Spatial deformation in a Bayesian spatiotemporal model for matrix-variate responses

TESE DE DOUTORADO

por

Rodrigo de Souza Bulhões

Orientadora: Profa. Dra. Marina Silva Paez

Coorientador: Prof. Dr. Dani Gamerman

Universidade Federal do Rio de Janeiro

Instituto de Matemática

Programa de Pós-Graduação em Estatística

2024

Rodrigo de Souza Bulhões

SPATIAL DEFORMATION IN A BAYESIAN SPATIOTEMPORAL MODEL FOR MATRIX-VARIATE RESPONSES

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Estatística do Instituto de Matemática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Estatística.

Orientadora: Profa. Dra. Marina Silva Paez

Coorientador: Prof. Dr. Dani Gamerman

Rio de Janeiro, RJ - Brasil

Fevereiro de 2024

CIP - Catalogação na Publicação

de Souza Bulhões, Rodrigo
d933s Spatial deformation in a Bayesian spatiotemporal model for matrix-variate responses / Rodrigo de Souza Bulhões. -- Rio de Janeiro, 2024.
277 f.

Orientadora: Marina Silva Paez.
Coorientador: Dani Gamerman.
Tese (doutorado) - Universidade Federal do Rio de Janeiro, Instituto de Matemática, Programa de Pós Graduação em Estatística, 2024.

1. Modelos espaço-temporais matriz-variados. 2. Anisotropia. 3. Deformação espacial. 4. Geoestatística. 5. Modelos dinâmicos bayesianos. I. Silva Paez, Marina, orient. II. Gamerman, Dani, coorient. III. Título.

Elaborado pelo Sistema de Geração Automática da UFRJ com os dados fornecidos pelo(a) autor(a), sob a responsabilidade de Miguel Romeu Amorim Neto - CRB-7/6283.

Spatial deformation in a Bayesian spatiotemporal model for matrix-variate responses

Rodrigo de Souza Bulhões

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Estatística do Instituto de Matemática da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Estatística.

Aprovada em 21 / 02 / 2024

Dani Gamerman
Doutor – DME/IM/UFRJ, Presidente

João Batista de Morais Pereira
Doutor – DME/IM/UFRJ

Gustavo da Silva Ferreira
Doutor – ENCE/IBGE

Jony Arrais Pinto Junior
Doutor – GET/IME/UFF

Leonardo Soares Bastos
Doutor – PROCC/Fiocruz

Rio de Janeiro, RJ - Brasil
Fevereiro de 2024

Dedicatória

A meus pais, Geraldo Bulhões Bittencourt (in memoriam) e Cijara de Souza Bulhões, por acreditarem em mim e pelo apoio incondicional em todos os momentos.

Agradecimentos

Agradeço a todos que direta ou indiretamente contribuíram para que fosse possível a realização deste trabalho, expressando a minha gratidão especialmente:

A Deus e à Ciência, por ter atravessado a pandemia de COVID-19 física e mentalmente bem, dentro do que foi possível estar bem naquela época.

À professora Marina S. Paez, minha orientadora, por me aceitar como seu orientando e ter acompanhado bastante de perto o desenvolvimento de todas as etapas desta tese. Sou muito grato pela enorme paciência e compreensão, por suas palavras de encorajamento nos momentos em que tive dúvidas e por sua contínua disponibilidade para reuniões e leituras de versões preliminares do presente texto. A tenho como um exemplo de pessoa e profissional.

Ao professor Dani Gamerman, meu coorientador, por ter aceitado me orientar e pela sua disponibilidade em me atender todas as vezes que precisei. Sua vasta experiência me direcionou a caminhos que fizeram com que as contribuições desta tese se tornassem mais relevantes e interessantes.

À Cijara S. Bulhões, minha mãe, pelo carinho, exemplo de caráter e seu grande empenho para garantir meus estudos. Te amo!

A Geraldo B. Bittencourt Filho, meu irmão, e a Matheus F. Bittencourt, meu sobrinho, pela torcida para que tudo desse certo. Saibam que também torço pelo sucesso de ambos e que fico contente quando vocês alcançam o que desejam.

Aos docentes do Programa de Pós-Graduação em Estatística do IM/UFRJ (PPGE/IM/UFRJ) com quem tive a oportunidade de interagir enquanto cursava disciplinas nos dois primeiros anos de curso e/ou fazia os estágios de docência, pelos conhecimentos transmitidos. Ao professor Hugo T. Carvalho, a quem hoje chamo de amigo, pelo incentivo e suporte dados durante a escrita das minhas primeiras linhas de código no Python e, sobretudo, pelos ótimos papos e conselhos.

Aos funcionários do IM/UFRJ de setores como Secretaria de Ensino de Pós-Graduação, Laboratório de Sistemas Estocásticos (LSE) e Biblioteca, pelo suporte que me foi dado no decorrer do curso. Ao servidor Cláudio M. Rodrigues, pelo apoio que recebi desde a minha matrícula em assuntos burocráticos.

Aos professores Gustavo S. Ferreira, Jony Arrais P. Junior e Leonardo S. Bastos, por terem aceitado participar da banca de tese de doutorado. Aos professores Fidel E. Castro Morales e Mariane B. Alves, por terem aceitado participar da banca do exame de qualificação definitivo. Ao professor Carlos A. Abanto-Valle, por ter aceitado participar da banca do exame de qualificação preliminar. Ao professor João Batista M. Pereira, por ter aceitado participar das três bancas.

A meus colegas do PPGE/IM/UFRJ, pelas trocas de experiências durante intervalos para almoço ou pausas para tomar um cafezinho na copa do LSE e, principalmente, pelo apoio mútuo durante o período de disciplinas e de preparação para o exame de qualificação preliminar. A Marcel S. B. Quintana, querido colega de doutorado que também foi meu colega de mestrado no IME-USP, por ter me cedido uma rotina do `R` para elaborar gráficos de sua tese (Quintana, 2022) que me foi útil para gerar as Figuras 4.2(c) e 4.2(d).

A Adnê J. Moura Rodrigues, Elton B. D. Muniz, Felipe S. V. Almeida, Gabriel Alonso S. Ferreira, Geni P. Freitas, Maria das Graças (Gracinha) F. Nascimento, Rodrigo R. Cardoso, Sidnei Souza (in memoriam) e dona Terezinha J. S. Dias, queridos amigos e vizinhos que fiz na Cidade Maravilhosa durante a minha *fase tijucana*, por terem me apoiado em várias situações e me proporcionado tantos momentos leves e divertidos.

A Adriano R. Marques, Alan F. Lima, Carlos Wilker N. Santos, João Eduardo S. Araújo e Tayron S. Amaral, queridos amigos conterrâneos que acompanharam à distância a minha jornada no doutorado desde o começo, pela torcida por mim e por terem me recebido de forma calorosa quando eu voltava para a Bahia nos períodos de recesso acadêmico.

A meus colegas do Departamento de Estatística do IME/UFBa, pela concessão de afastamento para cursar doutorado por meio do Programa de Qualificação Docente.

À CAPES, pela bolsa de estudos.

Por último, mas não menos importante, gostaria de parafrasear Snoop Dogg para muito honestamente me agradecer por ter conquistado o prestigiado título de Doutor em Estatística pelo PPGE/IM/UFRJ. Tenho um imenso orgulho da UFRJ e me sinto honrado de ter passado por aqui nos últimos anos da minha formação acadêmica em Estatística!

Resumo

Neste trabalho, propomos um modelo espaço-temporal matriz-variado para ajustar duas ou mais variáveis respostas, mensuradas em diferentes pontos no tempo e locais distintos no espaço. Aqui relaxamos a hipótese de isotropia no espaço, uma vez que esta suposição não é razoável em várias situações reais. Tratamos a anisotropia do processo espacial usando o conceito de deformação espacial e lidamos com a estrutura temporal via modelos dinâmicos bayesianos. Através de estudos de simulação e também com dois exemplos, nós mostramos que a incorporação da deformação espacial ao modelo proporciona melhores interpolações em alguns cenários anisotrópicos. Apresentamos também uma solução para lidar com valores faltantes em diferentes variáveis respostas.

Palavras-Chaves: Modelos espaço-temporais matriz-variados, anisotropia, deformação espacial, geoestatística, modelos dinâmicos bayesianos.

Abstract

In this work we propose a matrix-variate spatiotemporal model to fit two or more response variables, measured at different points in time and distinct locations in space. Here we relax the hypothesis of isotropy in space, since this assumption is not reasonable in various real situations. We treat the anisotropy of the spatial process using the concept of spatial deformation and we deal with temporal structure via Bayesian dynamic models. Through simulation studies and also with two examples, we show that the incorporation of spatial deformation provides better interpolations in some anisotropic scenarios. We also present a solution to deal with missing values in different response variables.

Keywords: Matrix-variate spatiotemporal models, anisotropy, spatial deformation, geostatistics, Bayesian dynamic models.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

Environmental data sets are usually well characterized by processes that are observed over a finite set of times at fixed locations in a geographic region. In the last decades, several spatiotemporal models have been proposed to analyze such processes. These models generally assume that the process of interest follows a Gaussian process with some mean and some valid covariance function given by the product of the following elements: a variance parameter and an isotropic correlation function, defined as a function of the Euclidean distance between the locations (Schmidt and Guttorp, 2020). According to Guttorp and Schmidt (2013), it implies that the process is unchanged when the origin of the index set is translated (second-order stationarity implies strict stationarity for Gaussian processes) and that the process is invariant under rotation about the origin (by the isotropy assumption, it yields circular isocorrelation curves). Isotropic and stationary processes are said to be homogeneous. These concepts are briefly reviewed in Appendix A.2.

