$ was large then it would give rise to large η_i's which, in turn, generate large observations y_i. Therefore, one could identify in advance, based on the observed data, whether L1 would perform at least as well as the algorithm of Christensen et al. (2006) avoiding all the additional computational complexity of the latter.

Finally, the performance of our algorithms is greatly affected by increases in the dimension of the target. One possible solution could be to construct univariate proposals which update each component of $\boldsymbol{\eta}$ separately and resort to MH schemes other than the MHIS. If such proposals could be combined with a way of capturing the dependence structure of $\boldsymbol{\eta}$, in cases where the data are weak resulting in high posterior dependence, this could give rise to an efficient MCMC scheme. Such an approach is investigated in the following chapter.

As a final note, we would like to consider a potential improvement of algorithm L1. Let $\boldsymbol{\eta}^{prop} \sim \text{MVN}\left(\boldsymbol{\mu}_{\eta|y^l}, \boldsymbol{\Sigma}_{\eta|y^l}\right)$ as derived in Section 3.1.1, Expression (3.1.5) to Expression (3.1.7). Consider now a new proposal, $\boldsymbol{\eta}^{prop*}$, of the form,

$$\boldsymbol{\eta}^{prop*} = \boldsymbol{\mu}_{\eta|y^l} + \sqrt{1-\varepsilon^2}\left(\boldsymbol{\eta}^{cur} - \boldsymbol{\mu}_{\eta|y^l}\right) + \varepsilon\left(\boldsymbol{\eta}^{prop} - \boldsymbol{\mu}_{\eta|y^l}\right)$$

where $\varepsilon \in (0, 1]$ that would, in a way, weight the new proposed value for $\boldsymbol{\eta}$ according to the current value and $\boldsymbol{\eta}^{prop}$. At stationarity and if the posterior distribution were truly equal to its MVN $\left(\boldsymbol{\mu}_{\eta|y^l}, \boldsymbol{\Sigma}_{\eta|y^l}\right)$ approximation then $\boldsymbol{\eta}^{prop}$ would have an expectation of $\boldsymbol{\mu}_{\eta|y^l}$ and variance of $\boldsymbol{\Sigma}_{\eta|y^l}$ and so the acceptance probability for the move would be 1. If $\varepsilon = 1$, then we recover exactly $\boldsymbol{\eta}^{prop}$ whereas when $\varepsilon = 0$ we recover $\boldsymbol{\eta}^{cur}$. Our suggestion is to use $\boldsymbol{\eta}^{prop*}$ as the proposed value in the MCMC scheme where ε is set to

be sufficiently small, e.g., 0.05. In that way we will always propose a value closer to the current one avoiding proposals very far away from what already has been accepted. Given the low computational cost of the algorithm defining an efficient value for ε through trial and error should not be problematic.

CHAPTER 4

Single component MH proposals for correlated latent variables

In this chapter, as in Chapter 3, we employ MH proposals with a t- distribution. In contrast with the approach of the previous Chapter, however, our proposals are univariate and are, in general, not independence samplers. The general approach of Chapter 3 was to utilise Gaussian approximations to the likelihood in order to obtain a a joint Gaussian approximation to the posterior distribution of the latent process $\boldsymbol{\eta}$. Here however, we create univariate proposals using the Laplace approximation, by matching the mode and curvature of the conditional log posterior, and each component of the latent process is updated separately.

In Section 4.1 we introduce a simple MHIS within Gibbs algorithm in order to draw samples from the posterior distribution of the latent process using an approximation to the marginal posterior of each, component of the latent process, s_i, ignoring any existing correlation structure. The MHIS algorithm itself is not expected to be efficient

when the posterior correlation is strong; however it is a special case of, and will also serve as the basis for, a more sophisticated MH scheme (Section 4.2) that accounts for the correlation structure of the latent process by conditioning on the most important principal components of the prior correlation matrix. Section 4.2.2 we complete this MCMC scheme by improving its mixing through an update on the principal components. Section 4.2.3 outlines the issues associated with the number of principal on which we should condition and provides a diagnostic tool for this purpose. In Section 4.3.2 we illustrate results of these algorithm and assess their performance over various scenarios of parameter values.

4.1 A single component MHIS

Recall that the prior distribution of the latent variable $\boldsymbol{\eta}$, $p(\boldsymbol{\eta})$, is a multivariate normal distribution with mean $\boldsymbol{\mu}_\eta = \boldsymbol{F\beta}$ and covariance matrix $\boldsymbol{V}_\eta = \sigma^2\boldsymbol{R}$, and that this gives rise to Poisson observations $\boldsymbol{y}|\boldsymbol{\eta}$ such that $y_i|\eta_i \sim \text{Pois}(e^{\eta_i})$. In the following, we denote the Poisson likelihood by $L(\boldsymbol{\eta})$ and the resulting posterior by $\pi(\boldsymbol{\eta}|\boldsymbol{y})$ as in Chapter 3.

To create a simple approximation to the posterior, let us ignore the prior correlation structure of the process $\boldsymbol{\eta}$ and assume that $\boldsymbol{\eta} \sim \text{MVN}(\boldsymbol{\mu}_\eta, \sigma^2\boldsymbol{I})$. Assuming independence between the components of $\boldsymbol{\eta}$, and since the likelihood $L(\boldsymbol{\eta})$ is the product $\prod_{i=1}^n L(\eta_i)$, we may obtain an approximate posterior distribution $\tilde{\pi}(\boldsymbol{\eta}|\boldsymbol{y})$, which can be factorised into the following product,

$$\tilde{\pi}(\boldsymbol{\eta}|\boldsymbol{y}) = \prod_{i=1}^{n} \tilde{\pi}(\eta_i|y_i) = \prod_{i=1}^{n} \tilde{\pi}_i, \tag{4.1.1}$$

where $\tilde{\pi}_i$ denotes the approximation to the posterior distribution of the i-th component of $\boldsymbol{\eta}$. We propose an MHIS within Gibbs algorithm where each component, η_i, is updated

separately through a Student's t proposal, $q(\cdot)$, with location and scale parameters given respectively by the mode and curvature at the mode of $\tilde{\pi}_i$. If we denote the mode of $\tilde{\pi}_i$ by $\hat{\eta}_i$ and the inverse of the negative curvature at the mode by τ_i then our proposal will be

$$q_i(\eta_i) = q(\eta_i|y_i) \equiv t_{10}(\eta_i; \hat{\eta}_i, \tau_i).$$

Let, η_i^{prop}, η_i^{cur} be the proposed and current value of the i-th component of $\boldsymbol{\eta}^{prop}$ and $\boldsymbol{\eta}^{cur}$. The proposed algorithm would be as follows.

- Set initial values $\boldsymbol{\eta}^{cur} = \boldsymbol{\eta}^0$
- For i in $1:d$
  - Evaluate $\hat{\eta}_i$ and τ_i, the location and scale parameters of the proposal, by maximising $\tilde{\pi}_i(\eta_i|y_i)$
- For j in $1:J$
  - For i in $1:d$
    - Propose, η_i^{prop} from $q_i(\eta_i^{prop})$
    - Accept with probability,

$$\alpha(\boldsymbol{\eta}^{cur}, \boldsymbol{\eta}^{prop}) = \min\left(1, \frac{\pi(\boldsymbol{\eta}^{prop}|\boldsymbol{y})q(\eta_i^{cur}|y_i)}{\pi(\boldsymbol{\eta}^{cur}|\boldsymbol{y})q(\eta_i^{prop}|y_i)}\right)$$

    - If $\boldsymbol{\eta}^{prop}$ is accepted, set $\boldsymbol{\eta}^{cur} = \boldsymbol{\eta}^{prop}$

We will call this algorithm U-MHIS, where U stands for univariate updates. A naive implementation of U-MHIS would be computationally expensive since at each iteration, we would evaluate the quadratic form in the multivariate normal prior of $\boldsymbol{\eta}$. Therefore, the computational cost, at a single iteration, j (which loops through all d components),

would be $\mathcal{O}(d^3)$. Additionally, although the maximisation of $\tilde{\pi}(\eta_i|y_i)$ or equivalently $\log\{\tilde{\pi}(\eta_i|y_i)\}$ needs to be performed only once at the beginning of the algorithm, the computational time may be large if an appropriate interval for the maximisation is not provided. We now illustrate ways to reduce the computational demand of this algorithm. These same ideas will also be useful in the algorithms developed later in this Chapter.

4.1.1 Reduction of computational cost

Maximisation of $\tilde{\pi}(\eta_i|y_i)$

We first deal with the performance of the maximisation and provide tight bounds on the location of the maximum. For ease of notation we drop the subscript i. To begin with, for some constants, c and $c^* = c + \frac{1}{2}\sigma^2 y^2$

$$\begin{aligned} \log\{\tilde{\pi}(\eta|y)\} &= c - \frac{1}{2\sigma^2}(\eta - \mu_\eta)^2 + \eta y - e^\eta \\ &= c^* - \frac{1}{2\sigma^2}(\eta - \mu^*)^2 - e^\eta, \end{aligned} \tag{4.1.2}$$

where $\mu^* = \mu_\eta + \sigma^2 y$. The mode, $\hat{\eta}$, therefore satisfies,

$$\hat{\eta} = \mu^* - \sigma^2 e^{\hat{\eta}}. \tag{4.1.3}$$

This provides us with a first lower bound as $\hat{\eta} < \mu^*$ and hence an upper bound is $\hat{\eta} > \mu^* - \sigma^2 e^{\mu^*}$. The width of this interval is $\sigma^2 e^{\mu^*}$ which can potentially be large. Making use of the Taylor series for the exponential and logarithmic function, tighter bounds can be obtained by the following result.

Proposition 4.1.1. *The solution $\hat{\eta}$ of equation (4.1.3) satisfies the following bounds.*

Without any constraints on the parameter values (except for $\sigma^2 > 0$) the following bounds

hold,

$$\mu^* - \sigma^2 e^{\mu^*} < \hat{\eta} < \frac{\mu^* - \sigma^2}{1 + \sigma^2}.$$

In addition, if $\mu^* - \sigma^2 e^{\mu^*} > -1$ *then a lower bound is,*

$$\hat{\eta} > \max\left\{ \log\left(\frac{1+\mu^*}{1+\sigma^2}\right), \mu^* - \sigma^2 e^{\mu^*} \right\},$$

and when $\mu^* > 0$ *then an upper bound is,*

$$\hat{\eta} < \min\left\{ \frac{\mu^* - \sigma^2}{1+\sigma^2}, \frac{\mu^*}{1+\mu^*} \log\left(\frac{\mu^*}{\sigma^2}\right) \right\}. \tag{4.1.4}$$

Proof. From equation (4.1.3) we have already obtained the bounds $\mu^* - \sigma^2 e^{\mu^*} < \hat{\eta} < \mu^*$. Now, $e^{\hat{\eta}} > 1 + \hat{\eta}$, so,

$$\begin{aligned} \hat{\eta} &= \mu^* - \sigma^2 e^{\hat{\eta}} < \mu^* - \sigma^2(1+\hat{\eta}) \\ \Rightarrow \hat{\eta} &< \frac{\mu^* - \sigma^2}{1+\sigma^2}. \end{aligned} \tag{4.1.5}$$

Next, for $\hat{\eta} > -1$, $\log(1+\hat{\eta}) < \hat{\eta}$, so using (4.1.3),

$$\begin{aligned} \log(1 + \mu^* - \sigma^2 e^{\hat{\eta}}) &\leq \hat{\eta} \\ \Rightarrow e^{\hat{\eta}} &\geq \frac{1+\mu^*}{1+\sigma^2} \\ \Rightarrow \hat{\eta} &\geq \log\left(\frac{1+\mu^*}{1+\sigma^2}\right). \end{aligned} \tag{4.1.6}$$

Here, since $\hat{\eta} > \mu^* - \sigma^2 e^{\mu^*}$, Equation (4.1.6) is certainly valid when $\mu^* > -1 + \sigma^2 e^{\mu^*}$. In order to obtain the last upper bound for $\hat{\eta}$ we consider Equation (4.1.3) and rearrange

as follows. Provided $\mu^* > 0$ and since $\hat{\eta} < \mu^*$

$$\begin{aligned}
\hat{\eta} = \mu^* - \sigma^2 e^{\hat{\eta}} \Rightarrow e^{\hat{\eta}} &= \frac{\mu^* - \hat{\eta}}{\sigma^2} \\
\Rightarrow \hat{\eta} &= \log\left(\frac{\mu^*}{\sigma^2}\right) + \log\left(1 - \frac{\hat{\eta}}{\mu^*}\right) \\
&\leq \log\left(\frac{\mu^*}{\sigma^2}\right) - \frac{\hat{\eta}}{\mu^*} \\
\Rightarrow \hat{\eta} &\leq \left(\frac{\mu^*}{\mu^* + 1}\right) \log\left(\frac{\mu^*}{\sigma^2}\right). \qquad (4.1.7)
\end{aligned}$$

Hence, combining the lower and upper bounds according to the value of μ^* we obtain the desired result. □

Having found tight upper and lower bounds for the maximum of $\tilde{\pi}(\eta_i|y_i)$ we proceed by illustrating how the computational cost of the acceptance probability can be reduced.

Calculation of acceptance probability

Let

$$r(\boldsymbol{\eta}^{cur}, \boldsymbol{\eta}^{prop}) = \frac{\pi(\boldsymbol{\eta}^{prop}|\boldsymbol{y})q(\eta_i^{cur}|y_i)}{\pi(\boldsymbol{\eta}^{cur}|\boldsymbol{y})q(\eta_i^{prop}|y_i)},$$

be the ratio in the MH acceptance probability and since in practice we work on the logarithmic scale consider,

$$\begin{aligned}
\log\{r(\boldsymbol{\eta}^{cur}, \boldsymbol{\eta}^{prop})\} &= \log\{\pi(\boldsymbol{\eta}^{prop}|\boldsymbol{y})\} + \log\{q(\eta_i^{cur}|y_i)\} \\
&\quad - \log\{\pi(\boldsymbol{\eta}^{cur}|\boldsymbol{y})\} - \log\{q(\eta_i^{prop}|y_i)\} \\
&= \Delta\pi - \Delta q, \qquad (4.1.8)
\end{aligned}$$

where $\Delta\pi = \log\{\pi(\boldsymbol{\eta}^{prop}|\boldsymbol{y})\} - \log\{\pi(\boldsymbol{\eta}^{cur}|\boldsymbol{y})\}$ and $\Delta q = \log\{q(\eta_i^{prop}|y_i)\} - \log\{q(\eta_i^{cur}|y_i)\}$. Since $q(\eta_i|y_i)$ is a univariate t_{10} distribution the computational cost of evaluating Δq is

$\mathcal{O}(1)$. However, the target distribution is d dimensional and since it includes the calculation of the quadratic form in the normal prior of $\boldsymbol{\eta}$, the computational cost of evaluating $\Delta\pi$ is $\mathcal{O}(d^2)$. In what follows we show how this can be reduced to be $\mathcal{O}(d)$

Decomposing further the target distribution in terms of log likelihood and prior distribution of $\boldsymbol{\eta}$ we obtain $\Delta\pi = \Delta L + \Delta p$. Here,

$$\Delta L = \log\{L(\eta_i^{prop}; y_i)\} - \log\{L(\eta_i^{cur}; y_i)\} \tag{4.1.9}$$

the computational cost of which is $\mathcal{O}(1)$, and

$$\begin{aligned} \Delta p &= \log\{p(\boldsymbol{\eta}^{prop})\} - \log\{p(\boldsymbol{\eta}^{cur})\} \\ &= -\frac{1}{2}\left[(\boldsymbol{\eta}^{prop} - \boldsymbol{\mu}_\eta)' \boldsymbol{V}_\eta^{-1} (\boldsymbol{\eta}^{prop} - \boldsymbol{\mu}_\eta) - (\boldsymbol{\eta}^{cur} - \boldsymbol{\mu}_\eta)' \boldsymbol{V}_\eta^{-1} (\boldsymbol{\eta}^{cur} - \boldsymbol{\mu}_\eta)\right], \end{aligned} \tag{4.1.10}$$

with computational cost $\mathcal{O}(d^2)$, provided that the inverse of the covariance matrix is calculated in advance. Note that for the update of component i, the vectors $\boldsymbol{\eta}^{prop}$ and $\boldsymbol{\eta}^{cur}$ differ only at the i-th component that is updated. Therefore we can set $\boldsymbol{\eta}^{prop} = \boldsymbol{\eta}^{cur} + c\boldsymbol{e}_i$, where $\boldsymbol{e}_i$ is the $(d \times 1)$ vector which is 1 at the $i-$th component and zero everywhere else. After some algebra we derive that

$$\Delta p = c(\boldsymbol{\mu}_\eta - \boldsymbol{\eta}^{cur})' \left[\boldsymbol{V}_\eta^{-1}\right]_{,i} - \frac{c^2}{2}\left[\boldsymbol{V}_\eta^{-1}\right]_{ii}, \tag{4.1.11}$$

with the subscripts $_{,i}$ and $_{ii}$ denoting the i-th column and (i,i)-th element of the matrix respectively. In this way the computational cost of evaluating Δp and, equivalently, of

evaluating the acceptance probability is now $\mathcal{O}(d)$ instead of $\mathcal{O}(d^2)$. Hence, we need only calculate $\log\{p(\boldsymbol{\eta}^{cur})\}$ once, before the first iteration of the algorithm, and then, if the proposed value for η_i is accepted, we update the prior by setting $p(\boldsymbol{\eta}^{cur}) = p(\boldsymbol{\eta}^{cur}) + \Delta p$. Given that we must perform this operation on each of d components the computational cost of each iteration, j, is now $\mathcal{O}(d^2)$ instead of $\mathcal{O}(d^3)$.

4.2 Principal components conditioning

The proposal distribution constructed in the previous section ignores any correlation structure of the process $\boldsymbol{S}$ as it is based on the marginal posterior distribution of each S_i. For that reason, it is expected that in cases where the posterior correlation is strong the performance of algorithm U-MHIS would be poor and result in low acceptance rates and effective sample sizes.

We can obtain a reasonably accurate normal, or Student's-t, approximation to the true Gibbs sampler proposal, $\pi(s_i|\boldsymbol{s}_{-i}, \boldsymbol{y})$, using, for example, the Laplace approximation and we would expect this to result in high acceptance rates. However, for a given correlation matrix $\boldsymbol{R}$, calculation of the prior expectation and variance of S_i involves inverting $\boldsymbol{R}_{[-i,-i]}$, i.e., the matrix $\boldsymbol{R}$ with the $i-$th row and column removed. A total of d such $\mathcal{O}(d^3)$ calculations would make the algorithm $\mathcal{O}(d^4)$. Furthermore, the Gibbs sampler is known to mix poorly when the target distribution is characterised by strong correlations, even in the two dimensional, case as described in Gilks & Roberts (1996).

In the following, we construct an approximation to $\pi(s_i|\boldsymbol{s}_{-i}, \boldsymbol{y})$ and use it in an MH scheme. However, instead of conditioning on the $d-1$ components of $\boldsymbol{S}$ we condition on the few most important principal components of $\boldsymbol{s}$; this allows reduction of the computational cost. Since these principal components account for much of the correlation

structure between the s_i's we aim to get approximately the same information as if conditioning on $\boldsymbol{s}_{-i}$. Furthermore, the fact of conditioning on these few principal components allows us to improve the mixing of the whole algorithm by performing an additional update on the principal components themselves.

In Section 4.2.1 we describe how the proposal distribution is constructed, assuming knowledge of the optimal number of principal components to be used, and describe the main algorithm. This algorithm it is then extended in Section 4.2.2 in order to improve the mixing of the principal components through a single block update on them. Finally, issues related to the number of principal components and how this can be chosen are outlined in Section 4.2.3.

4.2.1 A single component MH algorithm through principal components conditioning

In the following, we work with the non-centred parametrisation of the latent process. More explicitly, we remove the prior mean, $\boldsymbol{\mu}_\eta = \boldsymbol{F}\boldsymbol{\beta}$, of the process $\boldsymbol{\eta}$ and work with $\boldsymbol{S}$. In that way, the latent process $\boldsymbol{S}$ and the mean parameter $\boldsymbol{\beta}$ are a priori independent. Moreover, we will be interested in calculating the principal components of the latent process. To do so, the mean is subtracted in order to center the process around the origin of the principal axes. Finally, when needed the i-th element of the d-dimensional vector $\boldsymbol{\mu}_\eta$ will be denoted by μ_{η_i} (or μ_η in cases where the subscript i is dropped in order to simplify notation).

Consider the spectral decomposition of the prior correlation matrix of $\boldsymbol{S}$,

$$\boldsymbol{R} = \boldsymbol{L}\boldsymbol{\Lambda}\boldsymbol{L}^{'}, \tag{4.2.1}$$

where $\boldsymbol{L}$ is an orthogonal $(d \times d)$ matrix whose columns are the eigenvectors of $\boldsymbol{R}$, and $\boldsymbol{\Lambda}$ is a diagonal $(d \times d)$ matrix with the ordered eigenvalues, $\lambda_1 > \lambda_2 ... > \lambda_d > 0$, of $\boldsymbol{R}$ corresponding to the eigenvectors in the columns of $\boldsymbol{L}$. The principal components, say $\boldsymbol{P}$, of $\boldsymbol{S}$ are then given by,

$$\boldsymbol{P} = \boldsymbol{L}^{'}\boldsymbol{S}, \tag{4.2.2}$$

where $\boldsymbol{P}$ is a $(d \times 1)$ column vector. Since $\boldsymbol{S} \sim \text{MVN}\left(\boldsymbol{0}, \sigma^2\boldsymbol{R}\right)$ then $\boldsymbol{P} \sim \text{MVN}\left(\boldsymbol{0}, \sigma^2\boldsymbol{\Lambda}\right)$. Now let $\tilde{\boldsymbol{L}}$ be the $(d \times k)$ matrix consisting of the first k columns of $\boldsymbol{L}$ and $\tilde{\boldsymbol{\Lambda}}$ the $(k \times k)$ diagonal matrix with the corresponding k eigenvalues on its diagonal. The first k principal components, $\tilde{\boldsymbol{P}}$, of $\boldsymbol{S}$ can therefore be obtained through,

$$\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{L}}^{'}\boldsymbol{S}. \tag{4.2.3}$$

The dimension of $\tilde{\boldsymbol{P}}$ is now $(k \times 1)$ and computational complexity of the calculation in equation (4.2.3) is $\mathcal{O}(kd)$. The joint prior distribution of $\boldsymbol{S}, \tilde{\boldsymbol{P}}$ is a multivariate normal with mean and covariance matrix as shown below,

$$\begin{bmatrix} \boldsymbol{S} \\ \tilde{\boldsymbol{P}} \end{bmatrix} \sim \text{MVN}\left(\begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{0} \end{bmatrix}, \sigma^2 \begin{bmatrix} \boldsymbol{R} & \tilde{\boldsymbol{L}}\tilde{\boldsymbol{\Lambda}} \\ \tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{L}}^{'} & \tilde{\boldsymbol{\Lambda}} \end{bmatrix}\right).$$

We can now consider the prior distribution of $\boldsymbol{S}|\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{p}}$. Using standard properties of the multivariate normal distribution we derive that,

$$\tilde{\boldsymbol{m}} := \mathbb{E}\left[\boldsymbol{S}|\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{p}}\right] = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{p}} \tag{4.2.4}$$

and

$$\sigma^2 \tilde{\boldsymbol{R}} := \mathbb{V}\text{ar}\left[\boldsymbol{S}|\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{p}}\right] = \sigma^2\left(\boldsymbol{R} - \tilde{\boldsymbol{L}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{L}}'\right). \tag{4.2.5}$$

Therefore,

$$\boldsymbol{S}|\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{p}} \sim \text{MVN}\left(\tilde{\boldsymbol{m}}, \sigma^2\tilde{\boldsymbol{R}}\right), \tag{4.2.6}$$

and the marginal posterior distribution of $s_i|\tilde{\boldsymbol{p}}$ is,

$$\tilde{\pi}\left(s_i|\tilde{\boldsymbol{p}}, y_i\right) \propto f\left(s_i; \tilde{m}_i(\tilde{\boldsymbol{p}}), \sigma^2\tilde{\boldsymbol{R}}_{ii}\right) L\left(e^{\mu_{\eta_i}+s_i}; y_i\right), \tag{4.2.7}$$

where μ_{η_i} is the i-th component of the prior mean $\boldsymbol{\mu}_\eta$, $f\left(\tilde{s}; \tilde{m}, \tilde{r}\right)$ denotes the density of a univariate normal$(\tilde{m}, \tilde{r})$ random variable $\tilde{S}$, L denotes the Poisson likelihood of the observations; $\tilde{m}_i$ is the $i-$th component of $\tilde{\boldsymbol{m}}$ and we make explicit that it is a function of the principal components. We work in exactly the same way as we did in Section 4.1 to obtain a Student's$-t$ approximation, $q_i(s_i|\tilde{\boldsymbol{p}})$, to $\tilde{\pi}\left(s_i|\tilde{\boldsymbol{p}}, y_i\right)$ and use this as a proposal in our MCMC scheme. In order to define the location and scale parameters of the proposal we follow the same arguments as in Section 4.1.1 and evaluate the mode and curvature at the mode of $\log\left\{\tilde{\pi}\left(s_i|\tilde{\boldsymbol{p}}, y_i\right)\right\}$. The only differences in the calculation procedure are that now instead of σ^2 we use $\sigma^2\tilde{R}_{ii}$ and we set $\mu^* = \tilde{m}_i + \sigma^2\tilde{R}_{ii}y_i$. Hence, equation (4.1.3) becomes,

$$\hat{s} = \mu^* - \sigma^2\tilde{R}_{ii}e^{\hat{s}+\mu_{\eta_i}}. \tag{4.2.8}$$