The assumption of isotropy is often unrealistic for various environmental processes (e.g. temperature, pollution, soil moisture and rainfall) because such processes tend to be affected by local aspects that lead to different behaviours in neighborhoods of distinct sites. Spatial characteristics such as landscape, topography, wind direction, and proximity to the ocean may influence the correlation structure (Morales et al., 2013, 2022). Various methods have been developed to model heterogeneous spatial processes, in which the most used in practice are: the deformation approach, the convolution approaches, covariates in the covariance function, and the stochastic partial differential equation approach (Mardia and Goodall, 1993; Sampson et al., 2001; Schmidt et al., 2002; Sampson, 2010; Guttorp and Schmidt, 2013; Schmidt and Guttorp, 2020).

This thesis aims to use the deformation approach proposed by Sampson and Guttorp (1992) to deal with non-homogeneous spatial processes. The main idea behind this methodology is to

map the original geographic coordinates to a new latent space where isotropy holds, applying multidimensional scaling to estimate the coordinates of gauged sites in the latent space and using thin-plate splines to make interpolation at any ungauged site in the geographic region. In their model, the authors assumed stationarity in time and, to satisfy this hypothesis in the context of real data sets, they worked with the residuals (commonly obtained after removing the temporal trends of the observations for each gauged site) by treating them as repeated measurements.

A limitation of the model proposed by Sampson and Guttorp (1992) is that it does not take into account the uncertainty related to the estimation of the coordinates in the latent space, which was duly accommodated in the works of Damian et al. (2001) and Schmidt and O'Hagan (2003) under the Bayesian paradigm. To relax the assumption of stationarity in time, Damian et al. (2003) and Bruno et al. (2009) allowed the incorporation of a temporal trend in the model. However, a disadvantage of the approach proposed in these last two works cited here is that the parameters are estimated in separate steps.

Under the assumption of isotropy, Bayesian dynamic models (West and Harrison, 1997) have been applied to deal with spatiotemporal data (Sansó and Guenni, 1999; Stroud et al., 2001; Huerta et al., 2004). In addition to relaxing the assumption of stationarity in time, this approach has advantages such as considering the model uncertainty in a unified framework and incorporating explanatory variables. Schmidt et al. (2011, Sec. 3) and Morales et al. (2013) utilized Bayesian dynamic models to address temporal variations and suggested correcting anisotropy via spatial deformation, facilitating a simultaneous estimation of all spatial and temporal components in a single step.

In this thesis, we are interested in analyzing multidimensional spatiotemporal processes that are observed over a finite set of times at fixed monitoring stations in a geographic region. Working with an extension of the matrix-normal Bayesian dynamic models (Quintana, 1987; Landim and Gamerman, 2000), Paez et al. (2008) modelled matrix-variate data from multidimensional spatiotemporal processes proposing the use of the isotropic Matérn covariance function to describe the spatial structure. In an application presented in their paper, the authors compared the interpolation performance between two univariate models (one for each response variable) and one bivariate model (where the two response variables are jointly modeled), showing that the predictions were very much improved when working with the response variables jointly. Our proposal for the analysis of matrix-variate observations made in space and time is to develop a new spatiotemporal model that uses a matrix-normal Bayesian dynamic model to capture the temporal structure and also to

impose a spatial structure that incorporates anisotropy via spatial deformation. For this purpose, an efficient Markov chain Monte Carlo algorithm was developed by us to simulate the posterior distribution of the model parameters.

The remainder of this thesis is organized as follows.

In Chapter 2 we propose an adaptation to the spatiotemporal model with spatial deformation for multivariate responses proposed by Morales et al. (2013) for the matrix-variate case, using some ideas from the works of Paez et al. (2008) and Paez and Gamerman (2013). Through simulation and with a real illustrative example, we also compare the results of the our proposed model with an analogous model that assumes the spatial isotropy hypothesis. Unlike most studies on spatial deformation models under the Bayesian paradigm, we employ the slice sampling algorithm to sample from the posterior distributions of individual coordinates.

While the model developed in Chapter 2 requires missing values to be imputed beforehand, Chapter 3 details how missing values can be jointly estimated with the unknown parameters of the model. By means of simulation and a real illustrative example, we compared the accuracy of estimating missing values between the isotropic model and the model with spatial deformation. In a scenario where there are high percentages of missing values, Chapter 4 employs the methodology proposed in Chapter 3 in a practical application involving three response variables.

Finally, Chapter 5 presents concluding remarks and suggestions for future work.

Chapter 2

Bayesian spatiotemporal model for anisotropic matrix-variate data

2.1 Introduction

In this work, we propose a matrix-variate spatiotemporal model to fit two or more response variables, measured at different points in time and distinct locations in a continuous space.

Various spatiotemporal models have been proposed in the literature considering more than one response variable, under the isotropy assumption in space. However, this hypothesis is not realistic in practice due to characteristics of geographic regions such as wind and topography (Sampson, 2010). Our aim is to propose a realistic and flexible class of models for matrix-variate space-time data that relax the assumptions of isotropy in space and stationarity in time. In the Bayesian paradigm, we treat the anisotropy of the spatiotemporal process using the concept of spatial deformation (Sampson and Guttorp, 1992) and capture the temporal trend through a matrix-normal dynamic model (Quintana, 1987; West and Harrison, 1997, Sec. 16.4).

This chapter is organized as follows. Section 2.2 presents a way to incorporate the deformation process in a spatiotemporal model for matrix-variate responses. Sections 2.3 and 2.3.2 present strategies to make forecasting and interpolation, respectively. In Section 2.4 we present some criteria for model comparison. Simulation studies are discussed in Section 2.5. An illustrative example is presented in Section 2.6. Lastly, in Section 2.7 we present some considerations.

2.2 Statistical modelling

2.2.1 Spatial deformation

Sampson and Guttorp (1992) proposed a semiparametric approach to model spatial covariance structures. In this paper, the authors supposed temporal stationarity, but they did not assume isotropy in the spatial covariance structure. The main idea of their work is to map the geographic coordinates located in $\mathcal{S}$ (geographic region of interest) to a new latent space $\mathcal{D}$ (where isotropy holds). This procedure is known as *spatial deformation*.

Damian et al. (2001) and Schmidt and O'Hagan (2003) presented Bayesian versions for the spatial deformation model, considering mappings from $\mathbb{R}^r$ to $\mathbb{R}^r$ for $r = 2$. These manuscripts worked with a two-dimensional random process $d(\cdot) = \{d(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$, which is called the *deformation process*, that maps the coordinates from $\mathcal{S}$- to $\mathcal{D}$-space, where $\underset{\sim}{\mathbf{s}} = (\text{lon}, \text{lat}) \in \mathcal{S} \subset \mathbb{R}^2$ is a site in the geographic region of interest and $d(\underset{\sim}{\mathbf{s}}) = (d_1(\underset{\sim}{\mathbf{s}}), d_2(\underset{\sim}{\mathbf{s}})) \in \mathcal{D} \subset \mathbb{R}^2$ is a location in a latent space that verifies the assumption of isotropy. Schmidt and O'Hagan (2003) assigned a Gaussian process prior for $d(\cdot)$ with a prior mean function $\mu_d(\cdot)$ and a prior covariance function $\boldsymbol{\sigma}_d^2\rho_d(\cdot)$, where $\boldsymbol{\sigma}_d^2$ is a 2×2 covariance matrix and, for each pair of sites from $\mathcal{S}$-space, $\rho_d : \mathbb{R}^2 \to \mathbb{R}$ is a function which measures the prior correlation between these same two sites within $\mathcal{D}$-space. The quantities μ_d, $\boldsymbol{\sigma}_d^2$ and ρ_d may be specified as follows:

- $\mu_d : \mathbb{R}^2 \to \mathbb{R}^2$ is an endofunction (a.k.a. self-mapping). We usually specify $\mu_d(\cdot)$ as the identity function when there is no prior information about how $\mathcal{D}$ differs from $\mathcal{S}$, which implies that $\text{E}[d(\underset{\sim}{\mathbf{s}}_n)] = \mu_d(\underset{\sim}{\mathbf{s}}_n) = \underset{\sim}{\mathbf{s}}_n = (\text{lon}_n, \text{lat}_n)$ for each site $n \in \{1, \ldots, N\}$.

- The 2×2 covariance matrix $\boldsymbol{\sigma}_d^2$ calibrates the prior covariance structure of the coordinate system in $\mathcal{D}$-space, being recommended by Schmidt and O'Hagan (2003, Sec. 2.2.2) to take $\boldsymbol{\sigma}_d^2$ as a diagonal matrix to avoid unidentifiability issues in the model (i.e. assuming the form $\boldsymbol{\sigma}_d^2 = \text{diag}\{\sigma_{d_{1,1}}^2, \sigma_{d_{2,2}}^2\}$). Their diagonal elements control the level of distortion on the mapping from $\mathcal{S}$- to $\mathcal{D}$-space, pointing out that small values of $\sigma_{d_{1,1}}^2$ and $\sigma_{d_{2,2}}^2$ suggest that the deformations are smooth.

  As usually $\boldsymbol{\sigma}_d^2$ is not well estimated, some works impose an informative prior distribution for this quantity (Schmidt and O'Hagan, 2003; Morales et al., 2013, 2022). Because of this, here we treat $\boldsymbol{\sigma}_d^2$ as a hyperparameter as done by Damian et al. (2001) and Morales and Vicini (2020) in a similar context.

Denote by $\mathbf{S} = \left[\begin{array}{ccc} \underset{\sim}{\mathbf{s}}_1 & \cdots & \underset{\sim}{\mathbf{s}}_N \end{array} \right]$ the $2 \times N$ matrix formed by the N gauged sites in $\mathcal{S}$. When $\boldsymbol{\sigma}_d^2$ is fixed rather than estimated, we note that $\boldsymbol{\sigma}_d^2$ need not be a diagonal matrix. If the researcher does not have tools or information to specify $\boldsymbol{\sigma}_d^2$, we suggest using the general specification that fixes $\boldsymbol{\sigma}_d^2$ as the sample covariance matrix of the matrix of the gauged sites[1], which is described by

$$\boldsymbol{\sigma}_d^2 = \widehat{\mathrm{Cov}}(\mathbf{S}) = \frac{1}{N-1} \cdot \left[\begin{array}{cc} \sum_{n=1}^{N} \left(\mathrm{lon}_n - \overline{\mathrm{lon}}\right)^2 & \sum_{n=1}^{N} \mathrm{lon}_n \cdot \mathrm{lat}_n - N \cdot \overline{\mathrm{lon}} \cdot \overline{\mathrm{lat}} \\ \sum_{n=1}^{N} \mathrm{lon}_n \cdot \mathrm{lat}_n - N \cdot \overline{\mathrm{lon}} \cdot \overline{\mathrm{lat}} & \sum_{n=1}^{N} \left(\mathrm{lat}_n - \overline{\mathrm{lat}}\right)^2 \end{array} \right], \quad (2.1)$$

where $\overline{\mathrm{lon}} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{lon}_n$ and $\overline{\mathrm{lat}} = \frac{1}{N} \sum_{n=1}^{N} \mathrm{lat}_n$.

In this thesis, we work with the specification $\boldsymbol{\sigma}_d^2 = \widehat{\mathrm{Cov}}(\mathbf{S})$. In Chapter 4, we will compare this choice for $\boldsymbol{\sigma}_d^2$ with another choice for $\boldsymbol{\sigma}_d^2$ made arbitrarily.

- Let $\mathbf{R}_d = \left[\begin{array}{c} R_{n,n'} \end{array} \right]_{N \times N}$ be a $N \times N$ correlation matrix with entries given by a Gaussian correlation function[2], where, for all $n, n' \in \{1, \ldots, N\}$, we have:

$$R_{n,n'} = \mathrm{Corr}[d(\underset{\sim}{\mathbf{s}}_n), d(\underset{\sim}{\mathbf{s}}_{n'})] = \rho_d(\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}) = \left\{ \begin{array}{ll} \exp\{-\psi \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{array} \right. .$$

According to Schmidt and O'Hagan (2003, Sec. 2.2.1), $\mathbf{R}_d$ gives the prior correlation structure of the gauged sites in $\mathcal{D}$-space and controls the degree of smoothness of the Gaussian process, where the term ψ is a fixed positive quantity that controls the prior shape of the configuration of the gauged sites in $\mathcal{D}$ as follows: greater spatial separation in $\mathcal{S}$-dimensional space results in lower prior correlation in $\mathcal{D}$-space, so their distances may be more distorted in $\mathcal{D}$. The limit cases are: $\mathbf{R}_d \to \mathbf{1}_{N \times N}$ if $\psi \to 0$, where $\mathbf{1}_{N \times N}$ is the $N \times N$ matrix of ones (is not valid, as it is not invertible); and $\mathbf{R}_d \to \mathbf{I}_N$ if $\psi \to +\infty$, where $\mathbf{I}_N$ is the identity matrix of order N.

In our experience with spatial deformation, specifying ψ to obtain satisfactory results over the isotropic approach may require several attempts. Authors like Morales et al. (2013, Sec. 3.1), Morales and Vicini (2020, Sec. 2) and Morales et al. (2022, Sec. 2.1) recommend assigning values such as $-2\ln(0.05)/\max_{n,n' \in \{1,\ldots,N\}} \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2$ and $1/\left(2 \max_{n,n' \in \{1,\ldots,N\}} \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2\right)$ for $\psi > 0$.

Let $\mathbf{D} = \left[\begin{array}{c} D_{m,n} \end{array} \right]_{2 \times N} = \left[\begin{array}{ccc} \underset{\sim}{\mathbf{d}}_1 & \cdots & \underset{\sim}{\mathbf{d}}_N \end{array} \right]$ be a $2 \times N$ random matrix, where $\underset{\sim}{\mathbf{d}}_n = d(\underset{\sim}{\mathbf{s}}_n) \in \mathcal{D}$ is its n^{th} column for all $n \in \{1, \ldots, N\}$. As a Gaussian process prior was assigned for $d(\cdot)$, by

[1] Although it is not a diagonal matrix, its off-diagonal elements are approximately equal to zero due to longitude and latitude being unrelated quantities.

[2] Other valid correlation functions can be considered.

Definition A.10 we have that $\mathbf{D}$ has a matrix-variate normal prior distribution with a $2 \times N$ mean matrix $\mathbf{S}$, a 2×2 left covariance matrix $\boldsymbol{\sigma}_d^2$ and a $N \times N$ right covariance matrix $\mathbf{R}_d$, which is denoted by $\mathbf{D} \sim \mathsf{N}_{2\times N}(\mathbf{S}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d)$.

For further details about $\boldsymbol{\sigma}_d^2$ and ψ, see the sensitivity studies for prior specifications presented in Schmidt and O'Hagan (2003, Sec. 4.1) and Morales et al. (2013, Sec. 4.1).

2.2.2 Spatiotemporal modelling for matrix-variate responses with spatial deformation

Morales et al. (2013) present a spatiotemporal model for multivariate responses that uses spatial deformation to treat anisotropy and also considers a dynamic model (West and Harrison, 1997) to study the temporal trend. In their work, data is a variable/index that is being measured over time and at different fixed locations. Thus, the response for each time is described by a vector whose components are measurements of a variable/index made at different fixed locations.

Based on Paez et al. (2008) and Paez and Gamerman (2013), we propose a dynamic model to accommodate matrix-variate responses observed in space and time. We followed texts such as Schmidt et al. (2011, Sec. 3) and Morales et al. (2013) to incorporate spatial deformation into the proposed model. In this work, data is a set of variables/indices that are measured over time and at different fixed locations. Thus, the response for each time is described by a matrix whose components in each column are measurements of a variable/index made at different fixed locations.

Given a geographic domain of interest described by a continuous space $\mathcal{S} \subset \mathbb{R}^r$ and a finite set of times $\mathcal{T} = \{1, \ldots, T\} \subset \mathbb{N}$, consider a q-dimensional Gaussian spatiotemporal response process $\{Y(\underset{\sim}{\mathbf{s}}, t) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}, t \in \mathcal{T}\}$, where $Y(\underset{\sim}{\mathbf{s}}, t) = (Y_1(\underset{\sim}{\mathbf{s}}, t), \ldots, Y_q(\underset{\sim}{\mathbf{s}}, t)) \in \mathcal{Y} \subset \mathbb{R}^q$ is a multivariate response for fixed $\underset{\sim}{\mathbf{s}} \in \mathcal{S}$ and $t \in \mathcal{T}$. Suppose further that q-variate observations are made in N sites or monitoring stations located in $\mathcal{S}$ and over T distinct points in time. Let $\underset{\sim}{\mathbf{s}}_n = (s_{1,n}, \ldots, s_{r,n})$ be the vector of coordinates of the n^{th} site. Denote $Y_{n,i,t} = Y_i(\underset{\sim}{\mathbf{s}}_n, t)$ the observed value at time t and site $\underset{\sim}{\mathbf{s}}_n$ of the response variable i, where $t \in \{1, \ldots, T\}$, $n \in \{1, \ldots, N\}$ and $i \in \{1, \ldots, q\}$. For each time $t \in \{1, \ldots, T\}$, define the following response matrix:

$$\mathbf{Y}_t = \left[\ Y_{n,i,t}\ \right]_{N\times q} = \begin{bmatrix} Y_{1,1,t} & \cdots & Y_{1,q,t} \\ \vdots & \ddots & \vdots \\ Y_{N,1,t} & \cdots & Y_{N,q,t} \end{bmatrix} = \left[\ \underset{\sim}{\mathbf{Y}}_{1,t} \quad \cdots \quad \underset{\sim}{\mathbf{Y}}_{q,t}\ \right], \tag{2.2}$$

where $\underset{\sim}{\mathbf{Y}}_{i,t} = \begin{bmatrix} Y_{1,i,t} \\ \vdots \\ Y_{N,i,t} \end{bmatrix}$ is the i^{th} column of $\mathbf{Y}_t$.

In this thesis, we are interested in working with two-dimensional spaces (i.e. $r = 2$). Therefore, here $\underset{\sim}{\mathbf{s}}_n = (\text{lon}_n, \text{lat}_n) \in \mathcal{S} \subset \mathbb{R}^2$ is the vector of geographic coordinates of the n^{th} site.