Each mean $\tilde{m}_i$ is calculated by only making use of the $i-$th coordinate of the k eigenvectors through,

$$\tilde{m}_i = \tilde{\boldsymbol{L}}_{[i,\ ]}\tilde{\boldsymbol{p}} \tag{4.2.9}$$

with computational cost $\mathcal{O}(k)$. In our MCMC scheme each individual update of s_i also

causes an update to all $\tilde{\boldsymbol{p}}$. Let $\boldsymbol{s}^{(cur)}$ and $\boldsymbol{s}^{(prop)}$ be the current and proposed vectors of $\boldsymbol{s}$ respectively differing only at the $i-$th component. Let also $\tilde{\boldsymbol{p}}^{(cur)}$ and $\tilde{\boldsymbol{p}}^{(prop)}$ be the corresponding principal components. Since at each iteration only one component of the vector $\boldsymbol{s}$ gets updated then $\tilde{\boldsymbol{p}}^{(prop)}$ is,

$$\tilde{\boldsymbol{p}}^{(prop)} = \tilde{\boldsymbol{p}}^{(cur)} + \tilde{\boldsymbol{L}}^{'}_{[\ ,i]} \left(s_i^{(prop)} - s_i^{(cur)} \right), \tag{4.2.10}$$

with computational complexity $\mathcal{O}(k)$. Let J denote the total number of iterations required, then our algorithm reads,

- Set initial values $\boldsymbol{s}^{(cur)}$ and define k.
- Obtain $\boldsymbol{L}$ from $\boldsymbol{R}$ via (4.2.1) and hence $\tilde{\boldsymbol{L}}$ and the diagonal of $\tilde{\boldsymbol{R}}$.
- Obtain $\tilde{\boldsymbol{p}}^{(cur)}$ from $\boldsymbol{s}^{(cur)}$ via (4.2.3).
- For j in $1:J$
  - For i in $1:d$
    - Obtain $\tilde{m}_i^{(cur)}$ from $\tilde{\boldsymbol{p}}^{(cur)}$ via (4.2.9).
    - Propose $s_i^{(prop)}$ from $q_i(s_i^{(prop)}|\tilde{\boldsymbol{p}}^{(cur)})$.
    - Obtain $\tilde{\boldsymbol{p}}^{(prop)}$ from $s_i^{(prop)}$ via (4.2.10).
    - Obtain $\tilde{m}_i^{(prop)}$ from $\tilde{\boldsymbol{p}}^{(prop)}$ via (4.2.9).
    - Set $\boldsymbol{s}^{(prop)} = \boldsymbol{s}^{(cur)} + \left(s_i^{(prop)} - s_i^{(cur)} \right) \boldsymbol{e}_i$.
    - Accept $\boldsymbol{s}^{(prop)}$ and hence $\tilde{\boldsymbol{p}}^{(prop)}$ with probability,

$$\alpha\left(\boldsymbol{s}^{(cur)}, \boldsymbol{s}^{(prop)}\right) := \min\left(1, \frac{\pi\left(\boldsymbol{s}^{(prop)}|\boldsymbol{y}\right) q_i\left(s_i^{(cur)}|\tilde{\boldsymbol{p}}^{(prop)}\right)}{\pi\left(\boldsymbol{s}^{(cur)}|\boldsymbol{y}\right) q_i\left(s_i^{(prop)}|\tilde{\boldsymbol{p}}^{(cur)}\right)}\right). \tag{4.2.11}$$

The acceptance probability in (4.2.11) is evaluated as outlined in Section (4.1.1) and its computational cost is $\mathcal{O}(d)$. Therefore, the overall computational cost for a single iteration, i.e. $I = 1$, through all d components remains $\mathcal{O}(d^2)$. We now also have the extra cost induced by the spectral decomposition of $\boldsymbol{R}$, which is $\mathcal{O}(d^3)$, but this need only be computed once at the beginning of the algorithm.

Each individual update of the proposed algorithm satisfies detailed balance with respect to $\pi(\boldsymbol{s}|\boldsymbol{y})$. In order to see that, we view the principal components as a function of $\boldsymbol{s}$, $\tilde{\boldsymbol{p}}(\boldsymbol{s})$, and therefore consider the proposal in terms of $\boldsymbol{s}$. In the following, when we write $\tilde{\boldsymbol{p}^c}$, $\tilde{\boldsymbol{p}^p}$ we consider $\tilde{\boldsymbol{p}}(\boldsymbol{s}^{(cur)})$ and $\tilde{\boldsymbol{p}}(\boldsymbol{s}^{(prop)})$ respectively. Let $\delta(x)$ be the Dirac delta function. Hence for the update of the $i-$th component,

$$\begin{aligned}
\pi\left(\boldsymbol{s}^c|\boldsymbol{y}\right) q\left(\boldsymbol{s}^p|\boldsymbol{s}^c\right) \alpha\left(\boldsymbol{s}^c, \boldsymbol{s}^p\right) &= \min\left\{\pi\left(\boldsymbol{s}^c|\boldsymbol{y}\right) q_i\left(s_i^p|\tilde{\boldsymbol{p}}^c\right), \pi\left(\boldsymbol{s}^p|\boldsymbol{y}\right) q_i\left(s_i^c|\tilde{\boldsymbol{p}}^p\right)\right\} \\
&\quad \times \prod_{j=1, j\neq i}^{d} \delta\left(s_j^p - s_j^c\right) \\
&= \pi\left(\boldsymbol{s}^p|\boldsymbol{y}\right) q_i\left(s_i^c|\tilde{\boldsymbol{p}}^p\right) \min\left\{1, \frac{\pi\left(\boldsymbol{s}^c|\boldsymbol{y}\right) q_i\left(s_i^p|\tilde{\boldsymbol{p}}^c\right)}{\pi\left(\boldsymbol{s}^p|\boldsymbol{y}\right) q_i\left(s_i^c|\tilde{\boldsymbol{p}}^p\right)}\right\} \\
&\quad \times \prod_{j=1, j\neq i}^{d} \delta\left(s_j^p - s_j^c\right) \\
&= \pi\left(\boldsymbol{s}^p|\boldsymbol{y}\right) q\left(\boldsymbol{s}^c|\boldsymbol{s}^p\right) \alpha\left(\boldsymbol{s}^p, \boldsymbol{s}^c\right).
\end{aligned}$$

We will refer to this algorithm as U-PC.

4.2.2 Improved mixing of $\tilde{p}$ through a single block update

The mixing of typical Gibbs samplers is known to be poor when there is strong correlation between the components being updated even in two dimensional targets (see Gilks & Roberts (1996), Hills & Smith (1992)). High dimensionality of the target amplifies

this problem rendering the exploration of the target by such schemes potentially very inefficient. In our MCMC scheme the mixing of the whole algorithm mainly depends on the mixing of $\tilde{\boldsymbol{p}}$ which is of low dimension, i.e., k. Therefore, we can improve the mixing of the algorithm by performing an additional update on these k principal components at each iteration j of the algorithm.

Single block RWM update

At a single iteration j, once all d components of the process $\boldsymbol{S}$ have been updated we perform an additional joint RWM step on $\tilde{\boldsymbol{p}}$ with an adaptive tuning parameter. We use the adaptive RWM proposed by Fearnhead et al. (2014) with the only difference being that we keep the covariance matrix of the proposal fixed and only adjust the tuning parameter.

To find an appropriate covariance matrix for our proposal we use the variance matrix, $\tilde{\boldsymbol{\Sigma}}$, used by Christensen et al. (2006) to standardise the process $\boldsymbol{S}$, see Section 2.2.7. More explicitly, let $\boldsymbol{s}^* = \log(\boldsymbol{y}) - \boldsymbol{\mu}_\eta$ be the mode of the Poisson likelihood and consider the approximation,

$$\begin{aligned} \tilde{\boldsymbol{\Sigma}} &= \left(-\frac{\partial^2}{\partial \boldsymbol{s}^2} \log \pi \left(\boldsymbol{s} | \boldsymbol{y} \right) \Big|_{\boldsymbol{s}=\boldsymbol{s}^*} \right)^{-1} \\ &= -\left(\operatorname{diag}(\boldsymbol{y}) + \left(\sigma^2 \boldsymbol{R} \right)^{-1} \right)^{-1}, \end{aligned}$$

for the posterior covariance matrix of $\boldsymbol{S}$, where the term $\log(\boldsymbol{y})$ comes from the second derivative of the Poisson log likelihood and the term $\left(\sigma^2 \boldsymbol{R}\right)^{-1}$ comes from the second derivative of the log prior of $\boldsymbol{S}$. Then the posterior covariance of $\tilde{\boldsymbol{P}}$ can be approximated

through, $\boldsymbol{\Sigma}_0 = \tilde{\boldsymbol{L}}^{'}\tilde{\boldsymbol{\Sigma}}\tilde{\boldsymbol{L}}$ and our proposal for the $j-$th iteration reads,

$$\tilde{\boldsymbol{p}}^{(prop)} \sim \begin{cases} \text{MVN}\left(\tilde{\boldsymbol{p}}^{(cur)}, \boldsymbol{\Sigma}_0\right) & \text{w.p} \quad 0.05, \\ \text{MVN}\left(\tilde{\boldsymbol{p}}^{(cur)}, t_j^2\boldsymbol{\Sigma}_0\right) & \text{w.p} \quad 0.95. \end{cases} \tag{4.2.12}$$

In turn, when the adaptive proposal has been used, the adaptive tuning parameter, t_j, is updated according to the following rule,

$$t_{j+1} = \begin{cases} t_j(1 + c_a\delta/n) & \text{if accepted} \quad \tilde{\boldsymbol{p}}^{(prop)}, \\ t_j(1 - c_r\delta/n) & \text{if rejected} \quad \tilde{\boldsymbol{p}}^{(prop)}, \end{cases} \tag{4.2.13}$$

where the adaptation parameter δ is set equal to 0.5, $c_a = 2.3, c_r = 1$, and n is the number of adaptive steps so far. Then, the RWM part of the algorithm is as follows,

- – Propose, $\tilde{\boldsymbol{p}}^{(prop)}$ according to (4.2.12) and adjust t_j according to (4.2.13).
  – Set, $\boldsymbol{s}^{(prop)} = \boldsymbol{s}^{(cur)} + \tilde{\boldsymbol{L}}\left(\tilde{\boldsymbol{p}}^{(prop)} - \tilde{\boldsymbol{p}}^{(cur)}\right)$.
  – Accept with probability,

$$\alpha\left(\boldsymbol{s}^{(cur)}, \boldsymbol{s}^{(prop)}\right) = \min\left(1, \frac{\pi\left(\boldsymbol{s}^{(prop)}|\boldsymbol{y}\right)}{\pi\left(\boldsymbol{s}^{(cur)}|\boldsymbol{y}\right)}\right). \tag{4.2.14}$$

In this part of the algorithm, the computational cost of obtaining $\boldsymbol{s}^{(prop)}$ is $\mathcal{O}(kd)$. Moreover, the calculation of the acceptance probability is $\mathcal{O}(d + k^2)$. This comes from the calculation of the likelihood which is $\mathcal{O}(d)$ and that of the prior which can be reduced to $\mathcal{O}(k^2)$ since we can replace $\boldsymbol{s}^T\boldsymbol{R}^{-1}\boldsymbol{s}$ with $\tilde{\boldsymbol{p}}^T\boldsymbol{\Lambda}^{-1}\tilde{\boldsymbol{p}}$. Hence, since $k < d$ the overall cost of the RWM step is at most $\mathcal{O}(kd)$. When the full algorithm, including the RWM step, is implemented we will refer to it as PC-RWM.

Single block MALA update

Once all d components of the latent process have been updated, an alternative to the RWM for updating the block of k components of $\tilde{\boldsymbol{p}}$ is the MALA. We use the same variance matrix $\boldsymbol{\Sigma}_0$ as above and propose $\tilde{\boldsymbol{p}}^{(prop)}$ from,

$$q\left(\tilde{\boldsymbol{p}}^{(prop)}|\tilde{\boldsymbol{p}}^{(cur)}\right) \equiv \text{MVN}\left(\tilde{\boldsymbol{p}}^{(cur)} + \frac{t}{2}\boldsymbol{\Sigma}_0\nabla_{\tilde{\boldsymbol{p}}}\log\pi(\boldsymbol{p}|\boldsymbol{y}), t\boldsymbol{\Sigma}_0\right), \tag{4.2.15}$$

where t is a fixed tuning parameter and the gradient is given by,

$$\nabla_{\tilde{\boldsymbol{p}}}\log\pi(\boldsymbol{p}|\boldsymbol{y}) = \tilde{\boldsymbol{L}}^{'}\left(\boldsymbol{y} - e^{\boldsymbol{\eta}} - \frac{1}{\sigma^2}\boldsymbol{R}^{-1}(\boldsymbol{\eta} - \boldsymbol{\mu}_\eta)\right). \tag{4.2.16}$$

Then, the MALA part of the algorithm is as follows,

- – Propose, $\tilde{\boldsymbol{p}}^{(prop)}$ according to (4.2.16).
  – Set, $\boldsymbol{s}^{(prop)} = \boldsymbol{s}^{(cur)} + \tilde{\boldsymbol{L}}\left(\tilde{\boldsymbol{p}}^{(prop)} - \tilde{\boldsymbol{p}}^{(cur)}\right)$.
  – Accept with probability,

$$\alpha\left(\boldsymbol{s}^{(cur)}, \boldsymbol{s}^{(prop)}\right) = \min\left(1, \frac{\pi\left(\boldsymbol{s}^{(prop)}|\boldsymbol{y}\right) q\left(\tilde{\boldsymbol{p}}^{(cur)}|\tilde{\boldsymbol{p}}^{(prop)}\right)}{\pi\left(\boldsymbol{s}^{(cur)}|\boldsymbol{y}\right) q\left(\tilde{\boldsymbol{p}}^{(prop)}|\tilde{\boldsymbol{p}}^{(cur)}\right)}\right). \tag{4.2.17}$$

As in the RWM update the overall computational cost of obtaining $\boldsymbol{s}^{(prop)}$ and calculating the ratio of the target in the acceptance probability is $\mathcal{O}(kd)$. The overall computational cost of the MALA update though will be determined by the cost of calculating the proposal. This mainly comes from calculating $\boldsymbol{R}^{-1}(\boldsymbol{\eta} - \boldsymbol{\mu}_\eta)$ in 4.2.16 which appears to be $\mathcal{O}(d^2)$. It can, however, be reduced to $\mathcal{O}(kd)$ by noting that

$$\boldsymbol{R}^{-1}(\boldsymbol{\eta} - \boldsymbol{\mu}) = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{L}}^{'}\boldsymbol{s} = \tilde{\boldsymbol{L}}\tilde{\boldsymbol{\Lambda}}^{-1}\tilde{\boldsymbol{p}}.$$

4.2.3 Choice of k

In what discussed so far we have assumed knowledge of the exact number k of principal components on which we should condition in order to construct efficient MCMC schemes. In the following, we outline the importance of specifying an appropriate k correctly and the effect of such a choice on the computational cost of the algorithm and the accuracy of the proposal distribution.

Computational cost

The mixing time of the RWM on k principal components is $\mathcal{O}(k)$ (Roberts & Rosenthal (2001)) and for the MALA it is $\mathcal{O}(k^{1/3})$, suggesting that k should be chosen as small as possible. Moreover, as illustrated in the previous Section a large value of k would increase the computational cost of the algorithm. In particular, the cost of the d individual updates, i.e., of a complete j iteration, is $\mathcal{O}(d^2)$ whereas that of the block update is $\mathcal{O}(kd)$. This suggests that, if k is $\mathcal{O}(d)$ then the CPU time needed for updating $\tilde{\boldsymbol{p}}$ would be of the same order as that of the individual updates. Equivalently choosing k to be $o(d)$ would render the cost of the RWM/MALA negligible relative to that of the individual updates.

Approximation of $\pi\left(s_i|\boldsymbol{s}_{-i}\right)$

Although the choice of k influences the computational cost of the algorithm and the mixing efficiency of the principal components another important aspect is the accuracy of our approximation, $q_i(s_i|\tilde{\boldsymbol{p}})$ to the true conditional of S_i given all the other components. Therefore, we would like to choose k such that $q_i\left(s_i|\tilde{\boldsymbol{p}}\right) \equiv \pi\left(s_i|\boldsymbol{s}_{-i}\right)$, so that the acceptance probability is close to 1; the acceptance probability of a true Gibbs sampler. We make two approximations, the Student's-t approximation to the conditional posterior of

s_i and the fact that we condition on $\tilde{\boldsymbol{p}}$ rather than $\boldsymbol{s}_{-i}$. Results for U-MHIS show that the former is sufficiently accurate in one dimension so we investigate the latter.

Since the likelihood is the same in both cases we compare the prior distributions. Both are normal distributions and therefore fully characterised by their mean and variance. For that reason, we simply assess whether,

$$\mathbb{V}\text{ar}\left[S_i|\tilde{\boldsymbol{P}}\right] \approx \mathbb{V}\text{ar}\left[S_i|\boldsymbol{S}_{-i}\right] \tag{4.2.18}$$

and

$$\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right] \approx \mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] \tag{4.2.19}$$

for a given value of k. Firstly, we know that on average the two expectations will be equal since

$$\mathbb{E}_{\boldsymbol{S}_{-i}}\left[\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right]\right] = \mathbb{E}_{\tilde{\boldsymbol{P}}}\left[\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right] = \mathbb{E}\left[S_i\right].$$

However, clearly in general $\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] \neq \mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]$; for example consider the case where $k = 0$ and hence $\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right] = \mathbb{E}\left[S_i\right]$. Nonetheless, if we can find a value for k such that

$$v_i^{\mathbb{E}} = \mathbb{V}\text{ar}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] - \mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right),$$

is small and also the variances, of the true conditional distribution and of our approximation, are approximately equal, i.e., Equation 4.2.19 holds, then our proposal would well approximate the truth (see Appendix 4.5.1 for the exact form of $v_i^{\mathbb{E}}$). Except for small edge effects, we expect $v_i^{\mathbb{E}}$ to be similar for every i due to the regular structure of the grid so we examine the mean over i for each value of k. More explicitly, we look at

$$\bar{v}^{\mathbb{E}} = \frac{1}{d}\sum_{i=1}^{d} v_i^{\mathbb{E}}, \quad \bar{v}^{a} = \frac{\sigma^2}{d}\sum_{i=1}^{d} \tilde{R}_{ii}, \quad \bar{v}^{t} = \frac{1}{d}\sum_{i=1}^{d} \mathbb{V}\mathrm{ar}\left[S_i|\boldsymbol{s}_{-i}\right]$$

for all possible values of k within a given dimension. To summarise, v_i^t is the true prior variance of $S_i|\boldsymbol{S}_{-i}$ and v_i^a is our approximation to it, i.e., it is the variance of $S_i|\tilde{\boldsymbol{P}}$. Therefore, Given 4.2.18 and 4.2.19, our goal is to find a value for k which would minimise $\bar{v}^{\mathbb{E}}$ and at the same time achieve $\bar{v}^a \approx \bar{v}^t$

Since its value is irrelevant to our goal, we set $\sigma^2 = 1$ and evaluate the above diagnostics for three different scenarios of correlation, low, strong and very strong under the exponential correlation function on three different dimensions and three different values for ϕ, $\phi \in \{1, 10, 100\}$. The above diagnostics are shown in Figure 4.1.

The first thing to notice is that for the given correlation function and ϕ the pattern in the graphs is extremely stable with changes in dimension. This is a result of the regular structure of the grid on the square $\left\{1, 2, ..., \sqrt{d}\right\}^2$. Also, once conditioning on many close values, conditioning on more points further away would not provide much more information and therefore will not materially change the mean and variance of $S_i|\boldsymbol{S}_{-i}$.

As ϕ increases the correlation between the components becomes stronger and therefore $\bar{v}^t$ (black dashed line) decreases since the remaining $d-1$ components are highly informative about s_i. On the other hand, when the s_i's are almost independent conditioning on all $d-1$ components adds little information and $\bar{v}^t$ is a little less than the marginal variance of s_i.

As the number, k, of principal components on which we condition increases, $\bar{v}^a$ (grey solid line) decreases since increasing k provides more information about S_i conditional on knowing all the first k principal components. For $k = 0$ we actually condition on no

principal components and hence obtain the marginal variance of s_i. The stronger the correlation in $\boldsymbol{S}$ the sharper the initial drop in $\bar{v}^a$ is. This occurs because as the correlation increases the first principal component explains more of the variability. Irrespective of the value of ϕ, when $k = 0$ then $v_i^{\mathbb{E}} = \mathbb{V}\text{ar}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right]\right)$ (estimated by the black solid line) which is equal to $\mathbb{V}\text{ar}\left[S_i\right] - \mathbb{V}\text{ar}\left[S_i|\boldsymbol{S}_{-i}\right] = 1 - \mathbb{V}\text{ar}\left[S_i|\boldsymbol{S}_{-i}\right]$; the behaviour of $\mathbb{V}\text{ar}[S_i|\boldsymbol{S}_{-i}]$ has already been discussed. Similarly, as $k \to d$, $v_i^{\mathbb{E}} \to \mathbb{V}\text{ar}\left[S_i|\boldsymbol{S}_{-i}\right]$.

Proposition 4.2.1. *Irrespective of the value of ϕ and dimension, for $k = d$ then,*

$$v_i^{\mathbb{E}} = \mathbb{V}ar\left(S_i|\boldsymbol{S}_{-i}\right).$$

The proof of Proposition 4.2.1 can be found in Appendix 4.5.2.

Critically, with the exponential correlation function, both of the criteria that make our approximation accurate are met for small k. To be more specific, for the exponential correlation function, Figure 4.1, in all cases $\bar{v}^{\mathbb{E}}$ is minimised for values close to $k = d/4$ but it is relatively flat between $d/8$ and $3d/8$. Therefore we choose $k = d/8$ wich reduces the computational cost.

It is obvious that for the exponential correlation function there is a small range of values for k within which both the expectation and variance of our proposal is very close to the true and small changes in choice of k would have a negligible effect on our approximation and also $\bar{v}^{\mathbb{E}}$ is small relative to $\bar{v}^t$. In order to account for any, even small, inconsistency between the modes of the two distributions we might ideally choose the value of k at which $\bar{v}^a$ is slightly greater than the true $\bar{v}^t$. In that way, our proposal would be more dispersed that the true density.