To model the i^{th} column or response variable, we assume that they all have the same spatial dependence structure given by a matrix $\mathbf{B}$ and their own temporal dependence structure:

$$
\begin{aligned}
\underset{\sim}{\mathbf{Y}}_{i,t} \mid \underset{\sim}{\boldsymbol{\beta}}_{i,t}, V, \phi, \mathbf{D} &\sim \mathsf{N}_N(\mathbf{X}_t \underset{\sim}{\boldsymbol{\beta}}_{i,t}, V \cdot \mathbf{B}),\ t \in \{1, \ldots, T\}, \\
\mathbf{B} &= \left[\ B_{n,n'}\ \right]_{N \times N}, \\
B_{n,n'} &= \begin{cases} \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'}) \|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases}, \\
\underset{\sim}{\boldsymbol{\beta}}_{i,t} \mid \underset{\sim}{\boldsymbol{\beta}}_{i,t-1}, V &\sim \mathsf{N}_p(\mathbf{G}_t \underset{\sim}{\boldsymbol{\beta}}_{i,t-1}, V \cdot \mathbf{W}),\ t \in \{1, \ldots, T\}, \\
\underset{\sim}{\boldsymbol{\beta}}_{i,0} \mid V &\sim \mathsf{N}_p(\underset{\sim}{\mathbf{M}}_{i,0}, V \cdot \mathbf{C}_0), \\
V &\sim \mathsf{IG}(a_V, b_V), \\
\phi &\sim \mathsf{G}(a_\phi, b_\phi), \\
\mathbf{D} &\sim \mathsf{N}_{2 \times N}(\mathbf{S}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d),
\end{aligned}
$$

where:

- $\mathsf{N}_u(\cdot, \cdot)$ denotes the u-variate normal distribution, $\mathsf{N}_{u \times v}(\cdot, \cdot, \cdot)$ denotes the $(u \times v)$-variate normal distribution, $\mathsf{G}(\cdot, \cdot)$ denotes the gamma distribution, and $\mathsf{IG}(\cdot, \cdot)$ denotes the inverse gamma distribution, whose definitions are reviewed in Appendix A.1;
- $\mathbf{X}_t$ and $\mathbf{G}_t$ are fixed $N \times p$ and $p \times p$ matrices, respectively, where p indicates the number of regression coefficients for each time and response variable; and
- $\underset{\sim}{\mathbf{M}}_{i,0}$ and $\mathbf{C}_0$, a_V and b_V, $a_{\boldsymbol{\Sigma}}$ and $\mathbf{b}_{\boldsymbol{\Sigma}}$, a_ϕ and b_ϕ, $\sigma^2_{d_{1,1}}$ and $\sigma^2_{d_{2,2}}$ (related to $\boldsymbol{\sigma}_d^2$), and ψ (related to $\mathbf{R}_d$) are hyperparameters to be assigned and $\mathbf{W}$ is a fixed $p \times p$ matrix.

Following some works such as West and Harrison (1997, Secs. 2.5 and 4.5), Gamerman and Lopes (2006, Sec. 2.5.3), Petris et al. (2009, Sec. 4.3.1) and Triantafyllopoulos (2021, Secs. 4.3.3 and 5.5.1), three covariance matrices of the model have V as a multiplier, or scale factor, to provide a scale-free model in terms of $\mathbf{B}$, $\mathbf{W}$ and $\mathbf{C}_0$. It should be emphasized that we obtained better estimation results by specifying a covariance matrix of the form $V \cdot \mathbf{W}$ with $V \sim \mathsf{IG}(a_V, b_V)$ and fixed $\mathbf{W} \in \text{Sym}^+(p)$ instead of eliciting $\mathbf{W} \sim \mathsf{IW}_p(a_{\mathbf{W}}, \mathbf{b}_{\mathbf{W}})$, or $\mathbf{W} = \text{diag}\{W_1, \ldots, W_p\}$ with $W_w \sim \mathsf{IG}(a_{W_w}, b_{W_w})$ for $w \in \{1, \ldots, p\}$, where $\mathsf{IW}_c(\cdot, \cdot)$ denotes the $c \times c$ inverse Wishart distribution and $\text{Sym}^+(c)$ denotes the space of symmetric positive-definite $c \times c$ matrices. Simulation studies

carried out in some dissertations and theses on spatiotemporal modelling show that the components of $\mathbf{W}$ are usually overestimated (Paez, 2004, Sec. 3.6.2; Hansen, 2009, Chap. 4; Morales, 2010, Sec. 2.5.1; Costa, 2011, Sec. 4.4.3). In this text, we will only work with the trivial choice $\mathbf{W} = \mathbf{I}_p$.

Based on Salvador et al. (2004, Sec. 2.1), define $\boldsymbol{\Sigma}$ a time-invariant $q \times q$ matrix to accommodate the relation among response variables. Similar to $\mathbf{Y}_t = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{1,t} & \cdots & \underset{\sim}{\mathbf{Y}}_{q,t} \end{bmatrix}$, write the following $p \times q$ matrices: $\mathbf{M}_0 = \begin{bmatrix} \underset{\sim}{\mathbf{M}}_{1,0} & \cdots & \underset{\sim}{\mathbf{M}}_{q,0} \end{bmatrix}$, $\boldsymbol{\beta}_0 = \begin{bmatrix} \underset{\sim}{\boldsymbol{\beta}}_{1,0} & \cdots & \underset{\sim}{\boldsymbol{\beta}}_{q,0} \end{bmatrix}$ and, for $t \in \{1, \ldots, T\}$, $\boldsymbol{\beta}_t = \begin{bmatrix} \underset{\sim}{\boldsymbol{\beta}}_{1,t} & \cdots & \underset{\sim}{\boldsymbol{\beta}}_{q,t} \end{bmatrix}$. We propose a matrix-variate spatiotemporal model given by:

$$
\begin{aligned}
\mathbf{Y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} &\sim \mathsf{N}_{N\times q}(\mathbf{X}_t\boldsymbol{\beta}_t, V \cdot \mathbf{B}, \boldsymbol{\Sigma}),\ t \in \{1, \ldots, T\}, \\
\mathbf{B} &= \begin{bmatrix} B_{n,n'} \end{bmatrix}_{N\times N}, \\
B_{n,n'} &= \begin{cases} \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'})\|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases}, \\
\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{p\times q}(\mathbf{G}_t\boldsymbol{\beta}_{t-1}, V \cdot \mathbf{W}, \boldsymbol{\Sigma}),\ t \in \{1, \ldots, T\}, \\
\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{p\times q}(\mathbf{M}_0, V \cdot \mathbf{C}_0, \boldsymbol{\Sigma}), \\
V &\sim \mathsf{IG}(a_V, b_V), \\
\boldsymbol{\Sigma} &\sim \mathsf{IW}_q(a_{\boldsymbol{\Sigma}}, \mathbf{b}_{\boldsymbol{\Sigma}}), \\
\phi &\sim \mathsf{G}(a_\phi, b_\phi), \\
\mathbf{D} &\sim \mathsf{N}_{2\times N}(\mathbf{S}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d).
\end{aligned}
\tag{2.3}
$$

In Equation (2.3), the collection of unknown model parameters is $\boldsymbol{\theta} = \{\phi, \mathbf{D}, V, \boldsymbol{\Sigma}, \boldsymbol{\beta}_0, \boldsymbol{\beta}\}$, where $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T\}$. The corresponding parametric space is:

$$
\boldsymbol{\Theta} = (0, +\infty) \times \mathbb{R}^{2\times N} \times (0, +\infty) \times \mathrm{Sym}^+(q) \times \left(\prod_{t=0}^{T} \mathbb{R}^{p\times q} \right).
$$

Assuming prior independence for the parameters ϕ, $\mathbf{D}$, V and $\boldsymbol{\Sigma}$, by the law of total probability we have

$$
\begin{aligned}
f(\boldsymbol{\theta}) &= f(\phi, \mathbf{D}, V, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_0, \boldsymbol{\beta} \mid \phi, \mathbf{D}, V, \boldsymbol{\Sigma}) \\
&= f(\phi) f(\mathbf{D}) f(V) f(\boldsymbol{\Sigma}) \\
&\times f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_1 \mid \boldsymbol{\beta}_0, V, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_2 \mid \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, V, \boldsymbol{\Sigma}) \cdots f(\boldsymbol{\beta}_T \mid \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_{T-1}, V, \boldsymbol{\Sigma}),
\end{aligned}
$$

and, by the Markovian property, we write the following joint prior density:

$$
f(\boldsymbol{\theta}) = f(\phi) f(\mathbf{D}) f(V) f(\boldsymbol{\Sigma}) f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \prod_{t=1}^{T} f(\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}). \tag{2.4}
$$

Define $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_T\}$ the collection of matrix-variate responses, assuming that $\mathbf{Y}_1, \ldots, \mathbf{Y}_T$ are T conditionally independent observations given $\boldsymbol{\theta}$. Thus, the likelihood function of $\boldsymbol{\theta}$ is:

$$l(\boldsymbol{\theta}; \mathbf{y}) = f(\mathbf{y} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}). \tag{2.5}$$

To assess only the sub-collection of parameters given by $\{V, \phi, \mathbf{D}, \boldsymbol{\Sigma}\}$, treating $\{\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T\}$ as nuisance parameters, see the corresponding marginal likelihood function in Appendix A.3.