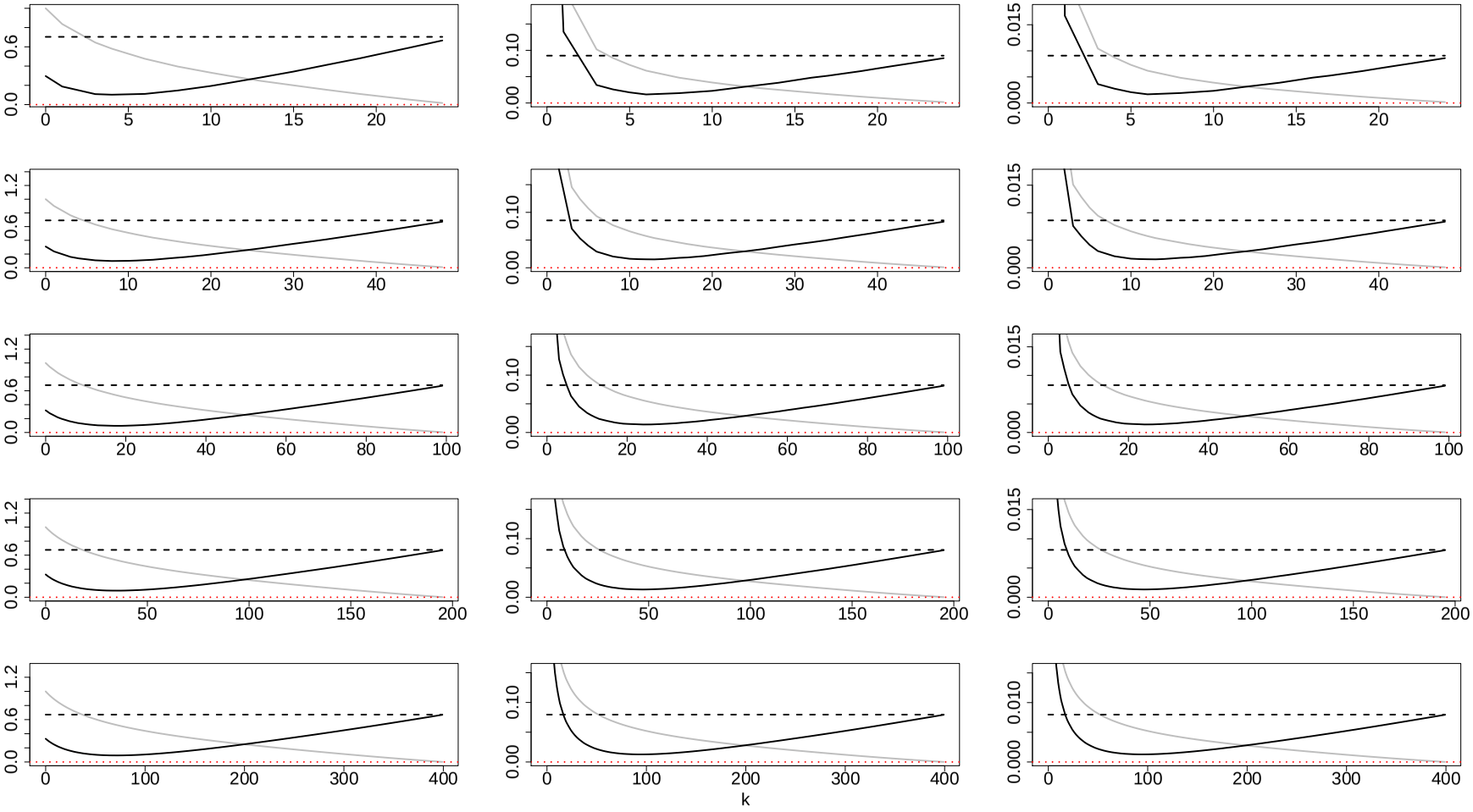

Figure 4.1: Plots of $\bar{v}^a$ (grey solid line), $\bar{v}^t$ (black dashed line) and $\bar{v}^{\mathbb{E}}$ (black solid line) against the number k of principal components. The prior correlation matrix $\boldsymbol{R}$ is constructed using the exponential correlation function. Top to Bottom: $d = \{25, 49, 100, 196, 400\}$. Left to Right: $\phi = 1$, $\phi = 10$, $\phi = 100$.

Approximation of $\pi\left(s_i|\boldsymbol{s}_{-i},\boldsymbol{y}\right)$

As k increases the diagonal terms in $\tilde{\boldsymbol{R}}$, (i.e., $\bar{v}^a$) decrease. Since this is the prior variance in our approximation, if it is very low then it could be so informative that the likelihood becomes less relevant. As the likelihood is of product form and therefore adds no further dependence, increasing k could actually increase the posterior correlation.

Additionally, the smaller $\tilde{R}_{ii}$ the smaller is the variance of our proposal for s_i and therefore the smaller the proposed changes for s_i. In cases where the U-PC algorithm is implemented there is no block update for the principal components and they are only updated indirectly through those small changes in each s_i. For that reason, increasing k could decrease the mixing of each principal component.

Analytical arguments to account for all the above factors would be too complex, so we conduct a simulation study to ascertain the actual efficiency of our algorithm, with or without the block update step of the principal components, $\tilde{\boldsymbol{p}}$, as a function of the number of principal components, k, over a variety of scenarios and dimensions.

All three algorithms, namely U-PC, PC-RWM and PC-MALA, were run for the same seven scenarios of parameter values that have also been used in the simulation studies on three different dimensions, $d \in \{25, 49, 100\}$ using the exponential correlation function for the construction of $\boldsymbol{R}$. For each scenario of parameter values and dimension each algorithm was implemented for a variety of values for $k \in [0, d-1]$ and the minimum ESS was recorded each time. The PC-RWM algorithm was tuned so that the adaptive part achieved acceptance rates between $32\% - 35\%$. This acceptance rate might be slightly larger than the optimal 0.234 but as illustrated in Roberts & Rosenthal (2001) the efficiency of a RWM is relatively stable for values of α between 0.15 and 0.5 In order to produce these diagnostics for the PC-MALA algorithm we implemented an adaptive

scheme for the tuning parameter similar to that constructed for the PC-RWM with the difference being that it was constructed so that the MALA step achieved acceptance rates close 59%.

In Figures 4.3-4.5 we display the results of this simulation study with each coloured line corresponding to a different scenario of parameter values (see Figure 4.2 for details) and each point on the line corresponds to the logarithm of the minimum ESS that was achieved for a specific value of k. The black dashed line corresponds to that value of k that was selected as optimal for the accuracy of our proposal based on Figure 4.1, i.e., $k = d/8$.

First of all, the observed pattern is similar in all three plots with the minimum ESS increasing up to a certain number of principal components and then gradually decreasing as we condition on more and more principal components. In all three algorithms the chosen value for k seems to be very close to that achieving the highest minimum ESS for all scenarios of parameter values and dimension validating our choice. Comparing across the three algorithms we see that as we move from U-PC to PC-RWM and then to PC-MALA the performance of the algorithm improves in terms of ESS. In particular, with the inclusion of the block update on $\tilde{\boldsymbol{p}}$ the improvement is obvious in all scenarios with the most important increase in ESS occurring for large values of k with the MALA update stabilising the minimum ESS at higher values than the RWM update.

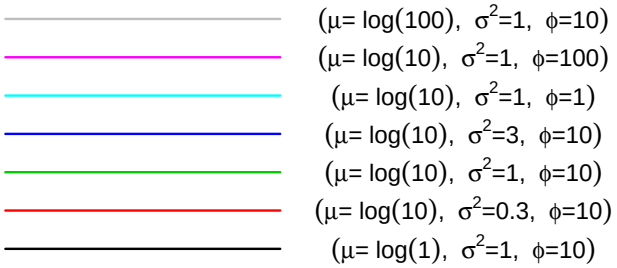

Figure 4.2: Colour configuration for Figures 4.3–4.5 and Figure 4.7. Each colour corresponds to a different scenario of parameter values.

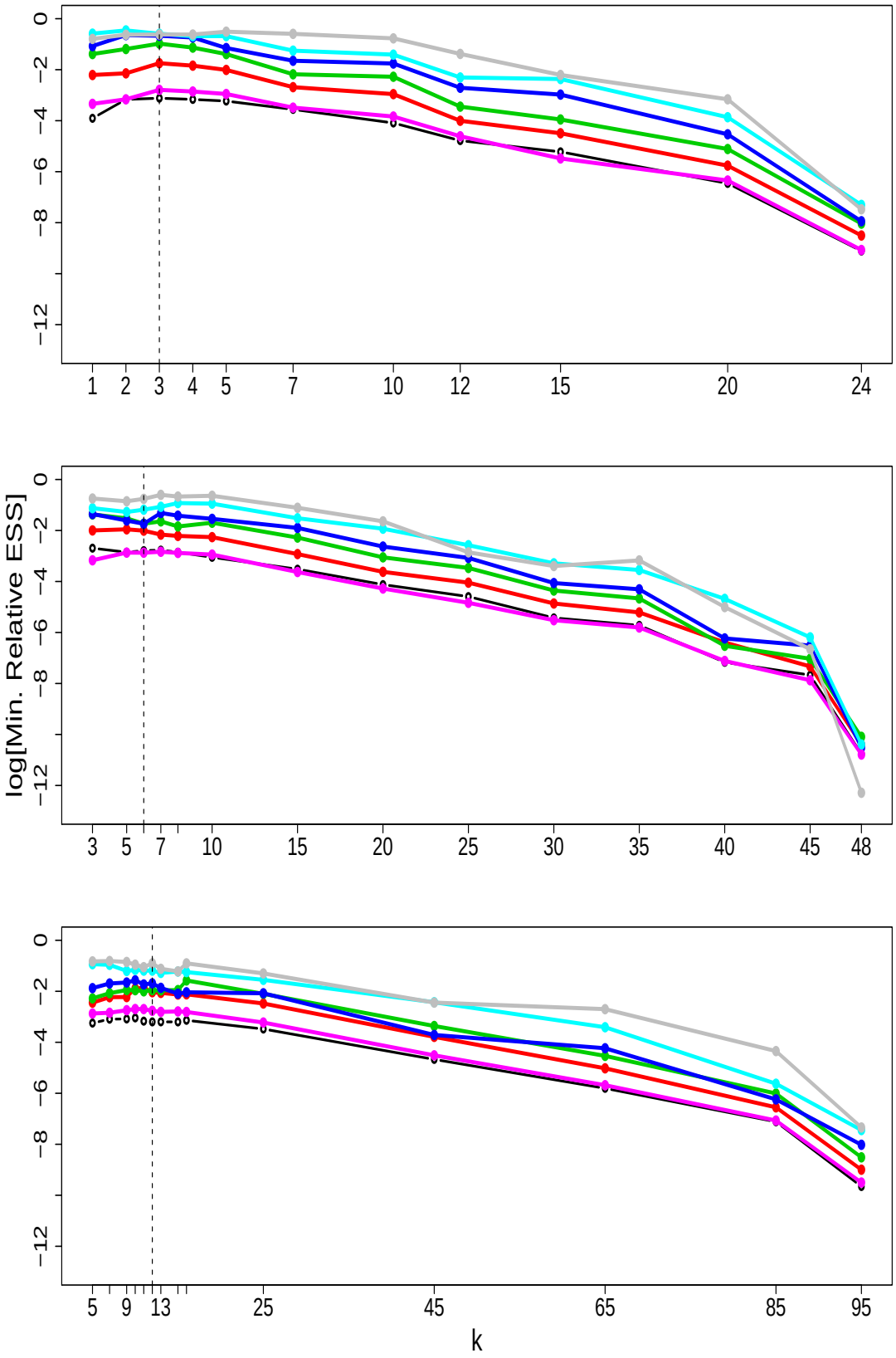

Figure 4.3: Algorithm U-PC. Logarithm of minimum relative ESS against different values of k. Top to bottom: Dimension, $d = \{25, 49, 100\}$. The correlation matrix $\boldsymbol{R}$ is constructed using the exponential correlation function.

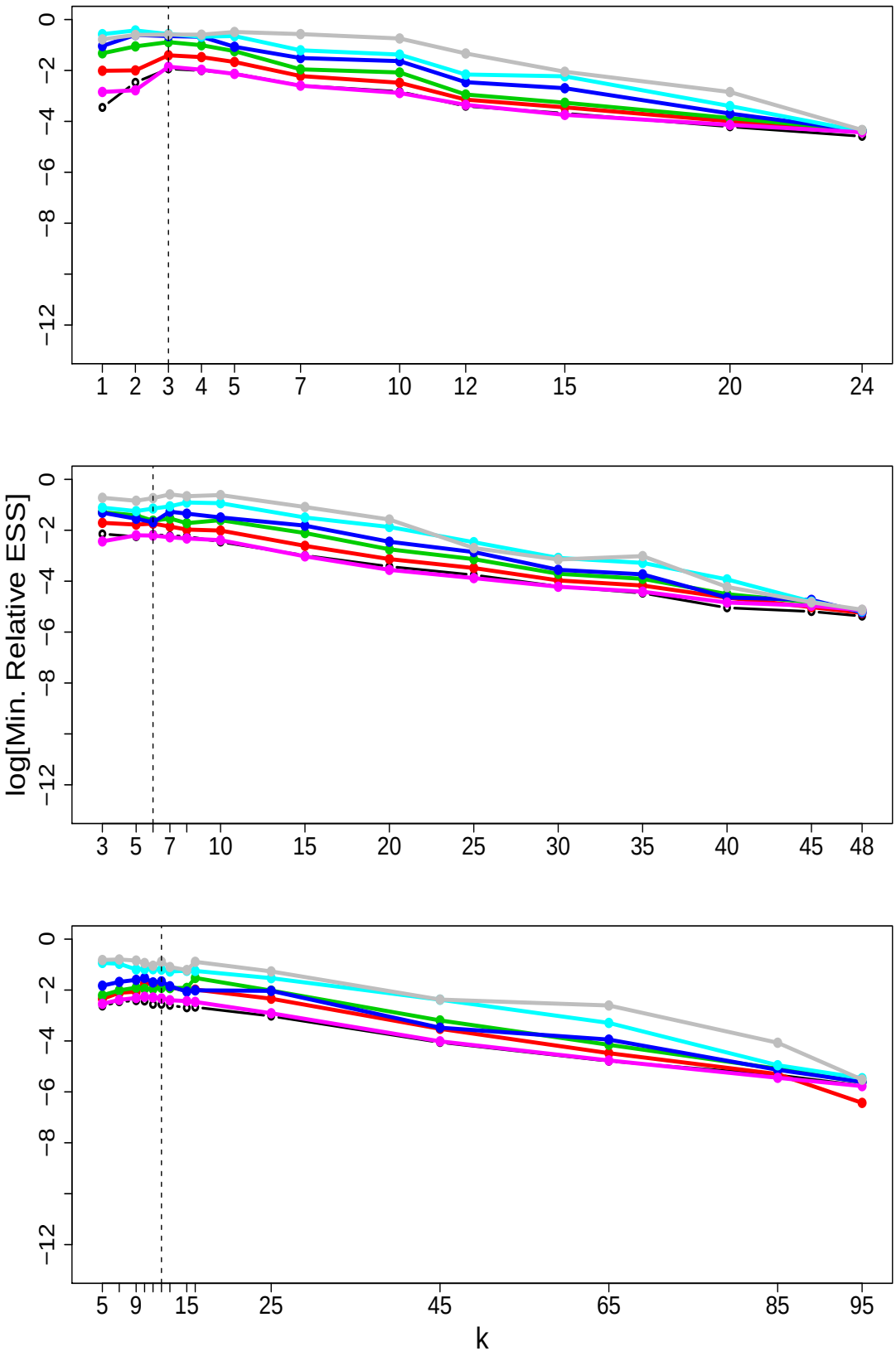

Figure 4.4: Algorithm PC-RWM. Logarithm of minimum relative ESS against different values of k. Top to bottom: Dimension, $d = \{25, 49, 100\}$. The correlation matrix $\boldsymbol{R}$ is constructed using the exponential correlation function.

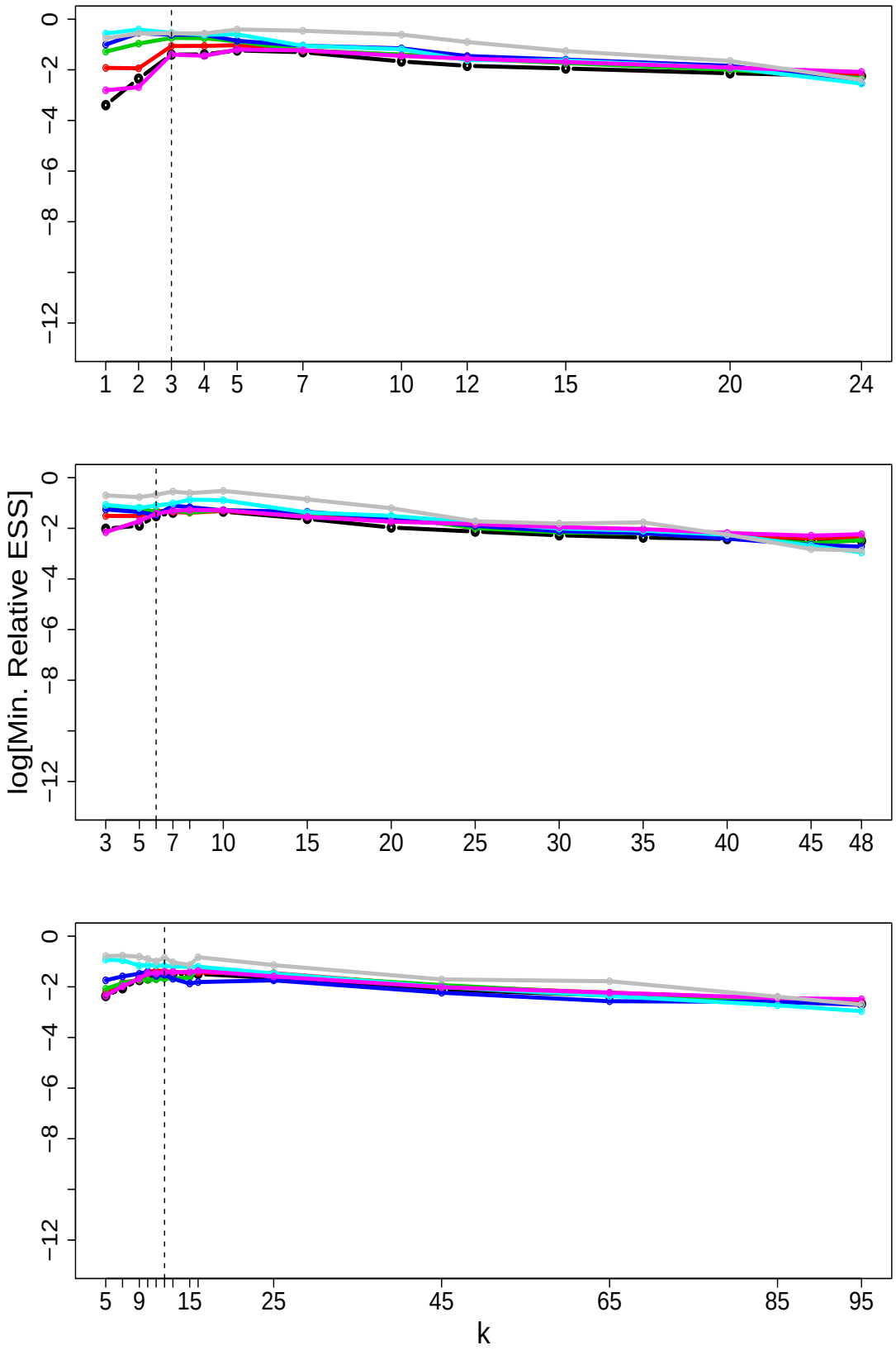

Figure 4.5: Algorithm PC-MALA. Logarithm of minimum relative ESS against different values of k. Top to bottom: Dimension, $d = \{25, 49, 100\}$. The correlation matrix $\boldsymbol{R}$ is constructed using the exponential correlation function.

4.3 Simulation study and results

In this section we assess the performance of the algorithms U-MHIS, U-PC, PC-RWM and PC-MALA. In Tables 4.3.1−4.3.3 we monitor minimum, median and maximum ESS, CPU time needed for 8×10^5 samples to be drawn, adjusted ESS and acceptance rates, α. Since now each η_i is accepted or rejected separately we obtain d different acceptance rates and we report the minimum, median and maximum values. The adaptive PC-RWM algorithm was constructed so that for the block update of $\tilde{\boldsymbol{p}}$ it achieved acceptance rate between $30\% - 35\%$. Finally, for all three algorithms, i.e., U-PC, PC-RWM and PC-MALA, k was chosen to be $d/8$. The PC-MALA algorithm was tuned so that the MALA update achieved acceptance rates between $57\% - 59\%$. All four algorithms and that of Christensen et al. (2006) were coded in C.

4.3.1 Simulation study

All five algorithms are implemented on the same seven scenarios of parameter values as those used in Section 3.3 in three different dimensions $d \in \{25, 49, 100\}$ using the same simulated datasets and initial values for the MCMC. Once more, the results are based on 8×10^5 samples drawn from the posterior distribution $\pi(\boldsymbol{\eta}|\boldsymbol{y})$ of a total of 10^6 samples. For a detailed description of the simulation study design see Section 3.3. Initial runs of the Christensen et al. (2006) algorithm were implemented for all scenarios for each dataset. A sample point from the posterior with marginal components always lying within the 0.01 and 0.99 quantiles of their distributions was chosen, by rejection sampling, and was used as starting value for each of the algorithms in our simulation study for that particular dataset. The algorithms were run for 10^6 iterations and the first 2×10^5 iterations were used as burn in. All results are based on 8×10^5 samples drawn from the posterior distribution of $\boldsymbol{\eta}|\boldsymbol{y}$.

4.3.2 Results

Tables 4.3.1-4.3.4 show the results from U-MHIS, U-PC, PC-RWM and PC-MALA respectively and Table 4.3.5 shows the results obtained from the algorithm of Christensen et al. (2006). Overall, both acceptance rates and effective sample sizes have improved compared to those of the algorithms presented in Chapter 3. Moreover, in contrast with the multivariate proposals of Chapter 3, increasing dimension of the latent process seems to have relatively little impact on the performance of the algorithms. All four algorithms exhibit the same pattern of performance over the different scenarios of parameter values. For that reason we outline the overall pattern of performance of the algorithms and remark on any differences where needed.

The performance of the algorithms seems to be mostly affected by the value of the mean $\boldsymbol{\mu}_\eta$ and the correlation parameter ϕ. The algorithms that employ the conditioning on the principal components appear to be less affected by changes in σ^2 especially as dimension increases.

The higher the prior mean $\boldsymbol{\mu}_\eta$, the more symmetric the Poisson distribution of the data is. Therefore, the normal approximation of the full conditional becomes more accurate resulting in the shape of our proposal being closer to the true shape of the posterior. Additionally, as the data become more informative the the likelihood dominates and the posterior correlation decreases resulting in better performance of our algorithms.

Increasing σ^2, increases the range of values for $\boldsymbol{\eta}$ giving rise to more η_i's with larger and smaller values. On the one hand, larger values of η_i result in a large mean for the Poisson likelihood and therefore our normal approximation becomes more accurate as explained above. On the other hand, small values of η_i correspond to a low mean for

the likelihood leading to a less accurate proposal. However, these are less correlated *a posteriori* since the prior becomes flat and the likelihood becomes more relevant resulting in lower posterior correlation.

Similarly, increasing ϕ implies higher posterior correlation having the same negative effect on all three algorithms. For instance, the U-MHIS proposal is most affected by such large values of ϕ since it doesn't account for any correlation and hence its performance benefits from low posterior correlations. Looking at the scenario with the highest correlation, $\phi = 100$, and comparing across algorithms the expected improvement in performance between U-MHIS and that of the remaining algorithms is obvious showing the positive effect of conditioning on $\tilde{\boldsymbol{p}}$. Additionally, incorporating the block update on $\tilde{\boldsymbol{p}}$ in the PC-RWM and PC-MALA does improve the mixing of the whole algorithm as their performance is even better. In the case of U-PC, PC-RWM and PC-MALA our proposal mimics that of a Gibbs sampler and increasing correlation hinders their performance. Although the conditioning on $\tilde{\boldsymbol{p}}$ is not as restrictive as full knowledge of all $\boldsymbol{s}_{-i}$, it still assumes their partial knowledge on the k most informative directions. Hence, high posterior correlation still has an effect on their performance.

The increased computational cost of U-PC algorithm compared to that of U-MHIS is illustrated in the CPU timings which are almost three times higher. On the other hand, the increase of CPU timings between U-PC and PC-RWM/PC-MALA is relatively small indicating the importance of choosing a low value for k so that the computational expense of updating $\tilde{\boldsymbol{p}}$ is comparable to that of the individual updates.

PC-MALA provides the best results in terms of acceptance rates and ESS in all cases followed by PC-RWM. Comparing the results of PC-MALA with those of Christensen et al. (2006) in Table 4.3.5 we see that PC-MALA always performs better in terms of

minimum ESS irrespective of dimension. Although computationally more expensive than the other three algorithms, PC-MALA seems to also achieve better adjusted ESS than the algorithm of Christensen et al. (2006) for dimensions $d = \{49, 100\}$.

As far as convergence is concerned, we notice that in the case of U-PC algorithm there are many scenarios were convergence to the same distribution with the algorithm of Christensen et al. (2006) is rejected. More explicitly, there are 11 rejected scenarios out of a total of 63 MCMC implementations with p-values between $1\% - 2\%$. We conducted several investigations in order to justify such an outcome but we were not able to identify any clear cause. Some of our investigations are presented in the Appendix 4.5.3.

We choose the best performing algorithm, PC-MALA, and perform additional simulations for dimensions $d = \{196, 400\}$ to assess the stability of the results obtained so far in higher dimensions. Table 4.3.6 and Table 4.3.7 show the results obtained from PC-MALA and the algorithm of Christensen et al. (2006) on these dimensions. Due to storage considerations, for dimension $d = 400$ the chains were thinned by a factor of two resulting in final samples of size 4×10^5 and therefore the maximum possible value for the relative ESS displayed in Table 4.3.7 will be 0.5 rather than 1. The pattern remains the same with PC-MALA still achieving better ESS and adjusted ESS.