From Equations (2.4) and (2.5), by the Bayes' theorem we have the following posterior density:

$$\begin{aligned} f(\boldsymbol{\theta} \mid \mathbf{y}) &\propto f(\boldsymbol{\theta}) l(\boldsymbol{\theta}; \mathbf{y}) \\ &= f(\phi) f(\mathbf{D}) f(V) f(\boldsymbol{\Sigma}) f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \left[\prod_{t=1}^{T} f(\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}) f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \right]. \end{aligned} \tag{2.6}$$

2.2.3 Bayesian inference via Markov chain Monte Carlo

The posterior density given in Equation (2.7) does not have a closed-form. For this reason, we resort the Markov chain Monte Carlo (MCMC) method to obtain samples from the model parameters. Next, we will obtain the full conditional distributions of the parameters in order to implement a hybrid algorithm.

Some model parameters are non-identifiable. Banerjee et al. (2014, Sec. 6.1.1.1) claim that the product $V \cdot \phi$ is identified, but not the individual parameters. The covariance parameters of the matrix-variate normal distribution are non-identifiable in the sense that, for any scale factor $\tau > 0$, we have that $\mathsf{N}_{a \times b}(\mathbf{M}, \mathbf{R}, \mathbf{C})$ and $\mathsf{N}_{a \times b}(\mathbf{M}, \tau \cdot \mathbf{R}, \frac{1}{\tau} \cdot \mathbf{C})$ are both equivalent to $\mathsf{N}_{ab}(\operatorname{vec} \mathbf{M}, \mathbf{C} \otimes \mathbf{R})$ by Definition A.6. Because of this fact, instead of analyzing the posterior distributions of the unknown parameters $V > 0$, $\phi > 0$ and $\boldsymbol{\Sigma} = [\Sigma_{i,i'}]_{q \times q}$ separately, we will study the posterior distribution of the elements $V \cdot \Sigma_{i,i'}$ or even $\phi \cdot V \cdot \Sigma_{i,i'}$ for $i, i' \in \{1, \ldots, q\}$, as recommended by Landim and Gamerman (2000, Sec. 4) in a similar context. It is also known that $\mathbf{D}$ is an unidentifiable parameter, which will be discussed in Section 2.2.3.5.

2.2.3.1 Sampling from the full conditional distribution of V

The full conditional density of V is given by

$$
\begin{aligned}
&f(V \mid \mathbf{y}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \propto f(V) f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \prod_{t=1}^{T} \left[f(\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}) f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \right] \\
&\propto V^{-a_V - 1} \exp\left\{ -\frac{b_V}{V} \right\} \mathbb{1}_{(0,+\infty)}(V) \\
&\times \left[\det(V \cdot \mathbf{C}_0) \right]^{-\frac{q}{2}} \exp\left\{ -\frac{1}{2V} \operatorname{tr}\left[(\boldsymbol{\beta}_0 - \mathbf{M}_0)^\intercal \mathbf{C}_0^{-1} (\boldsymbol{\beta}_0 - \mathbf{M}_0) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&\times \left\{ \left[\det(V \cdot \mathbf{W}) \right]^{-\frac{q}{2}} \right\}^T \exp\left\{ -\frac{1}{2V} \sum_{t=1}^{T} \operatorname{tr}\left[(\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1})^\intercal \mathbf{W}^{-1} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1}) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&\times \left\{ \left[\det(V \cdot \mathbf{B}) \right]^{-\frac{q}{2}} \right\}^T \exp\left\{ -\frac{1}{2V} \sum_{t=1}^{T} \operatorname{tr}\left[(\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^\intercal \mathbf{B}^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&\propto V^{-a'_V - 1} \exp\left\{ -\frac{b'_V}{V} \right\} \mathbb{1}_{(0,+\infty)}(V),
\end{aligned}
$$

which implies that

$$
V \mid \mathbf{Y} = \mathbf{y}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{IG}(a'_V, b'_V) \tag{2.7}
$$

by conjugacy, where

$$
a'_V = a_V + \frac{pq}{2} + \frac{Tpq}{2} + \frac{TNq}{2}
$$

and

$$
\begin{aligned}
b'_V \;=\; b_V \;&+\; \frac{1}{2} \operatorname{tr}\left[(\boldsymbol{\beta}_0 - \mathbf{M}_0)^\intercal \mathbf{C}_0^{-1} (\boldsymbol{\beta}_0 - \mathbf{M}_0) \boldsymbol{\Sigma}^{-1} \right] \\
&+\; \frac{1}{2} \operatorname{tr}\left[\sum_{t=1}^{T} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1})^\intercal \mathbf{W}^{-1} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1}) \boldsymbol{\Sigma}^{-1} \right] \\
&+\; \frac{1}{2} \operatorname{tr}\left[\sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^\intercal \mathbf{B}^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t) \boldsymbol{\Sigma}^{-1} \right].
\end{aligned}
$$

In Appendix B.1.2.1, we present a program written in Python to sample from the full conditional distribution of V given in Equation (2.7).

2.2.3.2 Sampling from the full conditional distribution of $\boldsymbol{\Sigma}$

The full conditional density of $\boldsymbol{\Sigma}$ is given by

$$
\begin{aligned}
&f(\boldsymbol{\Sigma} \mid \mathbf{y}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, V, \phi, \mathbf{D}) \propto f(\boldsymbol{\Sigma}) f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \prod_{t=1}^{T} \left[f(\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}) f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \right] \\
&\propto (\det \boldsymbol{\Sigma})^{-\left(q + \frac{a_{\boldsymbol{\Sigma}}}{2}\right)} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[\mathbf{b}_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \right] \right\} \mathbb{1}_{\mathrm{Sym}^+(q)}(\boldsymbol{\Sigma}) \\
&\times (\det \boldsymbol{\Sigma})^{-\frac{p}{2}} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[(\boldsymbol{\beta}_0 - \mathbf{M}_0)^\intercal (V \cdot \mathbf{C}_0)^{-1} (\boldsymbol{\beta}_0 - \mathbf{M}_0) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&\times [(\det \boldsymbol{\Sigma})^{-\frac{p}{2}}]^T \exp\left\{ -\frac{1}{2} \sum_{t=1}^{T} \operatorname{tr}\left[(\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1})^\intercal (V \cdot \mathbf{W})^{-1} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1}) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&\times [(\det \boldsymbol{\Sigma})^{-\frac{N}{2}}]^T \exp\left\{ -\frac{1}{2} \sum_{t=1}^{T} \operatorname{tr}\left[(\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^\intercal (V \cdot \mathbf{B})^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t) \boldsymbol{\Sigma}^{-1} \right] \right\} \\
&\propto (\det \boldsymbol{\Sigma})^{-\left(q + \frac{a'_{\boldsymbol{\Sigma}}}{2}\right)} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[\mathbf{b}'_{\boldsymbol{\Sigma}} \boldsymbol{\Sigma}^{-1} \right] \right\} \mathbb{1}_{\mathrm{Sym}^+(q)}(\boldsymbol{\Sigma}),
\end{aligned}
$$

which implies that

$$
\boldsymbol{\Sigma} \mid \mathbf{Y} = \mathbf{y}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, V, \phi, \mathbf{D} \sim \mathsf{IW}_q(a'_{\boldsymbol{\Sigma}}, \mathbf{b}'_{\boldsymbol{\Sigma}}) \tag{2.8}
$$

by conjugacy, where

$$
a'_{\boldsymbol{\Sigma}} = a_{\boldsymbol{\Sigma}} + p + Tp + TN
$$

and

$$
\begin{aligned}
\mathbf{b}'_{\boldsymbol{\Sigma}} \;=\; \mathbf{b}_{\boldsymbol{\Sigma}} \;&+\; (\boldsymbol{\beta}_0 - \mathbf{M}_0)^\intercal (V \cdot \mathbf{C}_0)^{-1} (\boldsymbol{\beta}_0 - \mathbf{M}_0) \\
&+\; \sum_{t=1}^{T} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1})^\intercal (V \cdot \mathbf{W})^{-1} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1}) \\
&+\; \sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^\intercal (V \cdot \mathbf{B})^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t).
\end{aligned}
$$

In Appendix B.1.2.2, we present a program written in Python to sample from the full conditional distribution of $\boldsymbol{\Sigma}$ given in Equation (2.8).

2.2.3.3 Sampling from the full conditional distribution of $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$

For all $t \in \{0, 1, \ldots, T-1, T\}$, the full conditional density of $\boldsymbol{\beta}_t$ is generally given by:

$$
f(\boldsymbol{\beta}_t \mid \mathbf{y}, \boldsymbol{\theta} \backslash \{\boldsymbol{\beta}_t\}) \propto \begin{cases} f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_1 \mid \boldsymbol{\beta}_0, V, \boldsymbol{\Sigma}), & t = 0 \\ f(\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}) f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_{t+1} \mid \boldsymbol{\beta}_t, V, \boldsymbol{\Sigma}), & t \in \{1, \ldots, T-1\} \\ f(\boldsymbol{\beta}_T \mid \boldsymbol{\beta}_{T-1}, V, \boldsymbol{\Sigma}) f(\mathbf{y}_T \mid \boldsymbol{\beta}_T, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}), & t = T \end{cases} . \tag{2.9}
$$

We use the Forward-Filtering Backward-Sampling (FFBS) algorithm (Frühwirth-Schnatter, 1994; Carter and Kohn, 1994; Chib and Greenberg, 1995) to sample from the posterior distribution of the parameters $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$. Theoretical aspects about the matrix-variate extension of the FFBS algorithm are discussed in texts such as Landim and Gamerman (2000), Salvador

et al. (2004) and Jiménez and Pereira (2021). This method is described in Algorithm 1 and its implementation is shown in Appendix B.1.2.3.