The results presented, indicate, first of all, that the diagnostic in Figure 4.1 is sufficient to provide a near optimal choice of k, as shown in Figures 4.3-4.5, and also, that given this choice of k our MCMC scheme can perform better than that suggested by Christensen et al. (2006). However, these arguments have so far been established based on the fact that the exponential correlation function is used. It seems natural to examine the applicability and efficiency of our methods under different correlation structures.

Effect of different correlation functions

In the following, we look at the Matérn correlation function with shape parameter $\kappa = 1.5$ which we will denote by $\rho_2(u/\phi_2)$. This correlation function is one time mean-square differentiable as opposed to the exponential correlation which is not mean-square differentiable. This implies that the former corresponds to a smoother latent process with relatively stronger dependence over short distances, i.e., the correlation decays slower near the origin. Denote the exponential correlation function by $\rho_1(u/\phi_1)$. In order for the two correlation functions, and the results presented, to be comparable we define ϕ_2 by minimising the absolute distance of the two correlation functions over an infinitely fine two-dimensional grid. More explicitly we set $\phi_1 = \{1, 10, 100\}$ as before and choose,

$$\phi_2 = \text{arg}_{\phi^*}\text{min} \int_0^\infty 2\pi u |\rho_1(u/\phi_1) - \rho_2(u/\phi^*)| du.$$

Figure 4.6 shows the diagnostic plots for the choice of k so that $q_i(s_i|\tilde{\boldsymbol{p}}) \equiv \pi(s_i|\boldsymbol{s}_{-i})$ (see Equation 4.2.18 and Equation 4.2.19). The behaviour of $\bar{v}^E$, $\bar{v}^t$ and $\bar{v}^a$ appears to be the same as in Figure 4.1. The plots, however, now indicate that we need to condition on more principal components than in the exponential correlation function in order for our criteria to be met. More explicitly, we see that for $\phi = \{10, 100\}$, k should be chosen to be $2d/7$ whereas for $\phi = 1$, we should choose $k = d/6$. We believe that this increase in the number of principal components has to do with the differentiability of the correlation function at the origin. As κ increases from 0.5 to 1.5 the correlation remains at very high levels for longer distances, therefore there are now more points on the grid that are strongly correlated. In Figure 4.7 we assess the performance of PC-MALA across different values of k. As is the case of the exponential correlation function our choice of k seems to be sensible since for higher values, the effective sample sizes either decrease

or at least do not appear to improve.

Based on the results in the simulation studies so far, which were on three replicates for each parameter scenario, we see that the effects of Monte Carlo variability and different simulated data sets on the results are relatively small. Each replicate run is, however, very demanding in terms of both computational time and storage space. Therefore, we now implement these two algorithms on only one dataset for each scenario of parameter values.

Tables 4.3.8 and 4.3.9 show the results from the new simulation study for dimensions $d = \{25, 49, 100\}$ and $d = \{196, 400\}$ respectively. As far as PC-MALA is concerned we see that the acceptance rates have slightly dropped but still stabilised between 50% and 70% regardless the dimension of the process. Moreover, as expected, conditioning on more principal components has increased the computational time. The ESSs have dropped with the median and maximum effective sample sizes being considerably affected, resulting, therefore in lower adjusted effective sample sizes. In dimensions $d = \{196, 400\}$ the minimum effective sample sizes are now about to 2 times lower than before giving rise to roughly 3 times lower adjusted effective sample sizes. We also note that changes on the parameter values do not seem to affect the performance of the algorithm as much as before. It is interesting that the most affected scenario is that with the highest mean, i.e., $\boldsymbol{\mu}_\eta = \log(100)$, where the achieved ESS has been reduced by a factor of 3 compared with $k = 0.5$ for dimensions $d = \{196, 400\}$ and by a factor of 4 for lower dimensions. On the other hand, the algorithm of Christensen et al. (2006) seems to be less affected by the different correlation structure. The minimum ESS are either on the same levels as before or slightly improved although the maximum ESS have decreased.

Although PC-MALA appears to be more affected by the change of correlation function

we see that in dimension $d = 400$ it still achieves better adjusted ESS than the algorithm of Christensen et al. (2006) while achieving similar performance in lower dimensions and always performing better when $\phi = \{1, 100\}$.

Table 4.3.1: Algorithm U-MHIS. Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = \{25, 49, 100\}$. Grey colour: K–S test does not support convergence.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$ $\overline{CPU} = 12$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.181, .193, .204)	(.311, .195, .331)	(.419, .256, .412)	(.634, .390, .541)	(.835, .712, .710)	(.198, .055, .085)	(.673, .676, .613)
Med α	(.386, .377, .389)	(.520, .496, .478)	(.671, .662, .616)	(.790, .787, .774)	(.903, .919, .874)	(.294, .279, .274)	(.882, .884, .887)
Max α	(.521, .440, .525)	(.575, .596, .596)	(.779, .809, .845)	(.924, .916, .946)	(.939, .953, .961)	(.376, .388, 0.349)	(.942, .941, .946)
Min. ESS	(.021, .020, .019)	(.085, .042, .093)	(.133, .112, .170)	(.311, .124, .221)	(.556, .382, .329)	(.029, .008, .012)	(.404, .401, .328)
Med. ESS	(.035, .034, .047)	(.155, .145, .134)	(.376, .377, .342)	(.611, .585, .536)	(.763, .800, .700)	(.036, .027, .026)	(.774, .754, .800)
Max. ESS	(.080, .045, .108)	(.197, .211, .207)	(.547, .550, .659)	(.866, .850, .905)	(.862, .912, .933)	(.042, .039, .039)	(.906, .893, .912)
Adj. Min. ESS	(1361, 1296, 1232)	(5511, 2723, 6030)	(8624, 7262, 11023)	(20167, 8040, 14331)	(36054, 24771, 21334)	(1880, 518, 778)	(26198, 26003, 21269)
Adj. Med. ESS	(2269, 2204, 3047)	(10051, 9402, 8689)	(24382, 24447, 22177)	(39621, 37935, 34757)	(49477, 51877, 45392)	(2334, 1750, 1686)	(50191, 48894, 51877)
Adj. Max. ESS	(5187, 2918, 7003)	(12774, 13682, 13423)	(35471, 35665, 42733)	(56157, 55119, 58686)	(55897, 59140, 60501)	(2723, 2529, 2529)	(58751, 57908, 59140)
$d = 49$ $\overline{CPU} = 31$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.236, .095, .208)	(.085, .056, .107)	(.158, .208, .108)	(.351, .191, .210)	(.767, .679, .717)	(.034, .057, .041)	(.616, .397, .605)
Med α	(.396, .380, .392)	(.435, .483, .480)	(.607, .617, .634)	(.733, .719, .746,)	(.897, .901, .890)	(.256, .263, .278)	(.846, .873, .857)
Max α	(.489, .578, .520)	(.609, .646, .645)	(.837, .868, .850)	(.935, .922, .927)	(.953, .955, .954)	(.362, .393 , .351)	(.951, .939, .934)
Min. ESS	(.032, .021, .034)	(.031, .010, .033)	(.069, .062, .045)	(.184, .091, .078)	(.420, .273, .348)	(.007, .009, .007)	(.307, .193, .294)
Med. ESS	(.049, .046, .053)	(.117, .123, .148)	(.300, .288, .340)	(.507, .492, .485)	(.751, .764, .751)	(.023, .027, .028)	(.681, .758, .705)
Max. ESS	(.082, .120, .098)	(.204, .272, .274)	(.673, .740, .695)	(.893, .852, .846)	(.923, .921, .921)	(.055, .046, .045)	(.926, .906, .895)
Adj. Min. ESS	(814, 534, 865)	(789, 254, 840)	(1757, 1578, 1145)	(4685, 2317, 1986)	(10695, 6951, 8861)	(178, 229, 178)	(7817, 4914, 7486)
Adj. Med. ESS	(1247, 1171, 1349)	(2979, 3132, 3768)	(7639, 7333, 8657)	(12910, 12528, 12350)	(19123, 19454, 19123)	(585, 687, 713)	(17341, 19302, 17952)
Adj. Max. ESS	(2088, 3055, 2495)	(5194, 6926, 6977)	(17137, 18843, 17697)	(22739, 21695, 21542)	(23503, 23452, 23452)	(1400, 1171, 1145)	(23580, 23070, 22790)
$d = 100$ $\overline{CPU} = 95$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.162, .089, .113)	(.139, .090, .075)	(.203, .137, .124)	(.246, .189, .233)	(.717, .611, .587)	(.061, .023, .020)	(.566, .422, .484)
Med α	(.383, .376, .373)	(.504, .477, .445)	(.671, .613, .600)	(.764, .758, .695)	(.908, .904, .891)	(.290, .298, .259)	(.870, .847, .864)
Max α	(.475, .574, .511)	(.582, .684, .631)	(.804, .860, .818)	(.900, .939, .932)	(.955, .957, .952)	(.376, .430, .389)	(.938, .948, .943)
Min. ESS	(.020, .015, .015)	(.047, .023, .016)	(.081, .023, .021)	(.127, .026, .033)	(.389, .212, .181)	(.015, .007, .004)	(.385, .311, .165)
Med. ESS	(.047, .051, .049)	(.157, .140, .110)	(.389, .309, .291)	(.572, .511, .440)	(.797, .784, .749)	(.034, .032, .022)	(.818, .786, .742)
Max. ESS	(.086, .205, .106)	(.221, .300, .235)	(.639, .720, .660)	(.835, .905, .888)	(.917, .930, .916)	(.055, .065, .042)	(.905, .922, .908)
Adj. Min. ESS	(166, 125, 125)	(391, 191, 133)	(675, 19, 175)	(1058, 216, 275)	(3243, 1767, 1509)	(125, 58, 33)	(2401, 1292, 1375)
Adj. Med. ESS	(391, 425, 408)	(1309, 1167, 917)	(3243, 2576, 2426)	(4769, 4260, 3668)	(6645, 6537, 6245)	(283, 266, 183)	(6153, 5745, 6187)
Adj. Max. ESS	(717, 1709, 883)	(1842, 2501, 1959)	(5328, 6003, 5503)	(6962, 7546, 7404)	(7646, 7754, 7638)	(458, 542, 350)	(7462, 7688, 7571)

Table 4.3.2: U-PC. Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = \{25, 49, 100\}$. Grey colour: K–S test does not support convergence.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\mu_\eta = \log(100)$
$d = 25$, $\overline{CPU} = 45$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.694, .697, .596)	(.745, .552, .624)	(.763, .654, .640)	(.812, .607, .648)	(.858, .713, .695)	(.693, .648, .660)	(.807, .809, .761)
Med α	(.744, .750, .728)	(.792, .745, .767)	(.844, .801, .805)	(.898, .873, .845)	(.933, .919, .763)	(.747, .745, .727)	(.940, .917, .895)
Max α	(.792, .792, .800)	(.831, .817, .824)	(.882, .888, .906)	(.934, .946, .955)	(.948, .957, .824)	(.787, .792, .787)	(.952, .949, .956)
Min ESS	(.044, .059, .042)	(.175, .140, .111)	(.376, .256, .186)	(.513, .211, .238)	(.554, .295, .367)	(.061, .064, .055)	(.538, .538, .457)
Med ESS	(.062, .070, .061)	(.212, .197, .182)	(.478, .389, .324)	(.658, .607, .502)	(.788, .760, .732)	(.072, .080, .066)	(.861, .808, .722)
Max ESS	(.116, .087, .130)	(.257, .250, .279)	(.554, .597, .714)	(.841, .873, .910)	(.867, .917, .928)	(.088, .094, .078)	(.890, .891, .913)
Adj. Min. ESS	(782, 1049, 747)	(3111, 2489, 1973)	(6684, 4551, 3307)	(9120, 3751, 4231)	(9849, 5244, 6524)	(1084, 1138, 978)	(9564, 9564, 8124)
Adj. Med. ESS	(1102, 1244, 1084)	(3769, 3502, 3236)	(8498, 6916, 5760)	(11698, 10791, 8924)	(14009, 13511, 13013)	(1280, 1422, 1173)	(15307, 14364, 12836)
Adj. Max. ESS	(2062, 1547, 2311)	(4569, 4444, 4960)	(9849, 10613, 12693)	(14951, 15520, 16178)	(15413, 16302, 16497)	(1564, 1671, 1387)	(15822, 15840, 16231)
$d = 49$, $\overline{CPU} = 100$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.622, .639, .580)	(.601, .607, .558)	(.552, .625, .575)	(.582, .651, .600)	(.733, .772, .721)	(.673, .609, .654)	(.782, .715, .766)
Med α	(.748, .763, .747)	(.779, .785, .788)	(.823, .838, .834)	(.845, .875, .861)	(.913, .912, .920)	(.742, .749, .752)	(.912, .904, .913)
Max α	(.820, .820, .816)	(.843, .837, .848)	(.897, .893, .898)	(.947, .951, .942)	(.954, .956, .956)	(.811, .813, .815)	(.956, .958, .953)
Min ESS	(.061, .055, .056)	(.134, .102, .121)	(.179, .191, .174)	(.176, .221, .195)	(.306, .394, .290)	(.057, .062, .060)	(.468, .419, .459)
Med ESS	(.081, .084, .087)	(.193, .196, .206)	(.400, .435, .410)	(.576, .621, .606)	(.750, .739, .742)	(.074, .082, .079)	(.776, .763, .783)
Max ESS	(.121, .191, .112)	(.252, .278, .279)	(.625, .661, .675)	(.880, .911, .888)	(.914, .913, .939)	(.091, .106, .095)	(.913, .922, .894)
Adj. Min. ESS	(488, 440, 448)	(1072, 816, 968)	(1432, 1528, 1392)	(1408, 1768, 1560)	(2448, 3152, 2320)	(456, 496, 480)	(3744, 3352, 3672)
Adj. Med. ESS	(648, 672, 696)	(1544, 1568, 1648)	(3200, 3480, 3280)	(4608, 4968, 4848)	(6000, 5912, 5936)	(592, 656, 632)	(6208, 6104, 6264)
Adj. Max. ESS	(968, 1528, 896)	(2016, 2224, 2232)	(5000, 5288, 5400)	(7040, 7288, 7104)	(7312, 7304, 7512)	(728, 848, 760)	(7304, 7376, 7152)
$d = 100$, $\overline{CPU} = 257$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.673, .645, .695)	(.640, .604, .582)	(.530, .639, .554)	(.611, .592, .565)	(.710, .634, .726)	(.698, .646, .615)	(.739, .668,.714)
Med α	(.761, .768, .763)	(.787, .787, .785)	(.819, .837, .819)	(.874, .879, .852)	(.925, .916, .917)	(.763, .764, .751)	(.923, .913, .908)
Max α	(.831, .830, .824)	(.848, .845, .845)	(.890, .922, .897)	(.942, .954, .952)	(.956, .956, .958)	(.820, .823, .803)	(.951, .958, .954)
Min ESS	(.041, .043, .051)	(.131, .100, .117)	(.141, .165, .175)	(.185, .177, .173)	(.305, .189, .344)	(.063, .056, .051)	(.406, .279, .376)
Med ESS	(.082, .099, .091)	(.205, .220,.197)	(.402, .454, .401)	(.607, .620, .569)	(.769, .750, .770)	(.086, .091, .069)	(.809, .778, .763)
Max ESS	(.127, .216, .139)	(.269, .303,.280)	(.647, .706, .671)	(.871, .896, .898)	(.915, .912, .921)	(.105, .119, .096)	(.902, .916, .909)
Adj. Min. ESS	(128, 134, 19)	(408, 311, 364)	(439, 514, 545)	(576, 551, 539)	(949, 588, 1071)	(196, 174, 159)	(1264, 868, 1170)
Adj. Med. ESS	(255, 308, 19)	(638, 685, 613)	(1251, 1413, 1248)	(1889, 1930, 1771)	(2394, 2335, 2397)	(268, 283, 215)	(2518, 2422, 2375)
Adj. Max. ESS	(395, 672, 31)	(837, 943, 872)	(2014, 2198, 2089)	(2711, 2789, 2795)	(2848, 2839, 2867)	(327, 370, 299)	(2808, 2851, 2830)

Table 4.3.3: PC-RWM. Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. Grey: KS test does not support convergence.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\mu_\eta = \log(100)$
$d = 25$ $\overline{CPU} = 49$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.695, .698, .596)	(.745, .553, .625)	(.762, .652, .640)	(.811, .608, .649)	(.857, .712, .759)	(.693, .648, .661)	(.807, .809, .760)
Med α	(.745, .751, .731)	(.793, .746, .768)	(.844, .800, .806)	(.899, .872, .846)	(.932, .919, .920)	(.748, .745, .728)	(.940, .917, .895)
Max α	(.791, .792, .800)	(.832, .818, .824)	(.882, .888, .906)	(.934, .946, .955)	(.948, .957, .959)	(.788, .792, .788)	(.952, .949, .956)
Min ESS	(.148, .161, .130)	(.250, .194, .183)	(.415, .276, .222)	(.522, .230, .248)	(.565, .298, .371)	(.159, .150, .148)	(.554, .553, .458)
Med ESS	(.167, .171, .159)	(.295, .265, .257)	(.508, .425, .378)	(.683, .625, .516)	(.792, .766, .736)	(.178, .178, .169)	(.871, .816, .747)
Max ESS	(.208, .182, .217)	(.337, .334, .363)	(.594, .633, .743)	(.860, .874, .925)	(.874, .927, .937)	(.193, .198, .187)	(.900, .887, .915)
Adj. Min. ESS	(2416, 2629, 2122)	(4082, 3167, 2988)	(6776, 4506, 3624)	(8522, 3755, 4049)	(9224, 4865, 6057)	(2596, 2449, 2416)	(9045, 9029, 7478)
Adj. Med. ESS	(2727, 2792, 2596)	(4816, 4327, 4196)	(8294, 6939, 6171)	(11151, 10204, 8424)	(12931, 12506, 12016)	(2906, 2906, 2759)	(14220, 13322, 12196)
Adj. Max. ESS	(3396, 2971, 3543)	(5502, 5453, 5927)	(9698, 10335, 12131)	(14041, 14269, 15102)	(14269, 15135, 15298)	(3151, 3233, 3053)	(14694, 14482, 14939)
$d = 49$ $\overline{CPU} = 112$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.622, .638, .581)	(.602, .608, .558)	(.552, .625, .575)	(.584, .652, .601)	(.733, .772, .721)	(.672, .607, .654)	(.782, .716, .766)
Med α	(.747, .762, .748)	(.781, .784, .788)	(.823, .837, .835)	(.846, .875, .861)	(.913, .912, .920)	(.743, .748, .751)	(.912, .904, .913)
Max α	(.820, .819, .816)	(.842, .837, .847)	(.897, .893, .898)	(.947, .951, .94)	(.955, .956, .956)	(.811, .813, .814)	(.956, .958, .953)
Min ESS	(.117, .115, .105)	(.177, .147, .151)	(.196, .211, .189)	(.188, .233, .205)	(.318, .397, .288)	(.116, .119, .113)	(.477, .425, .457)
Med ESS	(.145, .151, .148)	(.239, .242, .255)	(.422, .459, .434)	(.584, .628, .616)	(.754, .746, .745)	(.137, .145, .141)	(.776, .771, .784)
Max ESS	(.180, .242, .174)	(.302, .31, .315)	(.647, .671, .678)	(.884, .906, .876)	(.920, .921, .926)	(.158, .170, .164)	(.913, .919, .898)
Adj. Min. ESS	(836, 821, 750)	(1264, 1050, 1079)	(1400, 1507, 1350)	(1343, 1664, 1464)	(2271, 2836, 2057)	(829, 850, 807)	(3407, 3036, 3264)
Adj. Med. ESS	(1036, 1079, 1057)	(1707, 1729, 1821)	(3014, 3279, 3100)	(4171, 4486, 4400)	(5386, 5329, 5321)	(979, 1036, 1007)	(5543, 5507, 5600)
Adj. Max. ESS	(1286, 1729, 1243)	(2157, 2214, 2250)	(4621, 4793, 4843)	(6314, 6471, 6257)	(6571, 6579, 6614)	(1129, 1214, 1171)	(6521, 6564, 6414)
$d = 100$ $\overline{CPU} = 279$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.674, .646, .697)	(.641, .604, .584)	(.527, .637, .554)	(.616, .591, .564)	(.710, .632, .726)	(.699, .645, .615)	(.738, .668, .714)
Med α	(.760, .768, .763)	(.787, .788, .786)	(.819, .837, .820)	(.875, .879, .852)	(.925, .916, .918)	(.762, .764, .751)	(.923, .913, .908)
Max α	(.831, .831, .825)	(.849, .845, .845)	(.890, .923, .897)	(.942, .955, .952)	(.956, .956, .958)	(.820, .823, .803)	(.951, .959, .954)
Min ESS	(.078, .083, .088)	(.154, .126, .140)	(.149, .184, .186)	(.195, .184, .180)	(.302, .194, .347)	(.098, .092, .086)	(.413, .277, .382)
Med ESS	(.118, .134, .129)	(.232, .248, .221)	(.419, .466, .413)	(.610, .626, .575)	(.770, .754, .773)	(.124, .128, .106)	(.809, .782, .767)
Max ESS	(.159, .249, .173)	(.295, .330, .310)	(.649, .716, .681)	(.872, .901, .902)	(.922, .913, .924)	(.143, .159, .132)	(.906, .920, .910)
Adj. Min. ESS	(224, 238, 252)	(442, 361, 401)	(427, 528, 533)	(559, 528, 516)	(866, 556, 995)	(281, 264, 247)	(1184, 794, 1095)
Adj. Med. ESS	(338, 384, 370)	(665, 711, 634)	(1201, 1336, 1184)	(1749, 1795, 1649)	(2208, 2162, 2216)	(356, 367, 304)	(2320, 2242, 2199)
Adj. Max. ESS	(456, 714, 496)	(846, 946, 889)	(1861, 2053, 1953)	(2500, 2584, 2586)	(2644, 2618, 2649)	(410, 456, 378)	(2598, 2638, 2609)

Table 4.3.4: PC-MALA. Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = \{25, 49, 100\}$. Grey colour: K–S test does not support convergence.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\mu_\eta = \log(100)$
$d = 25$ $\overline{CPU} = 50$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.695, .698, .595)	(.746, .553, .626)	(.763, .653, .639)	(.812, .607, .649)	(.879, .762, .693)	(.693, .648, .661)	(.807, .810, .760)
Med α	(.745, .750, .729)	(.793, .745, .767)	(.845, .802, .806)	(.898, .872, .846)	(.932, .922, .909)	(.748, .745, .728)	(.940, .918, .895)
Max α	(.792, .793, .801)	(.831, .818, .823)	(.882, .888, .905)	(.934, .947, .954)	(.942, .954, .957)	(.788, .792, .787)	(.951, .949, .957)
Min ESS	(.255, .257, .193)	(.353, .250, .273)	(.487, .307, .287)	(.533, .263, .261)	(.660, .399, .285)	(.246, .251, .247)	(.571, .572, .470)
Med ESS	(.327, .329, .301)	(.419, .361, .373)	(.569, .485, .491)	(.721, .671, .549)	(.822, .847, .739)	(.341, .331, .318)	(.883, .831, .765)
Max ESS	(.376, .388, .370)	(.492, .466, .506)	(.667, .706, .785)	(.897, .904, .949)	(.905, .923, .932)	(.386, .386, .388)	(.909, .915, .929)
Adj. Min. ESS	(4163, 4196, 3151)	(5763, 4082, 4457)	(7951, 5012, 4686)	(8702, 4294, 4261)	(10776, 514, 4653)	(4016, 4098, 4033)	(9322, 9339, 76730)
Adj. Med. ESS	(5339, 5371, 4914)	(6841, 5894, 6090)	(9290, 7918, 8016)	(11771, 10955, 8963)	(13420, 13829, 12065)	(5567, 5404, 5192)	(14416, 13567, 12490)
Adj. Max. ESS	(6139, 6335, 6041)	(8033, 7608, 8261)	(10890, 11527, 12816)	(14645, 14759, 15494)	(14776, 15069, 15216)	(6302, 6302, 6335)	(14841, 14939, 15167)
$d = 49$ $\overline{CPU} = 114$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.625, .638, .583)	(.600, .607, .560)	(.551, .624, .576)	(.582, .651, .601)	(.705, .699, .735)	(.673, .717, .653)	(.781, .716, .766)
Med α	(.747, .762, .748)	(.780, .785, .788)	(.823, .838, .835)	(.846, .876, .862)	(.917, .895, .919)	(.743, .748, .752	(.912, .904, .912)
Max α	(.821, .820, .816)	(.842, .837, .847)	(.897, .893, .898)	(.947, .951, .942)	(.954, .956, .959)	(.811, .814, .814)	(.956, .957, .953)
Min ESS	(.229, .227, .213)	(.275, .249, .231)	(.260, .265, .232)	(.229, .256, .230)	(.316, .289, .330)	(.250, .239, .268)	(.510, .446, .474)
Med ESS	(.297, .312, .296)	(.366, .373, .379)	(.503, .532, .503)	(.605, .662, .658,)	(.812, .788, .799)	(.298, .301, .306)	(.799, .786, .794)
Max ESS	(.370, .394, .364)	(.453, .460, .469)	(.732, .714, .750)	(.908, .920, .908)	(.921, .930, .958)	(.379, .383, .393)	(.922, .927, .909)
Adj. Min. ESS	(1607, 1593, 1495)	(1930, 1747, 1621)	(1825, 1860, 1628)	(1607, 1796,1614)	(2218, 2028, 2316)	(1754, 1677, 1881)	(3579, 3130, 3326)
Adj. Med. ESS	(2084, 2189, 2077)	(2568, 2618, 2660)	(3530, 3733, 3530)	(4246, 4646, 4618)	(5698, 5530, 5607)	(2091, 2112, 2147)	(5607, 5516, 5572)
Adj. Max. ESS	(2596, 2765, 2554)	(3179, 3228, 3291)	(5137, 5011, 5263)	(6372, 6456, 6372)	(6463, 6526, 6723)	(2660, 2688, 2758)	(6470, 6505, 6379)
$d = 100$ $\overline{CPU} = 307$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.674, .646, .696)	(.639, .605, .582)	(.530, .638, .554)	(.612, .593, .565)	(.710, .634, .725)	(.698, .646, .615)	(.738, .669, .713)
Med α	(.761, .768, .762)	(.787, .788, .785)	(.819, .837, .820)	(.874, .879, .851)	(.925, .916, .917)	(.763, .763, .750)	(.923, .913, .908)
Max α	(.830, .831, .825)	(.848, .845, .845)	(.890, .922, .896)	(.943, .954, .952)	(.956, .956, .958)	(.819, .822, .802)	(.951, .958, .954)
Min ESS	(.227, .237, .231)	(.253, .217, .234)	(.190, .267, .232)	(.222, .211, .193)	(.321, .210, .365)	(.248, .223, .208)	(.429, .330, .393)
Med ESS	(.271, .287, .282)	(.358, .363, .349)	(.494, .521, .481)	(.639, .646, .597)	(.780, .765, .783)	(.285, .287, .274)	(.829, .795, .780)
Max ESS	(.355, .392, .327)	(.433, .452, .445)	(.704, .761, .746)	(.891, .905, .927)	(.928, .932, .941)	(.336, .351, .323)	(.910, .921, .928)
Adj. Min. ESS	(590, 616, 600)	(657, 564, 608)	(494, 694, 603)	(577, 548, 501)	(834, 545, 948)	(644, 579, 540)	(1114, 857, 1021)
Adj. Med. ESS	(704, 745, 732)	(930, 943, 906)	(1283, 1353, 1249)	(1660, 1678, 1551)	(2026, 1987, 2034)	(740, 745, 712)	(2153, 2065, 2026)
Adj. Max. ESS	(922, 1018, 849)	(1125, 1174, 1156)	(1829, 1977, 1938)	(2314, 2351, 2408)	(2410, 2421, 2444)	(873, 912, 839)	(2364, 2392, 2410)