Algorithm 1 FFBS algorithm to sample from the full conditional distribution of the parameters $\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T$.

Start with $\mathbf{M}_0$, $\mathbf{C}_0$ and $\boldsymbol{\theta}^{(j-1)} = \left\{\boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right\}$.

1: Compute $\tilde{\mathbf{C}}_0 = V^{(j-1)} \cdot \mathbf{C}_0$ and the matrix $\mathbf{B}_{(j-1)} = \left[\ B_{n,n'}^{(j-1)}\ \right]_{N\times N}$, where:

$$B_{n,n'}^{(j-1)} = \exp\{-\phi^{(j-1)}\|d^{(j-1)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j-1)}(\underset{\sim}{\mathbf{s}}_{n'})\|\}, \quad n, n' \in \{1, \ldots, N\}.$$

2: **for** $t \leftarrow 1$ to T **do**

$$\begin{aligned}
\mathbf{E}_t &= V^{(j-1)} \cdot \mathbf{W} + \mathbf{G}_t \tilde{\mathbf{C}}_{t-1} \mathbf{G}_t^\intercal, \\
\mathbf{Q}_t &= V^{(j-1)} \cdot \mathbf{B}_{(j-1)} + \mathbf{X}_t \mathbf{E}_t \mathbf{X}_t^\intercal, \\
\mathbf{a}_t &= \mathbf{G}_t \mathbf{M}_{t-1}, \\
\mathbf{M}_t &= \mathbf{a}_t + \mathbf{E}_t \mathbf{X}_t^\intercal \mathbf{Q}_t^{-1}(\mathbf{y}_t - \mathbf{X}_t \mathbf{a}_t), \\
\tilde{\mathbf{C}}_t &= \mathbf{E}_t - \mathbf{E}_t \mathbf{X}_t^\intercal \mathbf{Q}_t^{-1} \mathbf{X}_t \mathbf{E}_t.
\end{aligned}$$

3: **end for**

4: Sample $\boldsymbol{\beta}_T^{(j)}$ from the distribution $\mathsf{N}_{p\times q}(\mathbf{M}_T, \tilde{\mathbf{C}}_T, \boldsymbol{\Sigma}^{(j-1)})$.

5: **for** $t \leftarrow T-1$ to 0 **do**

6: Sample $\boldsymbol{\beta}_t^{(j)}$ from the distribution $\mathsf{N}_{p\times q}(\mathbf{H}_t \mathbf{h}_t, \mathbf{H}_t, \boldsymbol{\Sigma}^{(j-1)})$, where:

$$\begin{aligned}
\mathbf{H}_t &= \left\{\tilde{\mathbf{C}}_t^{-1} + \mathbf{G}_t^\intercal \left[V^{(j-1)} \cdot \mathbf{W}\right]^{-1} \mathbf{G}_t\right\}^{-1}, \\
\mathbf{h}_t &= \tilde{\mathbf{C}}_t^{-1} \mathbf{M}_t + \mathbf{G}_{t+1}^\intercal \left[V^{(j-1)} \cdot \mathbf{W}\right]^{-1} \boldsymbol{\beta}_{t+1}^{(j)}.
\end{aligned}$$

7: **end for**

2.2.3.4 Sampling from the full conditional distribution of ϕ

For all $\phi > 0$, the natural logarithm of the full conditional density of ϕ is given by:

$$\begin{aligned} \ln f(\phi \mid \mathbf{y}, \boldsymbol{\beta}, V, \mathbf{D}, \boldsymbol{\Sigma}) &= \text{constant} + \ln f(\phi) + \sum_{t=1}^{T} \ln f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\ &\propto \{(a_\phi - 1) \cdot \ln \phi - b_\phi \cdot \phi\} \\ &+ \left\{ -\frac{Tq}{2} \ln \det \mathbf{B} - \frac{1}{2} \operatorname{tr} \left[\sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^\intercal (V \cdot \mathbf{B})^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t) \boldsymbol{\Sigma}^{-1} \right] \right\}. \end{aligned} \tag{2.10}$$

Since $f(\phi \mid \mathbf{y}, \boldsymbol{\beta}, V, \mathbf{D}, \boldsymbol{\Sigma})$ does not have a known form, we sample from this density through the Metropolis-Hastings algorithm. The method is described in Algorithm 2 and its implementation is presented in Appendix B.1.2.4.

Algorithm 2 Metropolis-Hastings algorithm to sample from $f(\phi \mid \mathbf{y}, \boldsymbol{\beta}, V, \mathbf{D}, \boldsymbol{\Sigma})$.

Start with a_ϕ, b_ϕ and $\boldsymbol{\theta}^{(j-1)} = \left\{ \boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)} \right\}$, where $j \in \mathbb{N}$ and $\boldsymbol{\beta}^{(j-1)} = \left\{ \boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)} \right\}$.

1: Generate a random number $u \in (0, 1)$ and propose a new value for ϕ by sampling from the inverse normal distribution given by $\mathsf{IN}(\phi^{(j-1)}, \delta)$, where δ is a tuning parameter and the proposal density is:

$$g(a \mid b, \delta) = \sqrt{\frac{\delta}{2\pi a^3}} \exp \left\{ -\frac{\delta (a-b)^2}{2b^2 a} \right\} \mathbb{1}_{(0,+\infty)}(a).$$

Let ϕ_{prop} be the proposed value for the j^{th} MCMC iteration.

2: **if** $u \leq \min \left\{ 1, \dfrac{f(\phi_{\mathsf{prop}} \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}) g(\phi^{(j-1)} \mid \phi_{\mathsf{prop}}, \delta)}{f(\phi^{(j-1)} \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}) g(\phi_{\mathsf{prop}} \mid \phi^{(j-1)}, \delta)} \right\}$ **then**

3: $\quad \phi^{(j)} \leftarrow \phi_{\mathsf{prop}}$.

4: **else**

5: $\quad \phi^{(j)} \leftarrow \phi^{(j-1)}$.

6: **end if**

2.2.3.5 Sampling from the full conditional distribution of D

The natural logarithm of the full conditional density of $\mathbf{D}$ is given by:

$$\begin{aligned}\ln f(\mathbf{D} \mid \mathbf{y}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma}) &= \text{constant} + \ln f(\mathbf{D}) + \sum_{t=1}^{T} \ln f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\ &\propto \left\{ -\frac{1}{2} \operatorname{tr}\left[(\mathbf{D}-\mathbf{S})^{\intercal} \boldsymbol{\sigma}_d^{-2} (\mathbf{D}-\mathbf{S}) \mathbf{R}_d^{-1} \right] \right\} \\ &+ \left\{ -\frac{Tq}{2} \ln \det \mathbf{B} - \frac{1}{2} \operatorname{tr} \left[\sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^{\intercal} (V \cdot \mathbf{B})^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t) \boldsymbol{\Sigma}^{-1} \right] \right\}. \end{aligned} \tag{2.11}$$

Since $f(\mathbf{D} \mid \mathbf{y}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$ does not have a known form, we propose the use of two algorithms to draw random samples from this probability distribution:

- *Metropolis-Hastings algorithm.* We sequentially update each component $D_{m,n} = d_m(\underset{\sim}{\mathbf{s}}_n)$ of $\mathbf{D}$ through the random walk Metropolis-Hastings algorithm for $m \in \{1, 2\}$ (coordinates) and $n \in \{1, \ldots, N\}$ (sites), in $2N$ sub-steps. The method is described in Algorithm 3 and its implementation is presented in Appendix B.1.2.5.1.

- *Slice sampler.* We sequentially update each component $D_{m,n} = d_m(\underset{\sim}{\mathbf{s}}_n)$ of $\mathbf{D}$ through the univariate slice sampler (Neal, 2003) for $m \in \{1, 2\}$ (coordinates) and $n \in \{3, \ldots, N\}$ (sites), in $2N$ sub-steps. The method is described in Algorithms 4-5 and its implementation is presented in Appendix B.1.2.5.2. In this method it is not necessary to tune variances, which tends to generate chains with a little less autocorrelation.

As $B_{n,n'} = \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'}) \|\}$ is a strictly decreasing function of $\| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'}) \|$, the model is identifiable in the sense that $B_{n,n'}$ is unique up to a scaling factor and $d(\cdot)$ is unique up to homothetic transformations[3] (Perrin and Meiring, 1999). Despite that, Damian et al. (2001, Sec. 2.1) point out that this does not ensure identifiability for a finite network of sites or monitoring stations.

To use spatial deformation in the Bayesian approach, Damian et al. (2001, Page 168), Morales et al. (2013, Sec. 2) and Sampson and Meiring (2014) suggest fixing the locations of two sites in $\mathcal{D}$-space to avoid the unidentifiability problem of $d(\cdot)$. To impose the anchor points constraint on the model, without loss of generality we pick an initial value $\mathbf{D}^{(0)}$ that contains $\underset{\sim}{\mathbf{s}}_1$ and $\underset{\sim}{\mathbf{s}}_2$ as its first two columns[4]. To guarantee $\mathbf{D}_1^{(j)} = \underset{\sim}{\mathbf{s}}_1$ and $\mathbf{D}_2^{(j)} = \underset{\sim}{\mathbf{s}}_2$ in all iterations $j \in \mathbb{N}$ of Algorithm 3,

[3] Translations, rotations, reflections about a line or combinations of those. For instance, $\tilde{\phi} = \frac{\phi}{\tau} > 0$ and $\tilde{d}(\underset{\sim}{\mathbf{s}}) = \tau^2 \cdot d(\underset{\sim}{\mathbf{s}})$ with $\tau > 0$ result in $\tilde{B}_{n,n'} = \exp\{-\tilde{\phi} \| \tilde{d}(\underset{\sim}{\mathbf{s}}_n) - \tilde{d}(\underset{\sim}{\mathbf{s}}_{n'}) \|\} = \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'}) \|\} = B_{n,n'}$.