Table 4.3.5: Algorithm Christensen et al. (2006). Relative ESS, average CPU time and adjusted ESS for dimensions $d = \{25, 49, 100\}$. The algorithm was tuned so that it achieved acceptance rates $57\% - 59\%$.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta = \log(100)}$
$d = 25$ $\overline{CPU} = 24$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min. ESS	(.120, .115, .115)	(.153, .139, .146)	(.140, .138, .141)	(.141, .121, .138)	(.140, .113, .121)	(.162, .156, .154)	(.173, .170, .171)
Med. ESS	(.179, .168, .184)	(.184, .178, .184)	(.179, .166, .175)	(.169, .160, .164)	(.151, .127, .142)	(.185, .183, .183)	(.179, .182, .180)
Max. ESS	(.258, .236, .238)	(.199, .212, .222)	(.210, .232, .232)	(.189, .270, .233)	(.201, .431, .239)	(.196, .224, .217)	(.200, .195, .201)
Adj. Min. ESS	(3994, 3827, 3827)	(5092, 4626, 4859)	(4659, 4593, 4693)	(4693, 4027, 4593)	(4659, 3761, 4027)	(5392, 5192, 5125)	(5758, 5658, 5691)
Adj. Med. ESS	(5957, 5591, 6124)	(6124, 5924, 6124)	(5957, 5525, 5824)	(5625, 5325, 5458)	(5025, 4227, 4726)	(6157, 6091, 6091)	(5957, 6057, 5991)
Adj. Max. ESS	(8587, 7854, 7921)	(6623, 7056, 7388)	(6989, 7721, 7721)	(6290, 8986, 7755)	(6690, 14344, 7954)	(6523, 7455, 7222)	(6656, 6490, 6690)
$d = 49$ $\overline{CPU} = 74$							
Min. ESS	(.086, .084, .087)	(.110, .103, .107)	(.101, .102, .095)	(.097, .086, .084)	(.098, .088, .100)	(.120, .110, .124)	(.128, .120, .126)
Med. ESS	(.135, .137, .133)	(.137, .136, .139)	(.128, .129, .131)	(.119, .112, .116)	(.110, .100, .108)	(.137, .139, .138)	(.136, .135, .135)
Max. ESS	(.175, .175, .201)	(.174, .173, .167)	(.194, .195, .212)	(.184, .258, .222)	(.196, .267, .178)	(.154, .158, .159)	(.150, .187, .135)
Adj. Min. ESS	(926 904 937)	(1184 1109 1152)	(1087 1098 1023)	(1044 926 904)	(1055 948 1077)	(1292 1184 1335)	(1378 1292 1357)
Adj. Med. ESS	(1454, 1475, 1432)	(1475, 1464, 1497)	(1378, 1389, 1410)	(1281, 1206, 1249)	(1184, 1077, 1163)	(1475, 1497, 1486)	(1464, 1454, 1454)
Adj. Max. ESS	(1884, 1884, 2164)	(1873, 1863, 1798)	(2089, 2100, 2283)	(1981, 2778, 2390)	(2110, 2875, 1917)	(1658, 1701, 1712)	(1615, 2013, 1454)
$d = 100$ $\overline{CPU} = 279$							
Min. ESS	(.072, .083, .064)	(.086, .083, .080)	(.075, .078, .070)	(.070, .064, .058)	(.071, .063, .065)	(.092, .090, .087)	(.095, .089, .092)
Med. ESS	(.101, .101, .102)	(.103, .102, .103)	(.096, .098, .098)	(.091, .090, .086)	(.079, .072, .073)	(.103, .104, .104)	(.101, .100, .101)
Max. ESS	(.145, .130, .131)	(.128, .132, .127)	(.165, .143, .138)	(.187, .153, .157)	(.154, .218, .257)	(.114, .117, .120)	(.115, .137, .116)
Adj. Min. ESS	(207, 238, 184)	(247, 238, 230)	(215, 224, 201)	(201, 184, 166)	(204, 181, 187)	(264, 258, 250)	(273, 255, 264)
Adj. Med. ESS	(290, 290, 293)	(296, 293, 296)	(276, 281, 281)	(261, 258, 247)	(227, 207, 210)	(296, 299, 299)	(290, 287, 290)
Adj. Max. ESS	(416, 373, 376)	(367, 379, 365)	(474, 410, 396)	(537, 439, 451)	(442, 626, 738)	(327, 336, 344)	(330, 393, 333)

Table 4.3.6: Algorithm of Christensen et al. (2006) and PC-MALA. Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimension $d = 196$. Grey: KS test does not support convergence.

	$\mu_\eta = \log(1)$	$\mu_\eta = \log(10)$					$\mu_\eta = \log(100)$
$d = 196$ CRS $\overline{CPU} = 1040$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min ESS	(.055, .049, .045)	(.064, .063, .061)	(.061, .056, .057)	(.015, .046, 0)	(.046, .047, .049)	(.071, .067, .067)	(.071, .070, .069)
Med ESS	(.077, .078, .077)	(.078, .078, .078)	(.078, .075, .075)	(.021, .069, 0)	(.075, .057, .060)	(.079, .079, .079)	(.077, .077, .077)
Max ESS	(.100, .108, .107)	(.095, .094, .096)	(.103, .129, .111)	(.027, .159, 0)	(.110, .183, .145)	(.090, .090, .090)	(.098, .106, .101)
Adj. Min ESS	(42.31 37.7, 34.6)	(49.2, 48.5, 46.9)	(46.9, 43.1, 43.8)	(11.5, 35.4, 0)	(35.4, 36.1, 37.7)	(54.6, 51.5, 51.5)	(54.6, 53.8, 53.1)
Adj. Med ESS	(59.2, 60.0, 59.2)	(60.0, 60.0, 60.0)	(60.0, 57.7, 57.7)	(16.1 53.1, 0)	(57.7, 43.8, 46.1)	(60.7, 60.8, 60.8)	(59.2, 59.2, 59.2)
Adj. Max ESS	(76.9, 83.1, 82.3)	(73.1, 72.3, 73.8)	(79.2, 99.2, 85.3)	(20.7, 122.3, 0)	(84.6, 140.8, 111.5)	(69.2, 69.2, 69.2)	(75.3, 81.5, 77.7)
$d = 196$ PC-MALA $\overline{CPU} = 1208$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.662, .552, .654)	(.651, .574, .477)	(.611, .580, .567)	(.616, .673, .566)	(.725, .621, .684)	(.636, .645, .642)	(.748, .679, .681)
Med α	(.776, .770, .770)	(.804, .792, .794)	(.846, .832, .838)	(.879, .877, .869)	(.922, .906, .914)	(.770, .768, .769)	(.920, .920, .922)
Max α	(.827, .833, .826)	(.853, .853, .837)	(.906, .913, .926)	(.941, .954, .959)	(.958, .957, .960)	(.821, .825, .820)	(.956, .957, .959)
Min ESS	(.221, .168, .201)	(.242, .201, .163)	(.240, .224, .222)	(.235, .222, .185)	(.322, .198, .274)	(.200, .208, .208)	(.420, .372, .354)
Med ESS	(.266, .256, .256)	(.350, .332, .332)	(.519, .480, .481)	(.659, .628, .633)	(.800, .750, .769)	(.262, .257, .259)	(.803, .805, .814)
Max ESS	(.310, .327, .308)	(.421, .454, .408)	(.687, .748, .821)	(.875, .923, .947)	(.932, .930, .958)	(.289, .298, .294)	(.919, .940, .936)
Adj. Min ESS	(146, 111, 133)	(160, 133, 108)	(159, 148, 147)	(156, 147, 123)	(213, 131, 181)	(132, 138, 138)	(278, 246, 234)
Adj. Med ESS	(176, 170, 170)	(232, 220, 220)	(344, 318, 319)	(436, 416, 419)	(530, 497, 509)	(174, 170, 172)	(532, 533, 539)
Adj. Max ESS	(205, 217, 204)	(279, 301, 270)	(455, 495, 544)	(579, 611, 627)	(617, 616, 634)	(191, 197, 195)	(609, 623, 620)

Table 4.3.7: Algorithm of Christensen et al. (2006) and PC-MALA. Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimension $d = 400$. Grey: KS test does not support convergence.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 400$ CRS $\overline{CPU} = 4949$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min ESS	(.036, .035, .039)	(.048, .046, .049)	(.044, .044, .040)	(.039, .039, .035)	(.037, .037, .036)	(.051, .052, .052)	(.052, .051, .053)
Med ESS	(.059, .060, .059)	(.060, .060, .060)	(.057, .057, .057)	(.052, .053, .052)	(.043, .044, .043)	(.060, .060, .060)	(.059, .059, .059)
Max ESS	(.080, .079, .081)	(.077, .074, .075)	(.089, .084, .083)	(.109, .102, .106)	(.147, .126, .133)	(.069, .069, .071)	(.072, .078, .072)
Adj Min ESS	(5.8, 5.7, 6.3)	(7.8, 7.4, 7.9)	(7.1, 7.1, 6.5)	(6.3, 6.3, 5.7)	(6.0, 6.0, 5.8)	(8.2, 8.4, 8.4)	(8.4, 8.2, 8.6)
Adj Med ESS	(9.5, 9.7, 9.5)	(9.7, 9.7, 9.7)	(9.2, 9.2, 9.2)	(8.4, 8.6, 8.4)	(7.0, 7.1, 7.0)	(9.7, 9.7, 9.7)	(9.5, 9.5, 9.5)
Adj Max ESS	(12.9, 12.8, 13.1)	(12.4, 12.0, 12.1)	(14.4, 13.6, 13.4)	(17.6, 16.5, 17.1)	(23.8, 20.4, 21.5)	(11.2, 11.2, 11.5)	(11.6, 12.6, 11.6)
$d = 400$ PC-MALA $\overline{CPU} = 4137$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
Min α	(.662, .624, .654)	(.560, .501, .535)	(.497, .498, .572)	(.570, .543, .565)	(.639, .631, .621)	(.642, .560, .615)	(.672, .626, .672)
Med α	(.782, .775, .778)	(.797, .802, .804)	(.845, .842, .834)	(.883, .877, .868)	(.916, .910, .910)	(.775, .776, .779)	(.923, .920, .917)
Max α	(.836, .848, .829)	(.844, .863, .864)	(.921, .938, .931)	(.955, .961, .958)	(.961, .959, .960)	(.828, .833, .834)	(.960, .961, .960)
Min ESS	(.168, .161, .159)	(.166, .164, .170)	(.159, .149, .206)	(.187, .153, .170)	(.213, .200, .189)	(.180, .164, .175)	(.293, .251, .293)
Med ESS	(.221, .200, .215)	(.278, .276, .284)	(.379, .374, .372)	(.447, .434, .434)	(.482, .478, .480)	(y.215, .211, .215)	(.487, .485, .483)
Max ESS	(.277, .399, .316)	(.343, .396, .355)	(.481, .500, .492)	(.501, .506, .507)	(.506, .505, .502)	(.240, .268, .263)	(.505, .505, .508)
Adj Min ESS	(32.5, 31.1, 30.7)	(32.1, 31.7, 32.9)	(30.7, 28.8, 39.8)	(36.2, 29.6, 32.9)	(41.2, 38.7, 36.5)	(34.8, 31.7, 33.8)	(56.7, 48.5, 56.7)
Adj Med ESS	(42.7, 38.7, 41.6)	(53.8, 53.4, 54.9)	(73.3, 72.3, 71.9)	(86.4, 83.9, 83.9)	(93.2, 92.4, 92.8)	(41.6, 40.8, 41.6)	(94.2, 93.8, 93.4)
Adj Max ESS	(53.6, 77.2, 61.1)	(66.3, 76.6, 68.6)	(93.0, 96.7, 95.1)	(96.9, 97.8, 98.0)	(97.8, 97.7, 97.1)	(46.4, 51.8, 50.9)	(97.7, 97.7, 98.2)

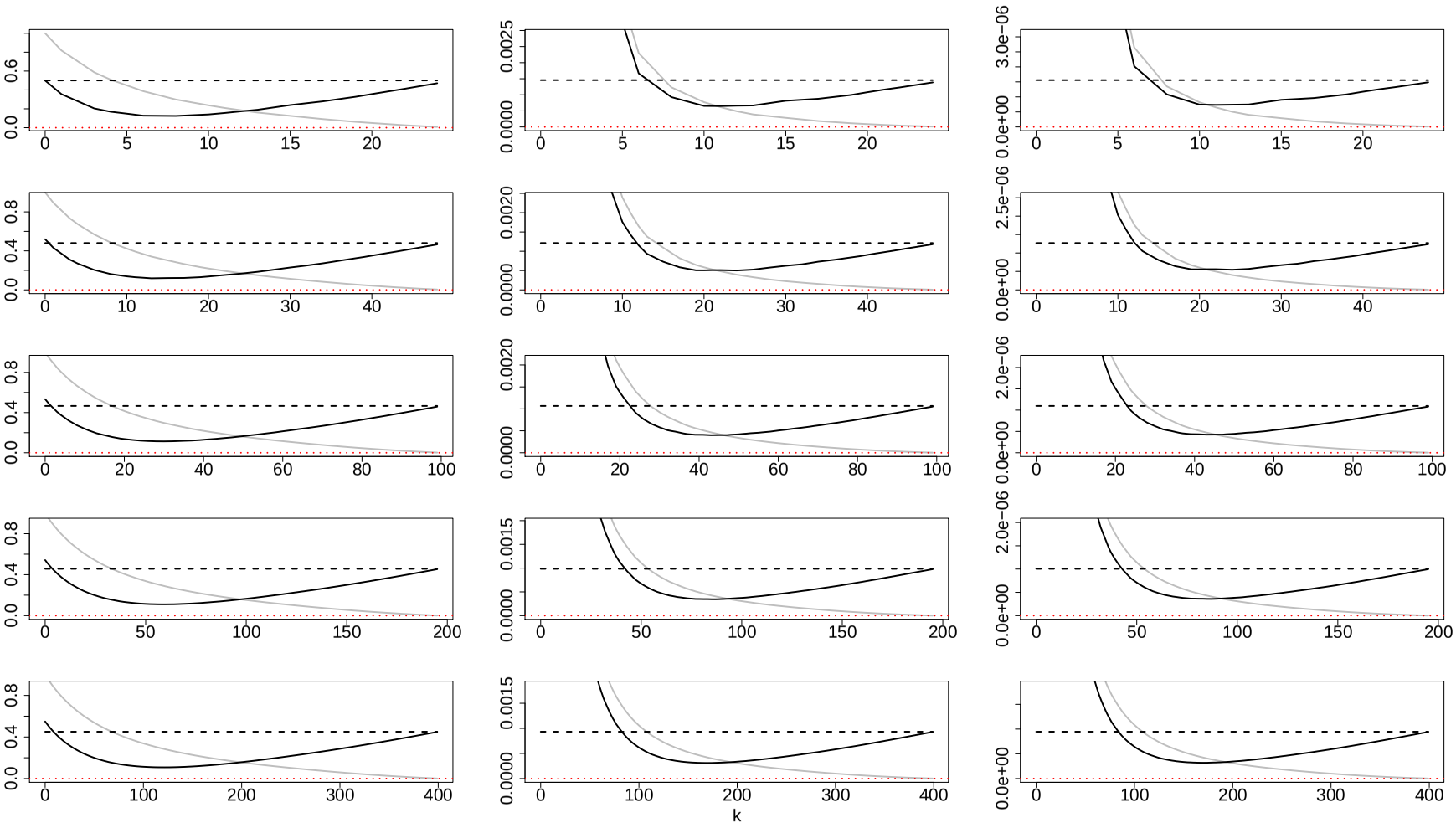

Figure 4.6: Plots of $\bar{v}^a$ (grey solid line), $\bar{v}^t$ (black dashed line) and $\bar{v}^{\mathbb{E}}$ (black solid line) against the number k of principal components. The prior correlation matrix $\boldsymbol{R}$ is constructed using the Matérn family with $\kappa = 1.5$. Top to Bottom:$d = \{25, 49, 100, 196, 400\}$. Left to Right: $\phi = \{1, 10, 100\}$

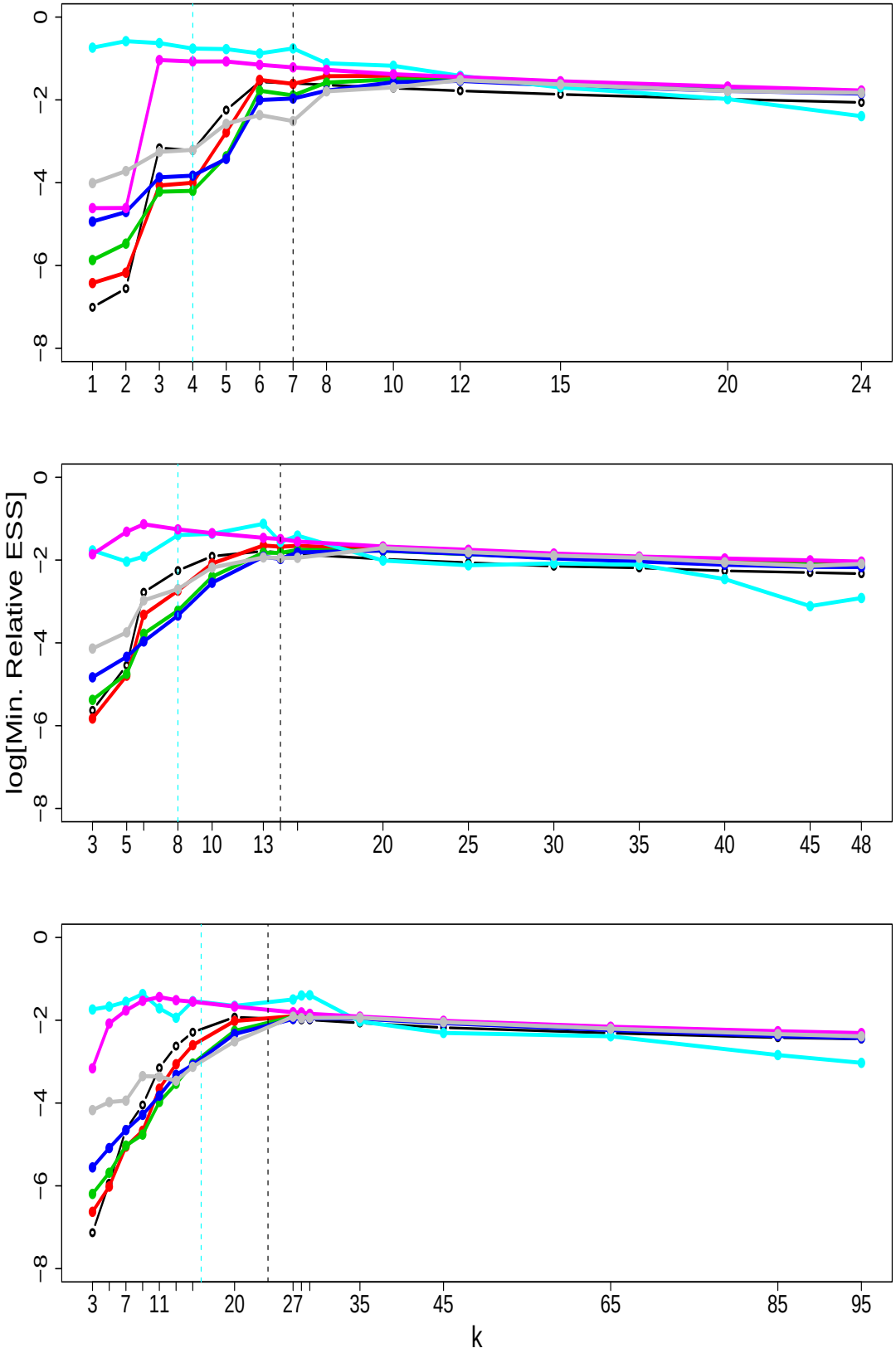

Figure 4.7: Algorithm PC-MALA. Logarithm of minimum relative ESS against different values of k. Top to bottom: Dimension, $d = \{25, 49, 100\}$. The correlation matrix $\boldsymbol{R}$ is constructed using the Matern family with $\kappa = 1.5$.

Table 4.3.8: Algorithm of PC-MALA (left) and Christensen et al. (2006) (right). Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 25, 49, 100$. The acceptance rates displayed correspond to PC-MALA. Grey: KS test does not support convergence. The correlation matrix $\boldsymbol{R}$ is constructed using the Matern correlation function with $\kappa = 1.5$

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 25$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
$\overline{CPU} = (75, 26)$							
Min. α	.515	.525	.526	.542	.797	.488	.327
Med. α	.617	.626	.625	.633	.909	.598	.595
Max. α	.661	.659	.664	.678	.932	.640	.704
Min. ESS	(.213, .131)	(.217, .170)	(.151, .165)	(.138, .162)	(.470, .141)	(.322, .175)	(.083, .170)
Med. ESS	(.300, .190)	(.292, .181)	(.253, .184)	(.220, .184)	(.765, .158)	(.333, .184)	(.185, .185)
Max. ESS	(.430, .271)	(.337, .200)	(.309, .203)	(.272, .200)	(.847, .208)	(.342, .191)	(.261, .196)
Adj. Min. ESS	(2242, 4031)	(2284, 5231)	(1589, 5077)	(1453, 4985)	(4947, 4338)	(3389, 5385)	(874, 5231)
Adj. Med. ESS	(3158, 5846)	(3074, 5569)	(2663, 5662)	(2316, 5662)	(8053, 4862)	(3505, 5662)	(1947, 5692)
Adj. Max. ESS	(4526, 8338)	(3547, 6154)	(3253, 6246)	(2863, 6154)	(8916, 6400)	(3600, 5877)	(2747, 6031)
$d = 49$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
$\overline{CPU} = (184, 95)$							
Min. α	.561	.561	.562	.565	.640	.548	.536
Med. α	.644	.646	.651	.659	.651	.637	.664
Max. α	.698	.697	.699	.702	.699	.682	.728
Min. ESS	(.171, .104)	(.190, .128)	(.165, .125)	(.144, .119)	(.246, .087)	(.235, .134)	(.140, .129)
Med. ESS	(.226, .136)	(.230, .139)	(.211, .138)	(.202, .137)	(.680, .104)	(.241, .137)	(.205, .139)
Max. ESS	(.273, .163)	(.246, .146)	(.250, .152)	(.234, .157)	(.922, .225)	(.247, .140)	(.242, .145)
Adj. Min. ESS	(1629, 876)	(1810, 1078)	(1571, 1053)	(1371, 1002)	(2343, 733)	(2238, 1128)	(1333, 1086)
Adj. Med. ESS	(2152, 1145)	(2190, 1171)	(2010, 1162)	(1924, 1154)	(6476, 876)	(2295, 1154)	(1952, 1171)
Adj. Max. ESS	(2600, 1373)	(2343, 1229)	(2381, 1280)	(2229, 1322)	(8781, 1895)	(2352, 1179)	(2305 1221)
$d = 100$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
$\overline{CPU} = (567, 336)$							
Min. α	.621	.621	.602	.571	.585	.618	.515
Med. α	.660	.658	.660	.663	.899	.650	.679
Max. α	.681	.682	.687	.699	.951	.679	.739
Min. ESS	(.144, .088)	(.161, .096)	(.153, .094)	(.148, .091)	(.204, .069)	(.172, .102)	(.148, .096)
Med. ESS	(.167, .101)	(.173, .103)	(.168, .103)	(.168, .103)	(.714, .081)	(.178, .105)	(.181, .103)
Max. ESS	(.233, .135)	(.194, .117)	(.199, .123)	(.187, .117)	(.926, .169)	(.184, .108)	(.211, .110)
Adj. Min. ESS	(203, 210)	(227 229)	(216, 224)	(209, 217)	(288, 164)	(243, 243)	(209, 229)
Adj. Med. ESS	(236, 240)	(244, 245)	(237, 245)	(237, 245)	(1007, 193)	(251, 250)	(255, 245)
Adj. Max. ESS	(329, 321)	(274, 279)	(281, 293)	(264, 279)	(1307, 402)	(260, 257)	(298, 262)

Table 4.3.9: Algorithm of PC-MALA (left) and Christensen et al. (2006) (right). Minimum, median and maximum acceptance rates (α), relative ESS, average CPU time and adjusted ESS for dimensions $d = 196, 400$. The acceptance rates displayed correspond to PC-MALA. Grey: KS test does not support convergence. The correlation matrix $\boldsymbol{R}$ is constructed using the Matern correlation function with $\kappa = 1.5$.

	$\boldsymbol{\mu_\eta = \log(1)}$			$\boldsymbol{\mu_\eta = \log(10)}$			$\boldsymbol{\mu_\eta} = \log(100)$
$d = 196$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
$\overline{CPU} = (1737, 979)$							
Min. α	.560	.637	.627	.590	.576	.624	.560
Med. α	.704	.678	.679	.684	.889	.673	.704
Max. α	.749	.701	.707	.721	.950	.696	.749
Min. ESS	(.130, .063)	(.120, .070)	(.117, .071)	(.111, .065)	(.178, .051)	(.126, .076)	(.130, .074)
Med. ESS	(.164, .078)	(.133, .079)	(.135, .079)	(.142, .079)	(.689, .061)	(.132, .080)	(.164, .079)
Max. ESS	(.190, .096)	(.146, .087)	(.146, .086)	(.158, .094)	(.897, .155)	(.141, .082)	(.190, .085)
Adj. Min. ESS	(59.9, 51.5)	(55.3, 57.2)	(53.9, 58.0)	(51.1, 53.1)	(82.0, 41.7)	(58.0, 62.1)	(59.9, 60.5)
Adj. Med. ESS	(75.5, 63.7)	(61.3, 64.6)	(62.2, 64.6)	(65.4, 64.6)	(317.3, 49.8)	(60.8, 65.4)	(75.5, 64.6)
Adj. Max. ESS	(87.5, 78.4)	(67.2, 71.1)	(67.2, 70.3)	(72.8, 76.8)	(413.1, 126.7)	(64.9, 67.0)	(87.5, 69.5)
$d = 400$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 0.3, \phi = 10)$	$(\sigma^2 = 1, \phi = 10)$	$(\sigma^2 = 3, \phi = 10)$	$(\sigma^2 = 1, \phi = 1)$	$(\sigma^2 = 1, \phi = 100)$	$(\sigma^2 = 1, \phi = 10)$
$\overline{CPU} = (5251, 4690)$							
Min. α	.630	.629	.631	.583	.578	.616	.510
Med. α	.685	.685	.687	.691	.886	.681	.705
Max. α	.706	.707	.713	.727	.959	0.704	.753
Min. ESS	(.083, .051)	(.089, .055)	(.092, .053)	(.092, .051)	(.174, .036)	(.093, .058)	(.101, .055)
Med. ESS	(.096, .059)	(.100, .060)	(.103, .060)	(.111, .060)	(.458, .046)	(.097, .060)	(.133, .060)
Max. ESS	(.134, .083)	(.110, .067)	(.115, .068)	(.144, .081)	(.500, .140)	(.102, .063)	(.171, .065)
Adj. Min. ESS	(12.6, 8.7)	(13.6, 9.4)	(14.0, 9.0)	(14.0, 8.7)	(26.5, 6.1)	(14.2, 9.9)	(15.4, 9.4)
Adj. Med. ESS	(14.6, 10.1)	(15.2, 10.2)	(15.7, 10.2)	(16.9, 10.2)	(69.8, 7.8)	(14.8, 10.2)	(20.3, 10.2)
Adj. Max. ESS	(20.4, 14.2)	(16.8, 11.4)	(17.5, 11.6)	(21.9, 13.8)	(76.2, 23.9)	(15.5, 10.7)	(26.1, 11.1)

4.4 Discussion

In this Chapter we have presented three MCMC schemes, U-PC, PC-RWM, PC-MALA, for sampling from the posterior distribution of the latent process conditional on the parameters. All three algorithms update each component of the process separately using as the proposal distribution an approximation to the true conditional Gibbs sampler proposal $\pi(s_i|\boldsymbol{s}_{-i}, \boldsymbol{y})$. This approximation $\pi(s_i|\tilde{\boldsymbol{p}}, \boldsymbol{y})$ instead of conditioning on $(d-1)$ components of $\boldsymbol{S}$, conditions on a small number, k, of principal components $\tilde{\boldsymbol{p}}$. Additionally, the algorithms PC-RWM and PC-MALA subsequenlty update $\tilde{\boldsymbol{p}}$ using either a RWM or MALA update. We have also provided a sufficient diagnostic for choosing the optimal number, k, of principal componnets on which to condition.

We saw, that by only updating the components of $\boldsymbol{S}$ conditionally on the few principal components was not sufficient and the U-PC algorithm performed poorly. However, the additional update of the principal components, schemes PC-RWM and PC-MALA, considerably improved the mixing of the algorithm with PC-MALA being the best performing algorithm. In summary, in the case of the exponential correlation function, PC-MALA has always provided both better ESS and adjusted ESS than the algorithm of Christensen et al. (2006) whereas in the case of the Matérn correlation function of order $\kappa = 1.5$ it always performs better in terms of ESS. In terms of minimum adjusted ESS it performs similarly to the algorihtm of Christensen et al. (2006) up to dimension $d = 196$ but performs better when $d = 400$.

As far as the update of the principal components is concerned, someone could use a MALA proposal with an adaptive tuning as we did in PC-RWM. When we actually used a MALA proposal with adaptive tuning, in order to assess the effect of k on the performance of PC-MALA (Figures 4.3-4.5 and Figure 4.7), we found that for the chosen

optimal number of principal components the ESS obtained from the adaptive and non adaptive MALA proposal were similar. However, the use of an adaptive proposal eliminates the need of careful tuning by trial and error. We also tried the use of a truncated MALA proposal (Roberts & Tweedie (1996)) when updating the principal components. However, in our examples it seems that it was not essential since the results obtained were very similar to those presented in Section 4.3.2.

4.5 Appendix

4.5.1 Analytic form of $v_i^{\mathbb{E}}$

In the following we derive the exact form of $v_i^{\mathbb{E}}$,

$$\begin{aligned} v_i^{\mathbb{E}} &= \mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] - \mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right) \\ &= \mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right]\right) + \mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right) - 2\mathrm{Cov}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right], \mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right) \end{aligned}$$

Let $\boldsymbol{S} \sim \mathrm{MVN}\left(\boldsymbol{0}, \boldsymbol{R}\right)$. Denote by $\boldsymbol{R}_{-i}$ the matrix $\boldsymbol{R}$ reduced by the $i-$th row and column and $\boldsymbol{r}_i$ the $i-$th column with the $i-$th element removed. Then the following hold,

$$\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] = \boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\boldsymbol{S}_{-i},$$

$$\begin{aligned} \mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right]\right) &= \left(\boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\right)\boldsymbol{R}_{-\boldsymbol{i}}\left(\boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\right)^{'} \\ &= \boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\boldsymbol{r}_i. \end{aligned} \tag{4.5.1}$$

Let $\tilde{\boldsymbol{P}}$ be the first k principal components of $\boldsymbol{S}$. Then,

$$\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right] = \tilde{\boldsymbol{L}}_{[i,\,]}\tilde{\boldsymbol{P}},$$

$$\begin{aligned}\mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right) &= \tilde{\boldsymbol{L}}_{[i,\,]}\mathbb{V}\mathrm{ar}\left[\tilde{\boldsymbol{P}}\right]\tilde{\boldsymbol{L}}_{[i,\,]}^{'} \\ &= \tilde{\boldsymbol{L}}_{[i,\,]}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{L}}_{[i,\,]}^{'}. \end{aligned} \tag{4.5.2}$$

Finally, consider that since $\tilde{\boldsymbol{P}} = \tilde{\boldsymbol{L}}^{'}\boldsymbol{S}$,

$$\begin{aligned}\mathrm{Cov}\left(\mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right],\mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right]\right) &= \mathrm{Cov}\left(\boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\boldsymbol{S}_{-i},\ \tilde{\boldsymbol{L}}_{[i,\,]}\tilde{\boldsymbol{P}}\right) \\ &= \left(\boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\right)\mathrm{Cov}\left(\boldsymbol{S}_{-i},\boldsymbol{S}\right)\left(\tilde{\boldsymbol{L}}_{[i,\,]}\tilde{\boldsymbol{L}}^{'}\right)^{'} \\ &= \left(\boldsymbol{r}_i^{'}\boldsymbol{R}_{-i}^{-1}\right)\boldsymbol{R}_{[-i,\,]}\left(\tilde{\boldsymbol{L}}_{[i,\,]}\tilde{\boldsymbol{L}}^{'}\right)^{'} \end{aligned} \tag{4.5.3}$$

Therefore combining the expressions (4.5.1) – (4.5.3) we obtain the exact form of $v_i^{\mathbb{E}}$.

4.5.2 Proof of Proposition 4.2.1

Proof. Let,

$$A = \mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] - \mathbb{E}\left[S_i|\tilde{\boldsymbol{P}}\right], \tag{4.5.4}$$

and consider

$$\mathbb{V}\mathrm{ar}\left[A\right] = \mathbb{E}\left(\mathbb{V}\mathrm{ar}\left[A|\boldsymbol{S}_{-i}\right]\right) + \mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[A|\boldsymbol{S}_{-i}\right]\right). \tag{4.5.5}$$

If $k = d$ then, $A = \mathbb{E}\left[S_i|\boldsymbol{S}_{-i}\right] - S_i$ and therefore, $\mathbb{E}\left[A|\boldsymbol{S}_{-i}\right] = 0$, $\mathbb{V}\mathrm{ar}\left(\mathbb{E}\left[A|\boldsymbol{S}_{-i}\right]\right) = 0$. Hence,

$$\mathbb{V}\mathrm{ar}\left[A\right] = \mathbb{E}\left(\mathbb{V}\mathrm{ar}\left[A|\boldsymbol{S}_{-i}\right]\right) = \mathbb{E}\left(\mathbb{V}\mathrm{ar}\left[S_i|\boldsymbol{S}_{-i}\right]\right) = \mathbb{V}\mathrm{ar}\left[S_i|\boldsymbol{S}_{-i}\right], \tag{4.5.6}$$

where the last equation holds because $\mathbb{V}\mathrm{ar}\left[S_i|\boldsymbol{S}_{-i}\right]$ does not depend on $\boldsymbol{S}_{-i}$. □

4.5.3 Assessment of convergence for the U-PC algorithm

We choose at random three of the rejected scenarios, one from each dimension, and investigate whether there exists any striking feature that leads to such rejections. The chosen scenarios are ($d = 25$, $\boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b), ($d = 49$, $\boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 3, \phi = 10$, dataset c) and ($d = 100$, $\boldsymbol{\mu}_\eta = \log(100), \sigma^2 = 1, \phi = 10$, dataset a).

For each scenario we found that some of the observed marginal KS statistics exceed the 95% quantile of their marginal distribution giving rise to a large value for our observed statistic $K := \sum_{i=1}^{d} KS_i$. In Figures 4.8-4.10 we see that for the first scenario there are 3, for the second scenario 5 and finally for the third scenario 7 such cases. For the first and second scenario, eliminating from the chains the components that contribute more to our statistic K, i.e., components η_{23} and η_{49} respectively, leads to failure of rejecting the null hypothesis whereas for the third scenario more than one components should be eliminated for convergence to be rejected.

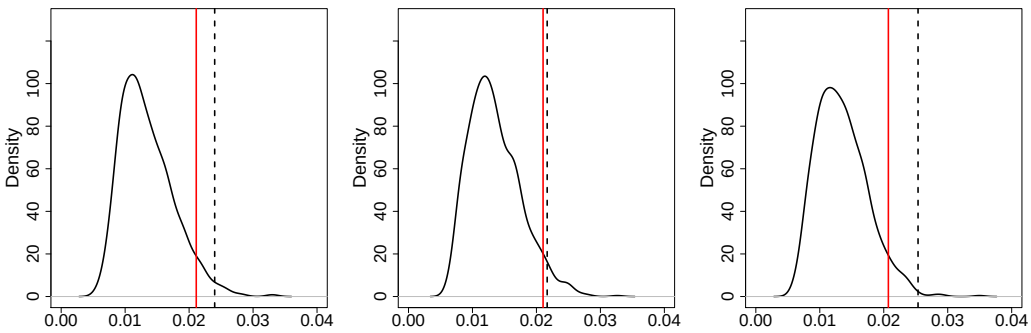

Figure 4.8: Density plots of $KS_{10}, KS_{11}, KS_{23}$ (left to right). The red lines indicate the 95% quantile and the black dashed line the observed value of the statistic. Scenario ($d = 25, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b).

In Figure 4.11 we display the posterior means and variances of the chains obtained from U-PC and the algorithm of Christensen et al. (2006) and illustrate with green colour the mean and variances of the components with high KS_i. From these plots, there does not

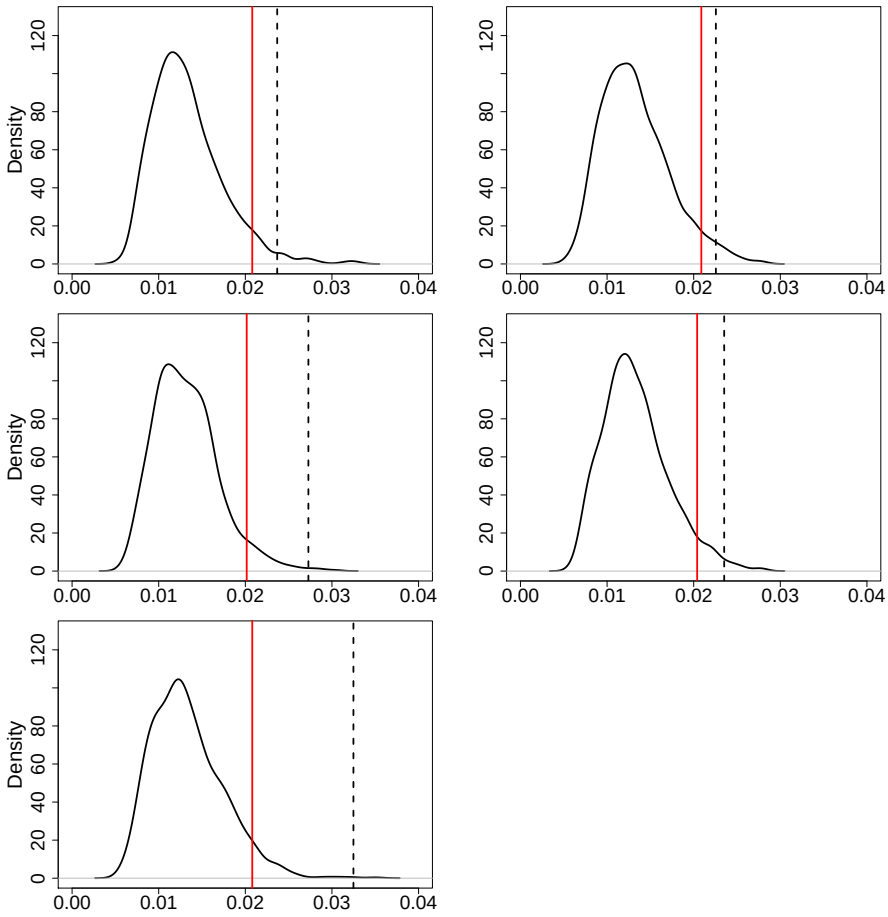

Figure 4.9: Density plots of $KS_7, KS_{12}, KS_{20}, KS_{48}, KS_{49}$ (left to right). The red lines indicate the 95% quantile and the black dashed line the observed value of the statistic. Scenario ($d = 49, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 3, \phi = 10$, dataset c).

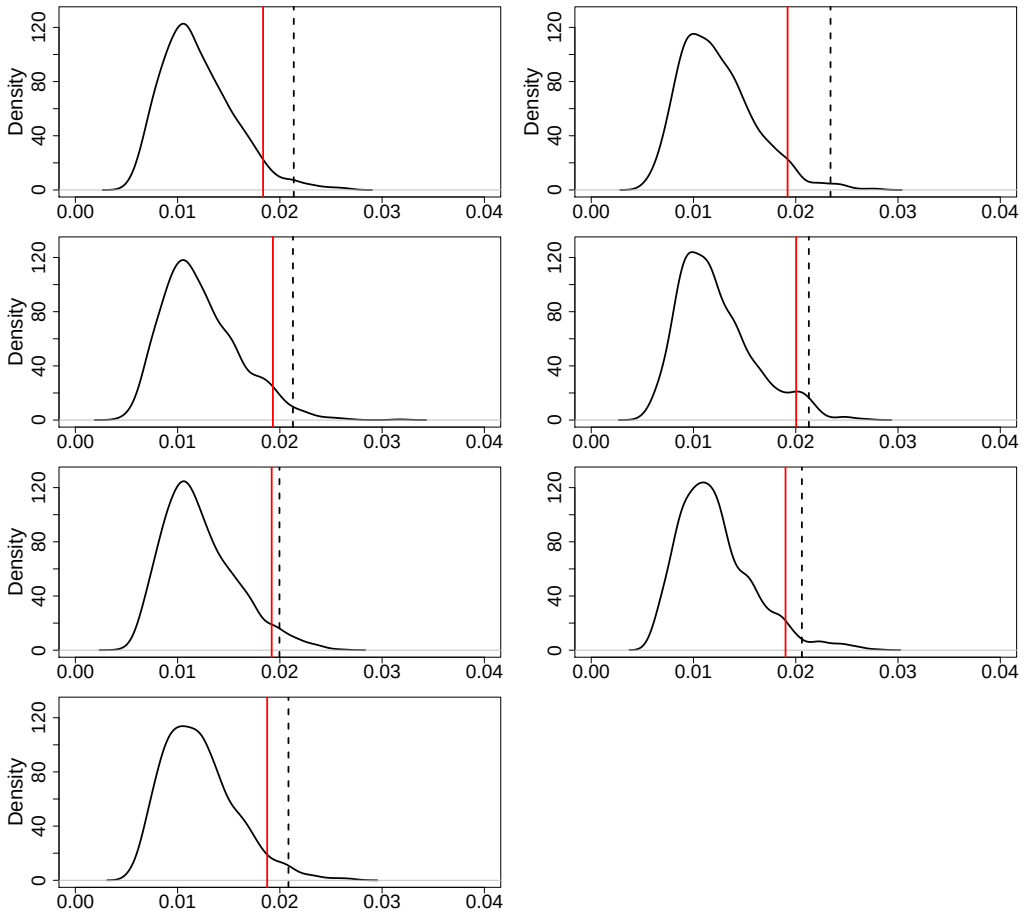

Figure 4.10: Density plots of KS_3, KS_8, KS_{33}, KS_{41}, KS_{71}, KS_{90}, KS_{93}(left to right and top to bottom). The red lines indicate the 95% quantile and the black dashed line the observed value of the statistic. Scenario ($d = 100, \boldsymbol{\mu}_\eta = \log(100), \sigma^2 = 1, \phi = 10$, dataset a).

appear to be a problem with estimation of the posterior mean and variance between the two algorithms. In the following, we investigate why, according to the KS tests, the two algorithms result in different marginal distributions for those components. We choose the QQ-plot as a graphical method for comparing the marginal distributions from the two algorithms and these are displayed in Figures 4.12-4.14. As we see there are cases where one of the two algorithms might explore better either one or both tails of the distribution. However, we can not find a systematic pattern appearing in every QQ-plot. In some of the plots though we see that U-PC explores the right tail of the distribution at least as well as the algorithm of Christensen et al. (2006) but sometimes might fail to go further out in the left tail.

Figures 4.15-4.17 show the traceplots of the chains obtained from the two algorithms. From the traceplots we would expect to identify any existing trend and generally differences in the mixing between the two algorithms. In Figure 4.15 we see that the traceplots of U-PC for the components η_{10} and η_{11} exhibit an increasing trend in the last 1000 iterations. A similar pattern can be seen in Figure 4.16 for component η_7. Finally, looking at the traceplot of η_{90} in Figure 4.17 we notice that there is more variability in the traceplot of Christensen et al. (2006) than in that of U-PC. We finally examine whether the rejection is due to different levels of correlation within the two samples. Figures 4.18-4.20 show the autocorrelation plots of the chains for those rejected scenarios along with 95% confidence intervals. For both chains the autocorrelation mainly lies within the 95% confidence interval showing that they have been appropriately thinned in order to be considered a white noise. Moreover, exceedences of the bands happen equally often for both algorithms. The only exception seems to be component η_{12} in Figure 4.19 where for some lags the autocorrelation for U-PC falls outside the 95% interval whereas that

of Christensen et al. (2006) always lies within the bands.