[4] Could be any other two locations, since $\underset{\sim}{\mathbf{s}}_1, \ldots, \underset{\sim}{\mathbf{s}}_N$ have no order relation. The idea of fixing $\underset{\sim}{\mathbf{s}}_1$ and $\underset{\sim}{\mathbf{s}}_2$ is based on the algorithm presented in Damian et al. (2001, Page 168).

we work with a $2 \times N$ tuning matrix given by

$$\boldsymbol{\delta}^2 = \begin{bmatrix} 0 & 0 & \delta_{1,3}^2 & \delta_{1,4}^2 & \cdots & \delta_{1,N}^2 \\ 0 & 0 & \delta_{2,3}^2 & \delta_{2,4}^2 & \cdots & \delta_{2,N}^2 \end{bmatrix}_{2\times N}.$$

In our implementation of Algorithm 3 presented in Appendix B.1.2.5.1, we used the suggestion described in Givens and Hoeting (2012, Sec. 8.1.1) to adapt the tuning matrix $\boldsymbol{\delta}^2$ during the warm-up period of the Markov chains. The main idea of this strategy is to make acceptance rates for each component of $\mathbf{D}$ around 44%, as suggested by Givens and Hoeting (2012, Sec. 7.3.1.3).

2.2.3.6 Hybrid MCMC algorithm

We propose the use of a hybrid MCMC algorithm to generate samples from the posterior distribution of the unknown model parameters $\boldsymbol{\theta} = \{\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}, \phi, \mathbf{D}\}$, adding FFBS, slice sampler and Metropolis-Hastings steps in the Gibbs sampler. The method is described in Algorithm 6 and its implementation is presented in Appendix B.1.5. Theoretical aspects about some hybrid algorithms, such as Metropolis-within-Gibbs and slice-within-Gibbs, can be found in texts like Gamerman and Lopes (2006, Chapter 6) and Turkman et al. (2019, Chapter 6).

After reaching convergence (say after J iterations), Algorithm 6 is used to store K samples from the posterior distributions of the unknown quantities (say $\boldsymbol{\theta}^{(j_1)}, \ldots, \boldsymbol{\theta}^{(j_K)}$, where $j_1, \ldots, j_K \in \mathbb{N}$ are indices such that $J + 1 \leq j_1 < \cdots < j_K$). Thinning is used in the code presented in Appendix B.1.5, in order to avoid autocorrelation in the chains.

The model based on the vectorization of Equation (2.3) and its inference procedure are quite similar to the one discussed here. The results are presented in Appendix A.6.

2.3 Forecasting and interpolation

Expressions for forecasting and interpolation may be obtained through the predictive distribution and the conditional independence hypothesis. For a review, see Migon et al. (2014, Chapter 7) and Paulino et al. (2018, Chapter 3).

2.3.1 Forecasting

Here we wish to predict the response matrix at time $T + t$ for all N gauged sites, where T is the last observed time and t is a positive integer. Using the Markovian structure of the model

Algorithm 3 Metropolis-Hastings algorithm to sample from $f(\mathbf{D} \mid \mathbf{y}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$.

Start with $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$, ψ and $\boldsymbol{\theta}^{(j-1)} = \left\{\boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right\}$, where $j \in \mathbb{N}$ and $\boldsymbol{\beta}^{(j-1)} = \left\{\boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}\right\}$.

1: Set a tuning matrix $\boldsymbol{\delta}^2 = \left[\ \delta^2_{m,n}\ \right]_{2\times N}$ and generate $2N$ random numbers in $(0,1)$, denoting them by $u_{1,1}, \ldots, u_{1,N}, u_{2,1}, \ldots, u_{2,N}$.

2: Update $\mathbf{D}^{(j-1)} = \left[\ D^{(j-1)}_{m,n}\ \right]_{2\times N}$ by $\mathbf{D}^{(j)} = \left[\ D^{(j)}_{m,n}\ \right]_{2\times N}$ proceeding as follows:

- Propose a new value $D^{\mathsf{prop}}_{1,1}$ from the distribution $\mathsf{N}(D^{(j-1)}_{1,1}, \delta^2_{1,1})$ and update this component:

$$D^{(j)}_{1,1} = \begin{cases} D^{\mathsf{prop}}_{1,1}, & u_{1,1} \leq \min\left\{1, \dfrac{f\left(\begin{bmatrix} D^{\mathsf{prop}}_{1,1} & D^{(j-1)}_{1,2} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & \cdots & D^{(j-1)}_{2,N} \end{bmatrix} \middle| \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)}{f\left(\begin{bmatrix} D^{(j-1)}_{1,1} & D^{(j-1)}_{1,2} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & \cdots & D^{(j-1)}_{2,N} \end{bmatrix} \middle| \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)}\right\} \\ D^{(j-1)}_{1,1}, & \text{otherwise} \end{cases}.$$

- Propose a new value $D^{\mathsf{prop}}_{1,2}$ from the distribution $\mathsf{N}(D^{(j-1)}_{1,2}, \delta^2_{1,2})$ and update this component:

$$D^{(j)}_{1,2} = \begin{cases} D^{\mathsf{prop}}_{1,2}, & u_{1,2} \leq \min\left\{1, \dfrac{f\left(\begin{bmatrix} D^{(j)}_{1,1} & D^{\mathsf{prop}}_{1,2} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & \cdots & D^{(j-1)}_{2,N} \end{bmatrix} \middle| \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)}{f\left(\begin{bmatrix} D^{(j)}_{1,1} & D^{(j-1)}_{1,2} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & \cdots & D^{(j-1)}_{2,N} \end{bmatrix} \middle| \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)}\right\} \\ D^{(j-1)}_{1,2}, & \text{otherwise} \end{cases}.$$

- $\vdots$

- Propose a new value $D^{\mathsf{prop}}_{2,N}$ from the distribution $\mathsf{N}(D^{(j-1)}_{2,N}, \delta^2_{2,N})$ and update this component:

$$D^{(j)}_{2,N} = \begin{cases} D^{\mathsf{prop}}_{2,N}, & u_{2,N} \leq \min\left\{1, \dfrac{f\left(\begin{bmatrix} D^{(j)}_{1,1} & \cdots & D^{(j)}_{1,N-1} & D^{(j)}_{1,N} \\ D^{(j)}_{2,1} & \cdots & D^{(j)}_{2,N-1} & D^{\mathsf{prop}}_{2,N} \end{bmatrix} \middle| \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)}{f\left(\begin{bmatrix} D^{(j)}_{1,1} & \cdots & D^{(j)}_{1,N-1} & D^{(j)}_{1,N} \\ D^{(j)}_{2,1} & \cdots & D^{(j)}_{2,N-1} & D^{(j-1)}_{2,N} \end{bmatrix} \middle| \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)}\right\} \\ D^{(j-1)}_{2,N}, & \text{otherwise} \end{cases}.$$

Algorithm 4 Slice sampling algorithm to sample from $f(\mathbf{D} \mid \mathbf{y}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$ – Part 1.

Start with $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$, ψ and $\boldsymbol{\theta}^{(j-1)} = \left\{\boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right\}$, where $j \in \mathbb{N}$ and $\boldsymbol{\beta}^{(j-1)} = \left\{\boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}\right\}$.
Update $\mathbf{D}^{(j-1)} = \left[\ D_{m,n}^{(j-1)}\ \right]_{2\times N}$ by $\mathbf{D}^{(j)} = \left[\ D_{m,n}^{(j)}\ \right]_{2\times N}$ proceeding as follows:

- Do $D_{1,1}^{(j)} \leftarrow D_{1,1}^{(j-1)}$ and $D_{1,2}^{(j)} \leftarrow D_{1,2}^{(j-1)}$.
- Obtain $D_{1,3}^{(j)}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\left[\begin{smallmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j-1)} & \cdots & D_{1,N}^{(j-1)} \\ D_{2,1}^{(j-1)} & D_{2,2}^{(j-1)} & D_{2,3}^{(j-1)} & \cdots & D_{2,N}^{(j-1)} \end{smallmatrix}\right] \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{1,3} : f\left(\left[\begin{smallmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3} & \cdots & D_{1,N}^{(j-1)} \\ D_{2,1}^{(j-1)} & D_{2,2}^{(j-1)} & D_{2,3}^{(j-1)} & \cdots & D_{2,N}^{(j-1)} \end{smallmatrix}\right] \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D_{1,3}^{(j-1)}$ that contains all, or much, of the slice.
  3. Draw the new point, $D_{1,3}^{(j)}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.
- $\vdots$
- Obtain $D_{1,N}^{(j)}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\left[\begin{smallmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j)} & \cdots & D_{1,N}^{(j-1)} \\ D_{2,1}^{(j-1)} & D_{2,2}^{(j-1)} & D_{2,3}^{(j-1)} & \cdots & D_{2,N}^{(j-1)} \end{smallmatrix}\right] \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{1,N} : f\left(\left[\begin{smallmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j)} & \cdots & D_{1,N} \\ D_{2,1}^{(j-1)} & D_{2,2}^{(j-1)} & D_{2,3}^{(j-1)} & \cdots & D_{2,N}^{(j-1)} \end{smallmatrix}\right] \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D_{1,N}^{(j-1)}$ that contains all, or much, of the slice.
  3. Draw the new point, $D_{1,N}^{(j)}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.