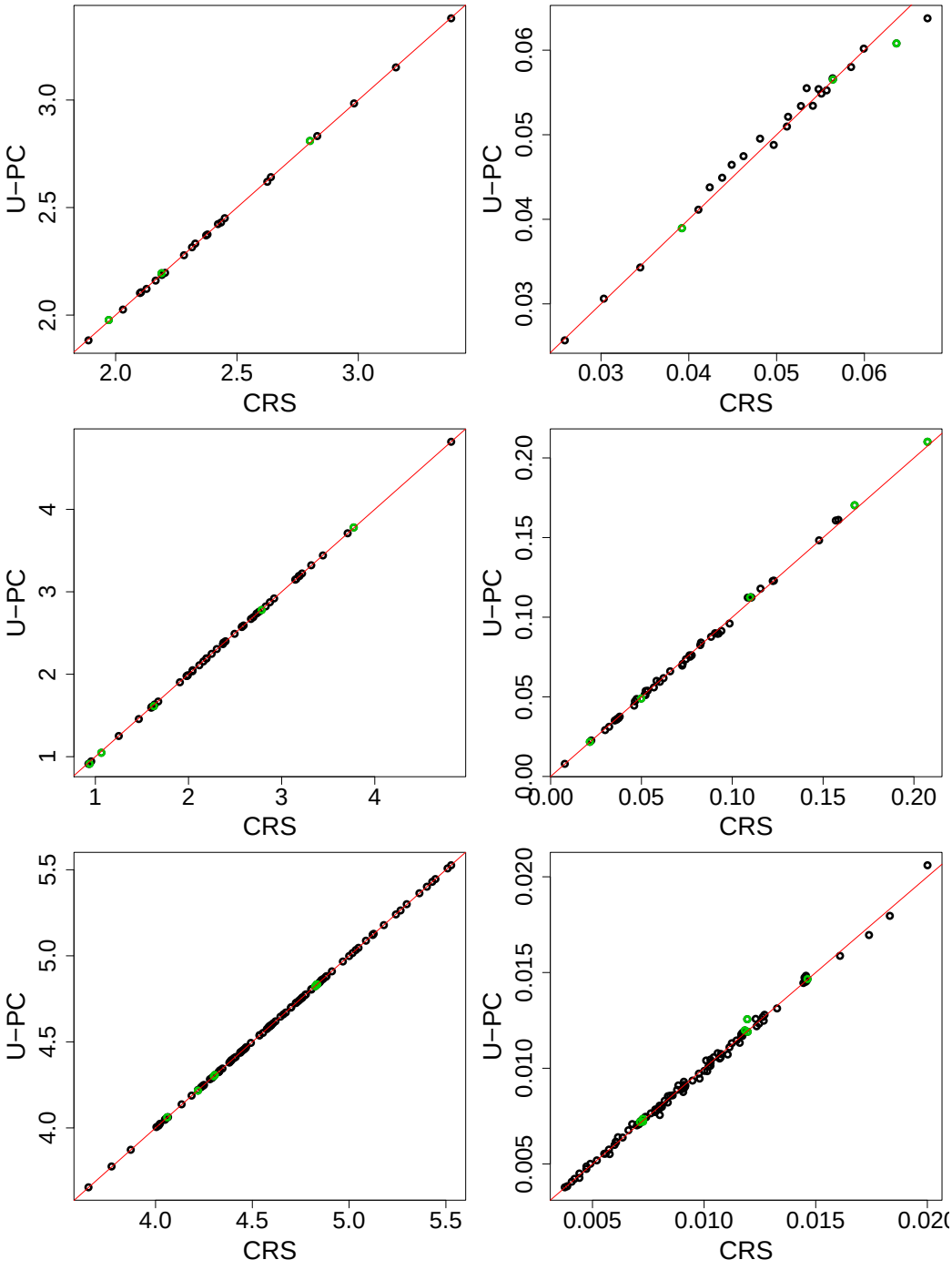

Figure 4.11: Plots of posterior means (left) and variances (right) of $\boldsymbol{\eta}$. x-axis: Christensen et al. (2006) algorithm, y-axis: U-PC algorithm. Top to Bottom: scenario ($d = 25$, $\boldsymbol{\mu}_\eta = \log(10)$, $\sigma^2 = 1$, $\phi = 10$, dataset b), scenario ($d = 49$, $\boldsymbol{\mu}_\eta = \log(10)$, $\sigma^2 = 3$, $\phi = 10$, dataset c), scenario ($d = 100$, $\boldsymbol{\mu}_\eta = \log(100)$, $\sigma^2 = 1$, $\phi = 10$, dataset a)

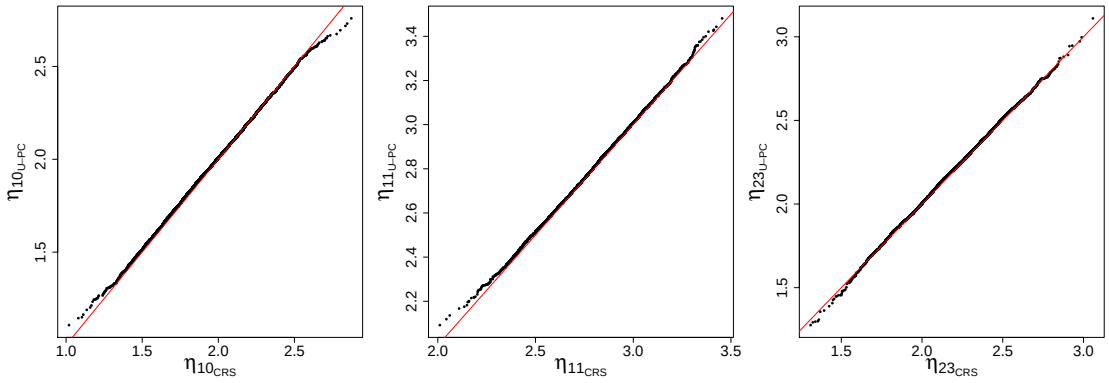

Figure 4.12: QQ-plots of η_{10}, η_{11}, η_{23}. x-axis: Algorithm of Christensen et al. (2006), y-axis: U-PC algorithm. Scenario ($d = 25, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b).

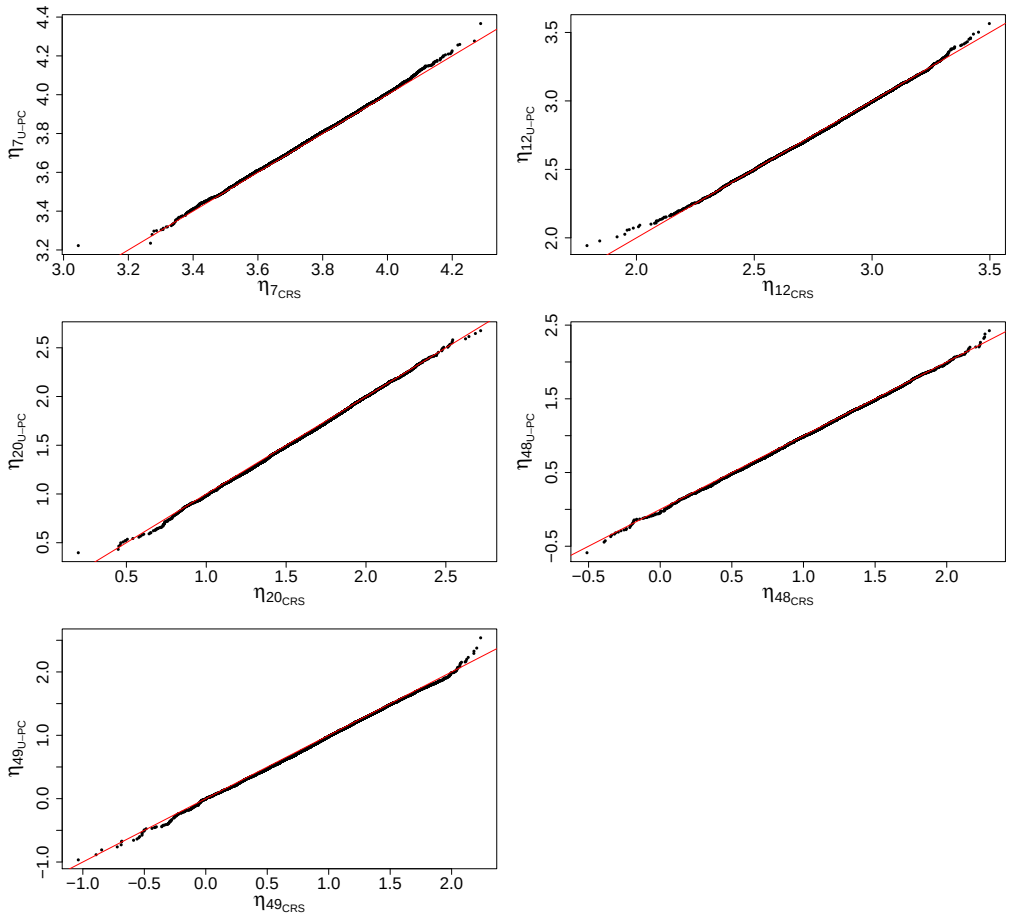

Figure 4.13: QQ-plots of η_{10}, η_{11}, η_{23}. x-axis: Algorithm of Christensen et al. (2006), y-axis: U-PC algorithm. Scenario ($d = 19, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b).

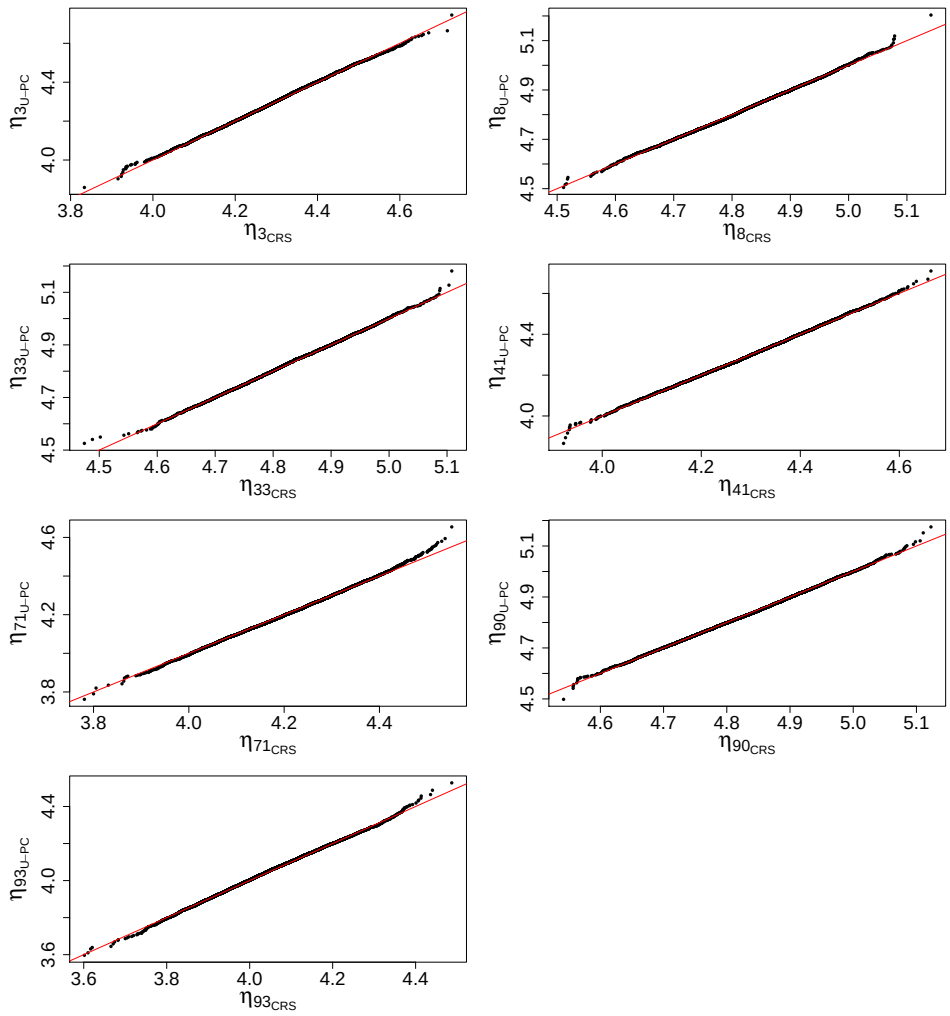

Figure 4.14: QQ-plots of η_{10}, η_{11}, η_{23}. x-axis: Algorithm of Christensen et al. (2006), y-axis: U-PC algorithm. Scenario ($d = 100, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b).

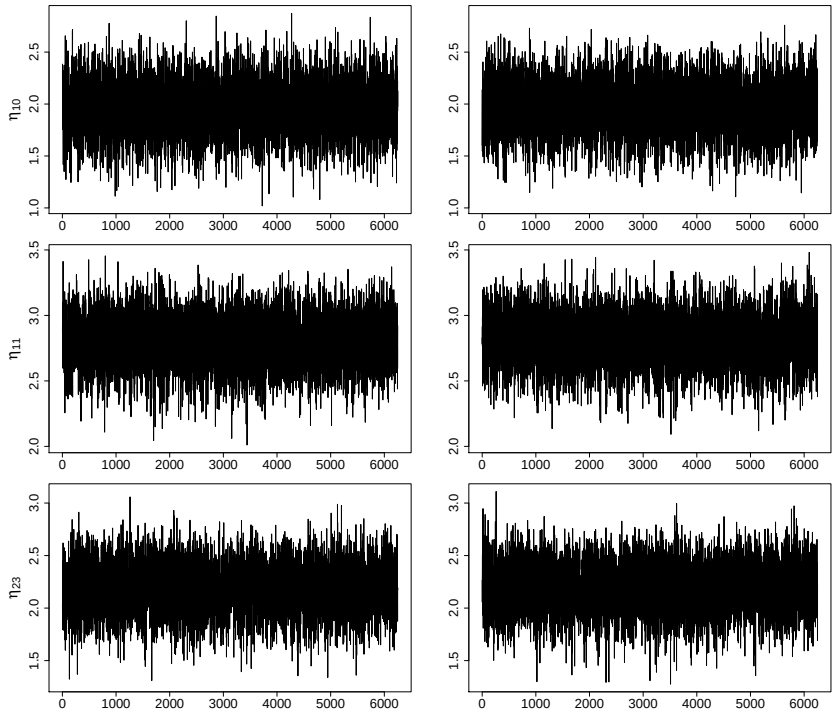

Figure 4.15: Traceplots for η_{10}, η_{11}, η_{23} (top to bottom) obtained from algorithm of Christensen et al. (2006) (left) and U-PC algorithm (right). Scenario ($d = 25, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b).

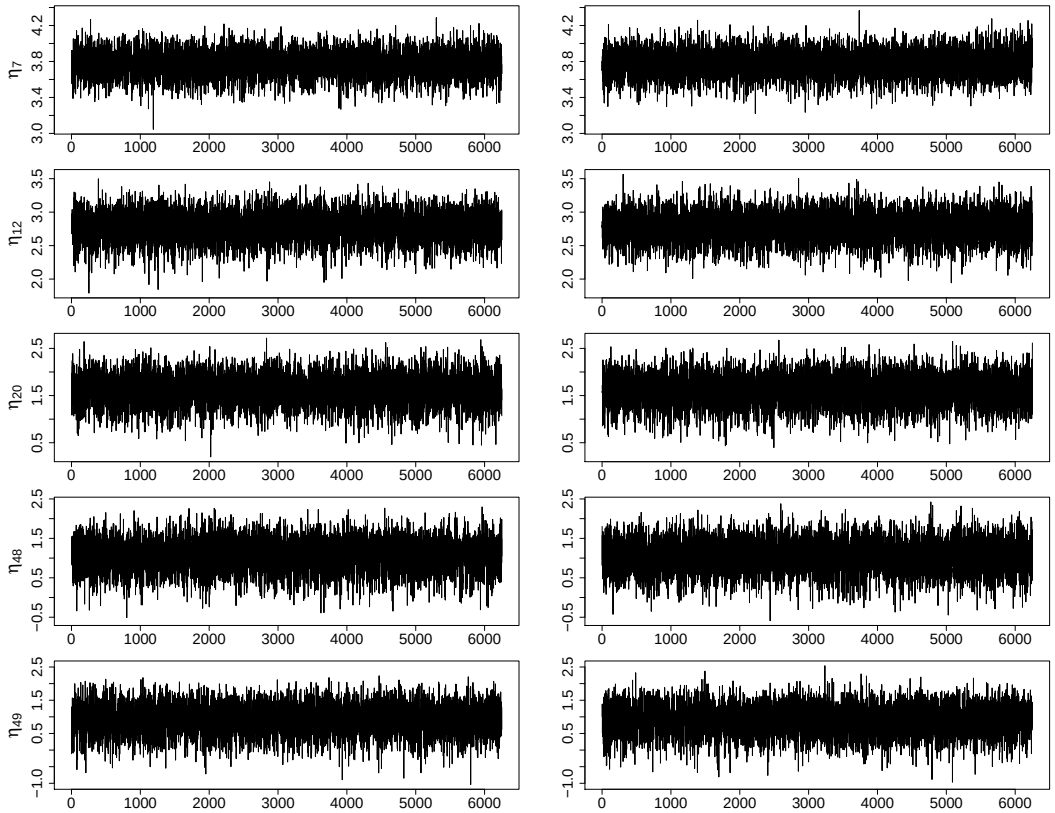

Figure 4.16: Traceplots for $\eta_7, \eta_{12}, \eta_{20}, \eta_{48}, \eta_{49}$ (top to bottom) obtained from algorithm of Christensen et al. (2006) (left) and U-PC algorithm (right). Scenario ($d = 49, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 3, \phi = 10$, dataset c).

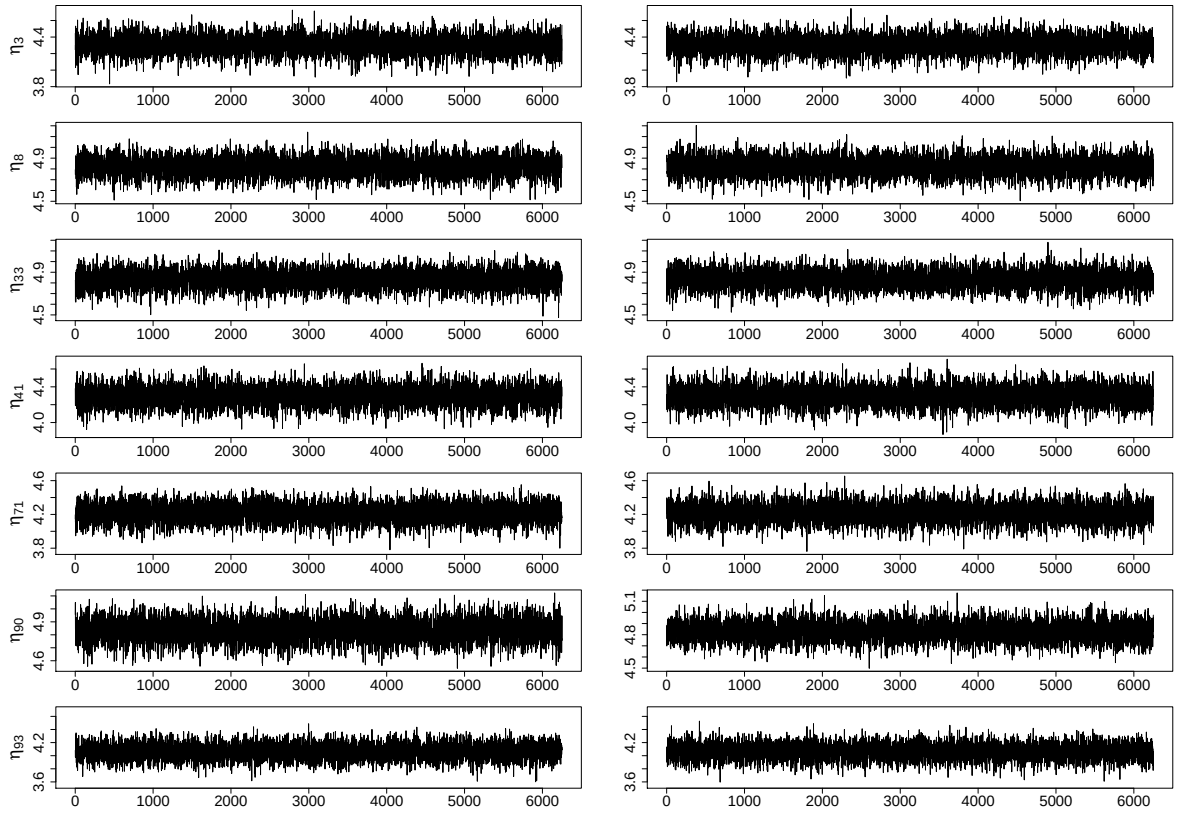

Figure 4.17: Traceplots for η_3, η_8, η_{33}, η_{41}, η_{71}, η_{90}, η_{93} (top to bottom) obtained from algorithm of Christensen et al. (2006) (left) and U-PC algorithm (right). Scenario ($d = 100, \boldsymbol{\mu}_\eta = \log(100), \sigma^2 = 1, \phi = 10$, dataset a).

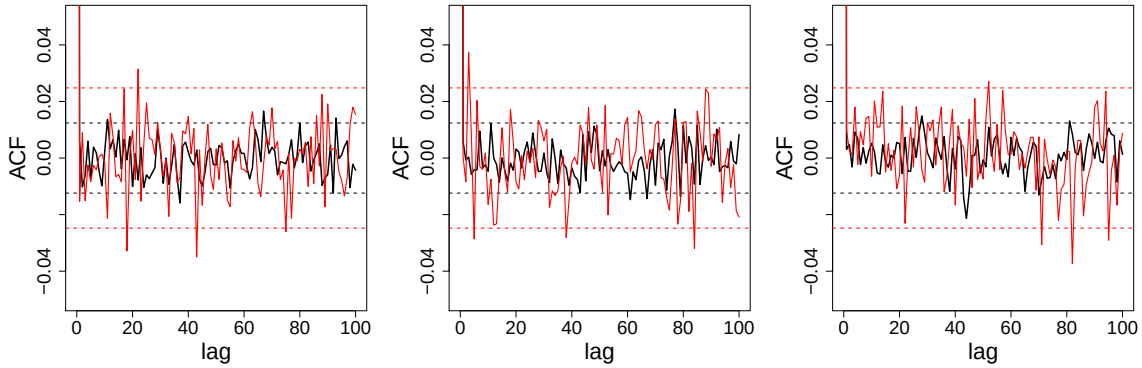

Figure 4.18: Autocorrelation plots of η_{10}, η_{11}, η_{23} (left to right and top to bottom). The black line corresponds to the ACF for the algorithm of Christensen et al. (2006) and the red line to U-PC. The dashed lines indicate the upper and lower bounds of a 95% confidence interval. Scenario ($d = 25, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 1, \phi = 10$, dataset b)

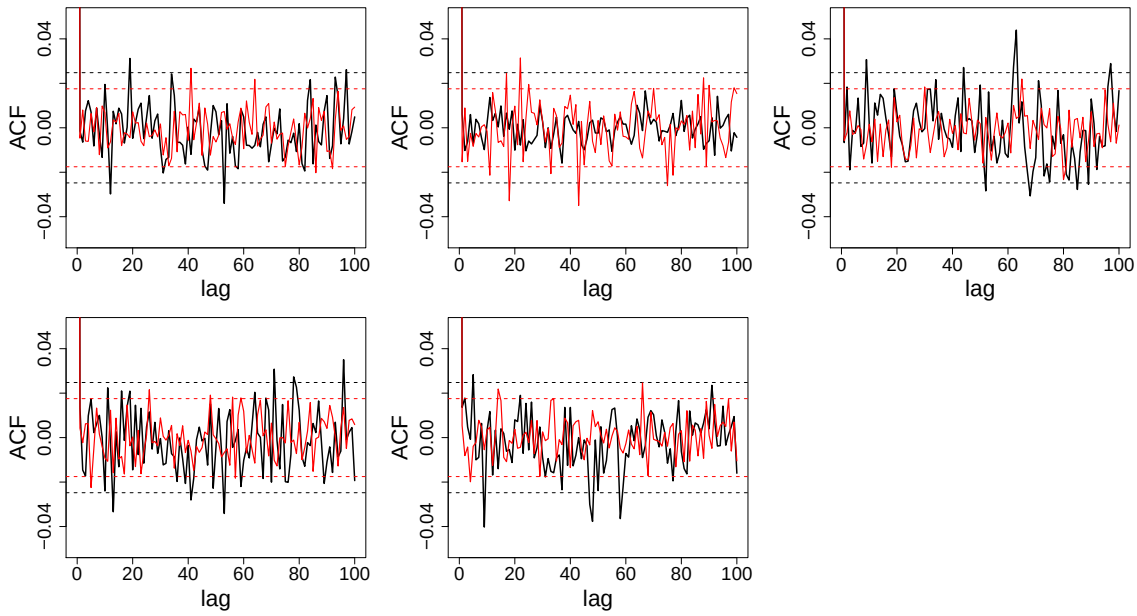

Figure 4.19: Autocorrelation plots of $\eta_7, \eta_{12}, \eta_{20}, \eta_{48}, \eta_{49}$ (left to right and top to bottom). The black line corresponds to the ACF for the algorithm of Christensen et al. (2006) and the red line to U-PC. The dashed lines indicate the upper and lower bounds of a 95% confidence interval. Scenario ($d = 49, \boldsymbol{\mu}_\eta = \log(10), \sigma^2 = 3, \phi = 10$, dataset c)

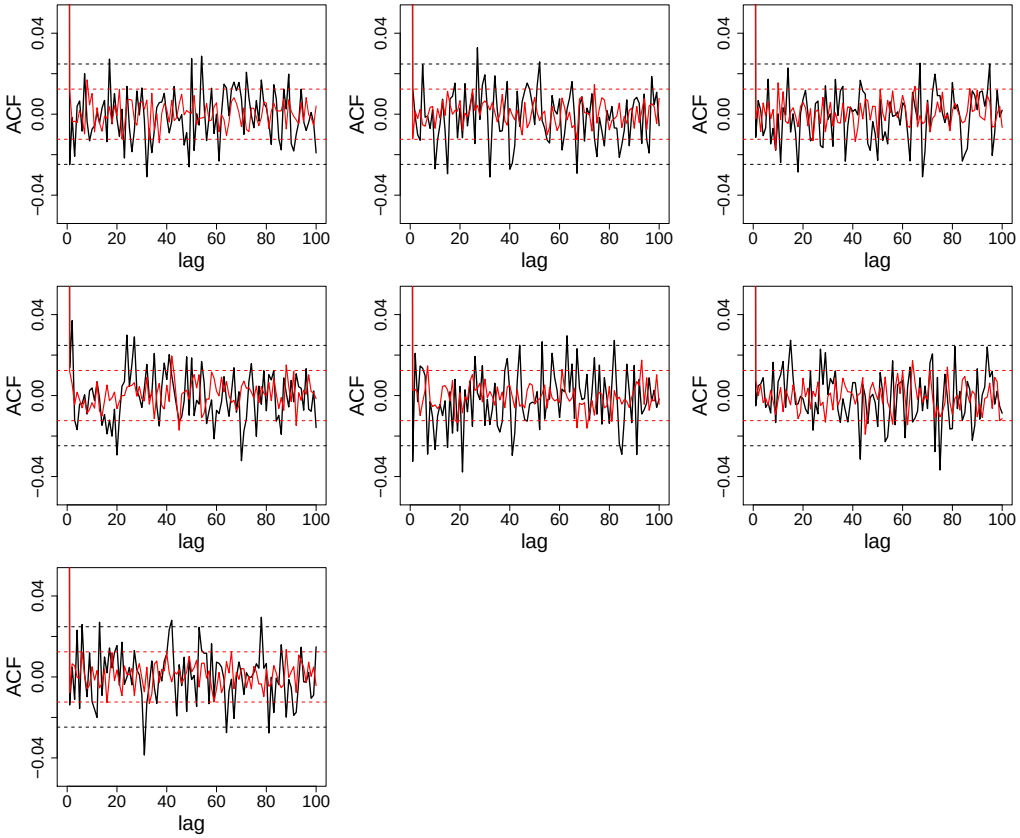

Figure 4.20: Autocorrelation plots of η_3, η_8, η_{33}, η_{41}, η_{71}, η_{90}, η_{93} (left to right and top to bottom).The black line corresponds to the ACF for the algorithm of Christensen et al. (2006) and the red line to U-PC. The dashed lines indicate the upper and lower bounds of a 95% confidence interval. Scenario ($d = 100, \boldsymbol{\mu}_\eta = \log(100), \sigma^2 = 1, \phi = 10$, dataset a).

CHAPTER 5

Discussion

The scope of the present Thesis has been to construct new MCMC samplers for inference in the case of the generalised linear spatial model (GLSM). In particular, our focus has been placed on developing an efficient proposal distribution for updating the latent process of the model conditional on the model parameters. Under the framework of the GLSM the latent process is modelled as a Gaussian process and this has motivated of our approach throughout this Thesis. Throughout this study, our approach has been motivated by the simple fact that the latent process of the model is assumed to be Gaussian. Therefore, our strategy for constructing such proposals has been based on the creation of Gaussian approximations of the posterior distribution of the latent process given the model parameters. The performance of all the constructed samplers was assessed in terms of the ESS scaled by the CPU time required and compared against the simplified MMALA of Girolami & Calderhead (2011) and the algorithm suggested by Christensen et al. (2006) through extensive simulation studies.

Our initial approach, in Chapter 3, has been to employ a transformation of the data in order to obtain a single Gaussian approximation of the likelihood and consequently of the target distribution. In turn, a heavy tailed version of this approximation was used as the proposal distribution in an Independence sampler that attempted to update the latent process in a single block. Our finding was that in the vast majority of the cases that were studied such an approach has not been successful since the target is far from Gaussian and therefore a single global Gaussian approximation cannot capture its shape. As the dimension of the latent process increases this problem becomes more prominent and in combination with the choice of the specific updating mechanism, the Independence sampler, led to poorly performing MCMC schemes. The performance of the constructed samplers depends heavily on the values of the model parameters and the informativeness of the data since these two factors determine the posterior correlation of the latent process and therefore the shape of the target distribution. For instance, our simplest algorithm L1, where the link function is used to transform the data, can perform sufficiently well the data is very informative, i.e., when the prior mean $\boldsymbol{\mu}_\eta$ is large, provided that σ^2 is small. Therefore an observed sample of high measurements, $\boldsymbol{y}$, could provide an indication that L1 could be preferred over the more complicated and computationally costly alternatives.

In Chapter 4 we took a different approach and placed our focus on creating an efficient MCMC scheme that performs individual updates on each component of the latent process. The motivation behind the developed MCMC scheme was to mimic a Gibbs sampler in terms of acceptance rates while overcoming its mixing problems in the case of correlated multi-dimensional targets. This was achieved by making use of the principal components obtained from the prior correlation matrix of the latent process. The key

finding for the development of this approach was that for a given correlation function we could define a number, $k << d$, of principal components, $\tilde{\boldsymbol{p}}$, for which the Gaussian distribution of $S_i|\tilde{\boldsymbol{p}}$ could be used to approximate that of $S_i|\boldsymbol{S}_{-i}$. This was subsequently combined with the Poisson likelihood and the Laplace approximation of the distribution of $S_i|\tilde{\boldsymbol{p}}, \boldsymbol{y}$ was derived using its mode and curvature at the mode. This was used as the proposal distribution in a MCMC scheme that within an iteration updated each s_i conditionally on the current value of $\tilde{\boldsymbol{p}}$. Although the update of a single s_i also caused an update of $\tilde{\boldsymbol{p}}$ this was not enough in order to obtain a well mixing chain. The addition of a single block update of $\tilde{\boldsymbol{p}}$ through a MALA update significantly improved the mixing of the chain and gave rise to, PC-MALA, our final MCMC scheme. Our simulation studies showed that PC-MALA had a robust efficient performance across many scenarios of parameter values and dimensions and that always performed at least as well as the algorithm of Christensen et al. (2006). We have also provided with a simple diagnostic that aids to define the number k of principal components that should be conditioned on and have also shown empirically, through simulation studies, that such a choice appears to be close to optimal irrespective of the dimension and parameter values. Under the framework of fixed parameter values, We have provided with ways to reduce the computational cost of the iterative part of the algorithm from $O(d^3)$ to $O(d^2)$ but we still have to deal with a $O(d^3)$ cost from the spectral decomposition of the prior correlation matrix $\boldsymbol{R}$.

Throughout the Thesis, we have made certain assumptions in order to simplify the working setting. For instance, in reality all model parameters would be unknown and therefore a full MCMC scheme would be used in order to draw inferences. As already discussed, such schemes would usually alternate between updates of the model parameters given

the latent process and updates of the latent process given the current values of the parameters. Since our focus has been placed on the development of efficient proposals for the later we have assumed that the model parameters are fixed in our simulation studies. Additionally, all of the MCMC schemes proposed in this Thesis can accommodate the presence of the nugget effect. In what presented however, for ease of illustration, we have clearly assumed that the nugget effect is equal to zero which in many cases might not realistic. Even if the measurement process was extremely accurate so as to provide us with zero measurement error the nugget effect would still illustrate micro-scale variation, i.e., variation in distances finner than the minimum sampling distance.

As far as the Independence Sampler, L1, of Chapter 3 is concerned in the case of a full MCMC scheme where all parameters get updated at each iteration, both the mean and the variance-covariance matrix of the proposal distribution would have to be calculated. Since the calculation of both these expressions involve the the mean of the process and the inversion of the d-dimensional matrix $\left[\sigma^2(\boldsymbol{F}\boldsymbol{\Sigma}_\beta\boldsymbol{F}^* + \boldsymbol{R}) + \boldsymbol{\Sigma}^*\right]$ the additional computational cost will be of order $O(d^3)$. In the case of the algorithms presented in Chapter 4 and especially the PC-RWM and PC-MALA the main computational burden is the spectral decomposition that has to take place before the update of the components of the latent process. In particular, in order to calculate the principal components we have to obtain the eigenvectors and eigenvalues of the prior correlation matrix $\boldsymbol{R}$. Since it is the correlation matrix that we are dealing with, and not the covariance, any update of σ^2 would not have any effect on it since the eigenvectors and eigenvalues of $\boldsymbol{R}$ would be unchanged. In general, the update of parameters $\boldsymbol{\beta}$ and σ^2 does not have a considerable effect on the computational cost of the algorithm. For instance, both parameters are involved in the calculation of the bounds required for the mode of our proposal. However,

these already have to be calculated at each iteration since they are conditional on the principal components. They are also needed for the calculation of target in the acceptance probability which will have to be computed either way though. It is mainly the update of parameter ϕ that increases the computational cost of our approach through the spectral decomposition. To overcome this issue there are two straightforward alternatives that could be adopted. One solution would be to update ϕ only once every $O(d)$ iterations reducing the average computational cost of each iteration to $O(d^2)$. Alternatively, a discrete prior for ϕ could be adopted so that for example, the prior covariance matrix, the spectral decomposition and any associated quantities can be computed and stored in advance. However, each of these options entails different issues that should be considered. For instance, the former could lead to a slowly mixing chain whereas the later would require a considerable amount of storage. If a nugget effect, τ^2, was also to be included in the model the extra computational cost induced would be negligible compared to that induced by the update of ϕ since the spectral decomposition would not have to be conducted every time. To see this, first of all consider the, scaled, reparametrised prior covariance matrix $\boldsymbol{C} = (R + \nu \boldsymbol{I})$ where $\nu = \tau^2/\sigma^2$ and assume that ϕ and ν (or τ^2) are updated sequentially. Suppose that we are at the i-th iteration of the algorithm and currently have the eigenvalues and the matrix of eigenvectors of $\boldsymbol{R}$ and are denoted by $(\lambda_1^{(i)}, ..., \lambda_d^{(i)})$ and $\boldsymbol{L}$ respectively. Then the eigenvectors of $\boldsymbol{C}$ are also $\boldsymbol{L}$ but its eigenvalues are $(\lambda_1^{(i)} + \nu^{(i)}, ..., \lambda_d^{(i)} + \nu^{(i)})$. Consider now that ϕ is updated to $\phi^{(i+1)}$ so that $\boldsymbol{L}^{(i+1)}$ and $(\lambda_1^{(i+1)} + \nu^{(i)}, ..., \lambda_d^{(i+1)} + \nu^{(i)})$ are obtained. If now ν is updated to $\nu^{(i+1)}$ we only have to increment the eigenvalues by $\nu^{(i+1)}$, i.e., $(\lambda_1^{(i+1)} + \nu^{(i+1)}, ..., \lambda_d^{(i+1)} + \nu^{(i+1)})$. Therefore, the need to spectrally decompose $\boldsymbol{R}$ every time that the nugget effect gets updated is overcome. Finally, the alternative would be to assign a joint discrete prior to (ϕ, ν) (or (ϕ, τ)) and precompute the spectral decomposition for the possible pairs of (ϕ, τ) in

advance.

The immediate next step would be to assess the performance of our approach in the case of a full MCMC and for dimensions higher than $d = 400$. Moreover, we may have constrained ourselves to the case of the Poisson GLSM but the motivating ideas illustrated in this study can be extended to any GLSM. For instance, it would be interesting to assess the performance of the suggested algorithms in the case of the widely used logistic GLSM where the response variable follows the binomial distribution. The algorithm L1 from Chapter 3 is already generalised for any link function therefore its implementation would be straightforward. For the algorithm, PC-MALA the exact form of our approximation $\tilde{\pi}(\eta_i|y_i)$ will now be different, due to the binomial likelihood, therefore the exact form of the mode will also differ. Hence, some further work should be carried out in order to define bounds for the maximisation of $\tilde{\pi}(\eta_i|y_i)$, if possible. Moreover similar shortcuts for the calculation of the acceptance probability could also be obtained as in the case of the Poisson GLSM.

In the following, we discuss some additional directions of further work that should be followed and mainly focus on the approaches of Chapter 4 and especially the PC-MALA. We have mainly assessed our algorithms under the use of the exponential correlation function and also the case of the Matérn with $\kappa = 1.5$. Our finding was that as κ increases the number of principal components on wich we should condition also increases. As discussed in the previous Chapter, we believe that this is related to the differentiability of the correlation function at the origin. With increasing κ the underlying process becomes smoother and the dependence remains at very high levels for longer distances. A first attempt to empirically study the relationship between κ and k in the case of the Matérn family with κ up to $\kappa = 5.5$ was not successful. The correlation matrix $\boldsymbol{R}$ was

nearly singular and our derived diagnostic of Section 4.2.3, for the choice of k, could not be reproduced due to precision instabilities. Therefore, further research should be conducted to properly understand how the smoothness of the underlying process could be exploited in order to define the optimal number of principal components that should be conditioned on. Such a study could of course extend outside the Matérn family of correlation functions.

Some preliminary investigations were conducted to assess the applicability of our diagnostics and the performance of PC-MALA in the case of an irregular grid where the sampling locations were sampled uniformly in the square $\left\{1, 2, ..., \sqrt{d}\right\}^2$, for $d = \{25, 49, 100\}$. Under the use of the exponential correlation function and the same scenarios of parameter values as used in the presented simulation studies our findings were encouraging. First of all, we found that a fixed number of principal components k could still be defined as a function of d, using the diagnostic provided in Section 4.2.3. Also, the algorithm PC-MALA was found to perform at least as well as that of Christensen et al. (2006) in terms of minimum ESS. Further such explorations should of course be carried out under both different correlation functions and different sampling designs as we expect that the distribution of the sampling locations over the grid would impact the number of principal components that should be chosen.

From what discussed so far, there is definitely room for improvement and further work to be carried out in order for the suggested approach of Chapter 4 to be directly applicable under a general framework. However, we believe that the combination of the suggested approach and the reparametrisation of by Christensen et al. (2006) can give rise to an efficient and robust MCMC for inference in geostatistical problems.

Bibliography

Adler, R. J. & Taylor, J. E. (2007), *Random Fields and Geometry*, Springer.

Andrieu, C. & Thoms, J. (2008), 'A tutorial on adaptive MCMC', *Stat. Comput* **18**, 343–373.

Anscombe, F. J. (1948), 'The transformation of poisson, binomial and negative binomial data', *Biometrika* **35**, 246–254.

Boone, E. L., Merrick, J. R. W. & Krachey, M. J. (2014), 'A Hellinger distance approach to MCMC diagnostics', *Journal of Statistical Computation and Simulation* **84**.

Box, G. E. P. & Cox, D. R. (1964), 'An analysis of transformations (with discussion)', *Journal of the Royal Statistical Society, Series B* **26**, 211–152.

Breslow, N. E. & Clayton, D. G. (1993), 'Approximate inference in generalised linear mixed models', *Journal of the American Statistical Association* **88**, 9–25.

Brooks, S., Gelman, A., Jones, G. L. & Xiao-Li, M., eds (2011), *Handbook of Markov Chain Monte Carlo*, Chapman & Hall/ CRC.

Brooks, S. P. & Gelman, A. (1998*a*), 'General methods for monitoring convergence of iterative simulations.', *Journal of Computational and Graphical Statistics* **7**, 434–455.

Brooks, S. P. & Gelman, A. (1998*b*), 'General methods for monitoring convergence of iterative simulations', *Journal of Computational and Graphical Statistics* **7**, 434–455.

Brooks, S. P., Giudici, P. & Philippe, A. (2003), 'Nonparametric convergence assessment for MCMC model selection', *Journal of Computational and Graphical Statistics* **12**, 1–22.

Casella, G. & Berger, R. L. (1990), *Statistical inference*, Statistics/Probability Series, Wadsworth & Brooks/Cole Advanced Books & Software, Pacific Grove, CA.

Chilès, D. P. & Delfiner, P. (1999), *Geostatistics: Modelling Spatial Uncertainty*, Wiley, New York.

Christensen, O. F. (2004), 'Monte carlo maximum likelihood in model-based geostatistics', *Journal of Computational and Graphical Statistics* **13**, 702–718.

Christensen, O. F., Diggle, P. J. & Ribeiro, P. J. (2001), Analysing positive-valued spatial data: the transformed gaussian model, *in* 'GeoENV III - Geostatistics for Environmental Applications (eds. P. Monestiez, D. Allard and R. Froidevaux), Kluwer, Dordrecht, 287–298', Springer, Netherlands.

Christensen, O. F., Moller, J. & Waagepetersen, R. (2000), Analysis of spatial data using generalised linear mixed models and Langevin-type Markov Chain Monte Carlo, Technical report, Department of Mathematical Sciences, Aalborg University.

Christensen, O. F., Moller, J. & Waagepetersen, R. (2001), 'Geometric ergodicity of Metropolis Hastings algorithms for conditional simulation in generalised linear mixed models', *Methodology and Computing in Applied Probability* **3**, 309–327.

Christensen, O. F., Roberts, G. O. & Sköld, M. (2006), 'Robust Markov chain Monte Carlo methods for spatial generalized linear mixed models', *J. Comput. Graph. Statist.* **15**(1), 1–17.

Christensen, O. F. & Waagepetersen, R. (2002), 'Bayesian prediction of spatial count data using generalised linear mixed models', *Biometrics* pp. 280–286.

Conover, W. J. (1999), *Practical Nonparametric Statistics*, New York, NY: John Wiley & sons.

Davison, A. & Hinkley, D. (1997), *Bootstrap Methods and their Application*, Cambridge University Press.

Diggle, P., J., Paulo, J. & Ribeiro, J. (2007), *Model-based geostatistics*, Springer Series in Statistics, Springer, New York.

Diggle, P. J., Ribeiro, J. & Christensen, O. F. (2003), An introduction to model-based geostatistics, *in* 'Spatial statistics and computational methods (Aalborg, 2001)', Vol. 173 of *Lecture Notes in Statist.*, Springer, New York, pp. 43–86.

Diggle, P. J., Tawn, J. A. & Moyeed, R. A. (1998), 'Model-based geostatistics', *J. Roy. Statist. Soc. Ser. C* **47**(3), 299–350. With discussion and a reply by the authors.
URL: *http://dx.doi.org/10.1111/1467-9876.00113*

Fearnhead, P., Giagos, V. & Sherlock, C. (2014), 'Inference for reaction networks using the Linear Noise Approximation.', *Biometrics* .

Galassi, M., Davies, J., Theiler, J., Gough, B., Jungman, G., Alken, P., Booth, M. & Rossi, F. (2010), *GNU Scientific Library Reference Manual*, 3rd edn, The GSL Team.

Gamerman, D. & Lopes, H. F. (2006), *Markov Chain Monte Carlo: Stochastic Simulations for Bayesian Inference*, Chapman & Hall/CRC.

Gelman, A. & Rubin, D. B. (1992), 'Inference from iterative simulation using multiple sequences', *Statistical Science* **7**, 457–511.

Geweke, J. (1992), Evaluating the accuracy of sampling-based approaches to the calculation of posterior moments (with discussion), *in* B. J. M., J. O. Berger, A. P. Dawid & S. A. F. M., eds, 'Bayesian Statistics 4', Oxford Univ. Press, pp. 169–193.

Gilks, W. R., Richardson, S. & Spiegelhalter, D. J., eds (1996), *Markov Chain Monte Carlo in Practice*, Chapman & Hall.

Gilks, W. R. & Roberts, G. O. (1996), Strategies for improving MCMC, *in* W. R. Gilks, S. Richardson & D. J. Spiegelhalter, eds, 'Markov Chain Monte Carlo in Practice', Chapman & Hall, chapter 6.

Giorgi, E., Sesay, S. S. S., Terlouw, D. J. & Diggle, P. J. (2015), 'Combining data from multiple spatially referenced prevalence surveys using generalized linear geostatistical models', *Journal of the Royal Statistical Society: Series A* **178**, 445464.

Girolami, M. J. & Calderhead, B. A. (2011), 'Riemann manifold Langevin and Hamiltonian Monte Carlo methods', *J. Roy. Statist. Soc. Ser. B* **73**, 123–214. With discussion and a reply by the authors.

Haran, M. & Tierney, L. (2012), 'On automating Markov chain Monte Carlo for a class of spatial models', *ArXiv e-prints* .

Hastings, W. K. (1970), 'Monte carlo sampling methods using markov chains and their applications', *Biometrika* .

Hills, S. E. & Smith, A. F. M. (1992), Parameterization issues in Bayesian inference, *in* 'Bayesian statistics, 4. (Spain, 1991)', New York, NY: Clarendon Press.

Hughes, J. & Haran, M. (2013), 'Dimension reduction and alleviation of confounding for spatial generalised linear mixed models', *Journal of Royal Statistical Society Series B* **75**, 139–159.

Kolmogorov, A. N. (1933), 'Sulla determinazione empirica di una legge di distribuzione', *Giornale dell' Instituto Italiano degli Attuari* **4**, 83–91.

Krige, D. G. (1951), 'A statistical approach to some basic mine valuation problems on the Witwatersrand', *Journal of the chemical, Meattallurgical and Mining Society of South Africa* **52**, 119–139.

Lindgren, F., Rue, H. & Lindström, J. (2011), 'An explicit link between Gaussian fields and Gaussian markov random fields: the stochastic partial differential equation approach', *J. Roy. Statist. Soc B* **73**(4), 423–498.

Liu, J. (1996), 'Metropolized independent sampling with comparisons to rejection sampling and importance sampling', *Statistics and Computing* **6**(2), 113–119.

Mardia, K. V. & Watkins, A. J. (1989), 'On multimodality of the likelihood in the spatial linear model', *Biometrika* **76**, 289–296.

Matérn, B. (1960), Spatial variation, Technical report, Statens Skogsforsningsinstitut, Stockholm.

McCullagh, P. & Nelder, J. A. (1989), *Generalized linear models (Second edition)*, London: Chapman & Hall.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H. & Teller, E. (1953), 'Equations of state calculations by fast computing machine', *J. Chem. Phys* .

Murray, I. (2004), 'Note on rejection sampling and exact sampling with the metropolised independence sampler'.

Papaspiliopoulos, O., Roberts, G. O. & Sköld, M. (2003), Non-centered parameterizations for hierarchical models and data augmentation, *in* 'Bayesian statistics, 7 (Tenerife, 2002)', Oxford Univ. Press, New York, pp. 307–326. With a discussion by Alan E. Gelfand, Ole F. Christensen and Darren J. Wilkinson, and a reply by the authors.

Papaspiliopoulos, O., Roberts, G., O. & Sköld, M. (2007), 'A general framework for the parametrization of hierarchical models', *Statist. Science* **22**, 59–73.

R Core Team (2015), *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, Austria.
URL: *http://www.R-project.org/*

Raftery, A. E. & Lewis, S. M. (1992), 'One long run with diagnostics: Implementation strategies for Markov chain Monte Carlo', *Statisticsl Science* **7**, 493–497.

Reich, B. J., Hodges, J. S. & Zadnik, V. (2006), 'Effects of Residual Smoothing on the Posterior of the Fixed Effects in Disease-Mapping Models', *Biometrics* **62**, 1197–1206.

Roberts, G. O., Gelman, A. & Gilks, W. R. (1997), 'Weak convergence and optimal scaling of random walk Metropolis algorithms', *Ann. App. Probab* **7**, 110–120.

Roberts, G. O. & Rosenthal, J. S. (1998), 'Optimal scaling of discrete approximation to langevin diffusions', *J. Roy. Statist. Soc B* **60**.

Roberts, G. O. & Rosenthal, J. S. (2001), 'Optimal scaling for various Metropolis-Hastings algorithms', *Statist. Science* **16**, 351–367.

Roberts, G. O. & Rosenthal, J. S. (2009), 'Examples of Adaptive MCMC', *J. Comput. Graph. Statist.* **18**, 349–367.

Roberts, G. O. & Tweedie, R. L. (1996), 'Exponential Convergence of Langevin Distributions and Their Discrete Approximations', *Bernoulli* **2**, 341–366.

Roberts, G. & Stramer, O. (2003), 'Langevin diffusions and Metropolis Hastings algorithms', *Methodology and Computing in Applied Probability* **4**, 337–358.

Rue, H. & Held, L. (2005), *Gaussian Markov Random Fields: Theory and Applications*, Chapman & Hall.

Rue, H., Martino, S. & Chopin, N. (2009), 'Approxiamte Bayesian inference for latent Gaussian models by using integrated nested Laplace approximations', *Journal of the Royal Statistical Society, Series B* **71**, 1–35.

Sherlock, C., Xifara, T., Telfer, S. & Begon, M. (2013), 'A hidden Markov model for disease interactions in a host', *Journal of the Royal Statistical Society, Series C* **62(4)**, 609–627.

Sherlock, S., Fearnhead, P. & Roberts, G. O. (2010), 'The random walk Metropolis: linking theory and practice through a case study', *Statist. Science* **25**, 172–190.

Stein, M. L. (1999), *Interpolation of Spatial Data: Some Theory for Kriging*, Springer, New York.

Taylor, B. M. & Diggle, P. J. (2012), 'INLA or MCMC? A tutorial and comparative evaluation for spatial prediction in log-Gaussian Cox processes', *ArXiv e-prints* .

Tierney, L. (1994), 'Markov chains for exploring posterior distributions (with discussion)', *Annals of Statistics* .

Warnes, J. J. & Ripley, B. D. (1987), 'Problems with likelihood estimation of covariance functions of spatial Gaussian processes', *Biometrika* **74**, 640–642.

Xifara, T., Sherlock, C., Livingstone, S., Byrne, S. & Girolami, M. (2014), 'Langevin diffusions and the metropolis-adjusted langevin algorithm', *Statistics and Probability Letters* **91**, 14–19.