Algorithm 5 Slice sampling algorithm to sample from $f(\mathbf{D} \mid \mathbf{y}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$ – Part 2.

- Do $D_{2,1}^{(j)} \leftarrow D_{2,1}^{(j-1)}$ and $D_{2,2}^{(j)} \leftarrow D_{2,2}^{(j-1)}$.
- Obtain $D_{2,3}^{(j)}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j)} & \cdots & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & D_{2,3}^{(j-1)} & \cdots & D_{2,N}^{(j-1)} \end{bmatrix} \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{2,3} : f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j)} & \cdots & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & D_{2,3} & \cdots & D_{2,N}^{(j-1)} \end{bmatrix} \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D_{2,3}^{(j-1)}$ that contains all, or much, of the slice.
  3. Draw the new point, $D_{2,3}^{(j)}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.
- $\vdots$
- Obtain $D_{2,N}^{(j)}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & \cdots & D_{1,N-1}^{(j)} & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & \cdots & D_{1,N-1}^{(j)} & D_{2,N}^{(j-1)} \end{bmatrix} \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{2,N} : f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & \cdots & D_{1,N-1}^{(j)} & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & \cdots & D_{2,N-1}^{(j)} & D_{2,N} \end{bmatrix} \mid \mathbf{y}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D_{2,N}^{(j-1)}$ that contains all, or much, of the slice.
  3. Draw the new point, $D_{2,N}^{(j)}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.

Algorithm 6 Hybrid MCMC algorithm to sample from $f(V, \boldsymbol{\Sigma}, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T, \phi, \mathbf{D} \mid \mathbf{y})$.

1: Set initial values for the parameters (i.e. $\boldsymbol{\theta}^{(0)} = \{V^{(0)}, \boldsymbol{\Sigma}^{(0)}, \boldsymbol{\beta}_0^{(0)}, \boldsymbol{\beta}_1^{(0)}, \ldots, \boldsymbol{\beta}_T^{(0)}, \phi^{(0)}, \mathbf{D}^{(0)}\} \in \boldsymbol{\Theta}$) and do $j \leftarrow 1$. ▷ Fix the locations of two sites in $\mathbf{D}^{(0)}$, e.g. pick $\mathbf{D}^{(0)} = \mathbf{S}$.

2: **repeat**

3: Generate $V^{(j)}$ and $\boldsymbol{\Sigma}^{(j)}$ from their known full conditional distributions given in Equations (2.7) and (2.8), respectively.

4: Run Algorithm 1 to sample $\boldsymbol{\beta}_0^{(j)}, \boldsymbol{\beta}_1^{(j)}, \ldots, \boldsymbol{\beta}_T^{(j)}$ through the FFBS algorithm.

5: Run Algorithm 2 to sample $\phi^{(j)}$ through the Metropolis-Hastings algorithm.

6: Run Algorithms 4-5 to sample $\mathbf{D}^{(j)}$ through the slice sampler. ▷ Alternatively, run Algorithm 3 to sample $\mathbf{D}^{(j)}$ through the Metropolis-Hastings algorithm in this 6^{th} line.

7: Set $j \leftarrow j + 1$.

8: **until** convergence is reached.

and supposing that $\mathbf{G}_{T+1}, \mathbf{G}_{T+2}, \ldots, \mathbf{G}_{T+t}$ are t fixed $p \times p$ evolution matrices, one may write the following evolution equations:

$$\begin{array}{rcl@{\quad}l@{\quad}l@{\quad}l@{\quad}l}
\boldsymbol{\beta}_{T+1} & = & \mathbf{G}_{T+1}\boldsymbol{\beta}_T & + & \boldsymbol{\omega}_{T+1}, & \boldsymbol{\omega}_{T+1} \mid V, \boldsymbol{\Sigma} & \sim & \mathsf{N}_{p\times q}(\mathbf{0}_{p\times q}, V \cdot \mathbf{W}, \boldsymbol{\Sigma}), \\
\boldsymbol{\beta}_{T+2} & = & \mathbf{G}_{T+2}\boldsymbol{\beta}_{T+1} & + & \boldsymbol{\omega}_{T+2}, & \boldsymbol{\omega}_{T+2} \mid V, \boldsymbol{\Sigma} & \sim & \mathsf{N}_{p\times q}(\mathbf{0}_{p\times q}, V \cdot \mathbf{W}, \boldsymbol{\Sigma}), \\
\vdots & = & \vdots & + & \vdots & \vdots & \sim & \vdots \\
\boldsymbol{\beta}_{T+t} & = & \mathbf{G}_{T+t}\boldsymbol{\beta}_{T+t-1} & + & \boldsymbol{\omega}_{T+t}, & \boldsymbol{\omega}_{T+t} \mid V, \boldsymbol{\Sigma} & \sim & \mathsf{N}_{p\times q}(\mathbf{0}_{p\times q}, V \cdot \mathbf{W}, \boldsymbol{\Sigma}),
\end{array}$$

whose recursive relation implies for $t > 1$ the following:

$$\boldsymbol{\beta}_{T+t} = \mathbf{G}_{T+t} \cdots \mathbf{G}_{T+2}\mathbf{G}_{T+1}\boldsymbol{\beta}_T + \mathbf{G}_{T+t} \cdots \mathbf{G}_{T+3}\mathbf{G}_{T+2}\boldsymbol{\omega}_{T+1} + \cdots + \mathbf{G}_{T+t}\boldsymbol{\omega}_{T+t-1} + \boldsymbol{\omega}_{T+t}.$$

Notice that $\boldsymbol{\beta}_{T+1} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma} \sim \mathsf{N}_{p\times q}(\mathbf{G}_{T+1}\boldsymbol{\beta}_T, V \cdot \mathbf{W}, \boldsymbol{\Sigma})$. Suppose that $\boldsymbol{\omega}_{T+1}, \boldsymbol{\omega}_{T+2}, \ldots$ is a sequence of conditionally independent random matrices given V and $\boldsymbol{\Sigma}$. Applying the matrix-normal theory (Quintana, 1987; Gupta and Nagar, 2000), for $t > 1$ we have:

$$\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma} \sim \mathsf{N}_{p\times q}\left(\left[\prod_{t'=t}^{1} \mathbf{G}_{T+t'}\right]\boldsymbol{\beta}_T, V \cdot \left\{\mathbf{W} + \sum_{t'=t}^{2}\left[\prod_{t''=t}^{t'} \mathbf{G}_{T+t''}\right]^{\mathsf{T}} \mathbf{W} \left[\prod_{t''=t}^{t'} \mathbf{G}_{T+t''}\right]\right\}, \boldsymbol{\Sigma}\right).$$

Let $\mathbf{X}_{T+t}$ be a fixed $N \times p$ matrix for $t \in \mathbb{N}$. We may also assume the following:

$$\mathbf{Y}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N\times q}(\mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t}, V \cdot \mathbf{B}, \boldsymbol{\Sigma}). \tag{2.12}$$

Let $\mathbf{Y}_{\mathsf{pred}} = \{\mathbf{Y}_{T+1}, \ldots, \mathbf{Y}_{T+T^\star}\}$ be the collection of $T^\star$ response matrices to forecast after the last observed time. Now the collection of unknown parameters is augmented to $\boldsymbol{\theta}_{\mathsf{pred}} = \{\boldsymbol{\theta}, \boldsymbol{\beta}_{\mathsf{pred}}\}$,

where $\boldsymbol{\beta}_{\mathsf{pred}} = \{\boldsymbol{\beta}_{T+1}, \ldots, \boldsymbol{\beta}_{T+T^\star}\}$, with parametric space $\boldsymbol{\Theta}_{\mathsf{pred}} = \boldsymbol{\Theta} \times \left(\prod_{t=1}^{T^\star} \mathbb{R}^{p \times q} \right)$. One can forecast these T^* response matrices after the last observed time using the following predictive density:

$$
\begin{aligned}
f(\mathbf{y}_{\mathsf{pred}} \mid \mathbf{y}) &= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} f(\mathbf{y}_{\mathsf{pred}}, \boldsymbol{\theta}_{\mathsf{pred}} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{pred}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} \frac{f(\mathbf{y}_{\mathsf{pred}}, \boldsymbol{\theta}_{\mathsf{pred}}, \mathbf{y})}{f(\boldsymbol{\theta}_{\mathsf{pred}}, \mathbf{y})} \cdot \frac{f(\boldsymbol{\theta}_{\mathsf{pred}}, \mathbf{y})}{f(\mathbf{y})} \partial \boldsymbol{\theta}_{\mathsf{pred}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} f(\mathbf{y}_{\mathsf{pred}} \mid \boldsymbol{\beta}_{\mathsf{pred}}, \boldsymbol{\theta}, \mathbf{y}) \cdot \frac{f(\boldsymbol{\beta}_{\mathsf{pred}} \mid \boldsymbol{\theta}, \mathbf{y}) f(\boldsymbol{\theta} \mid \mathbf{y}) f(\mathbf{y})}{f(\mathbf{y})} \partial \boldsymbol{\theta}_{\mathsf{pred}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} \left[\prod_{t=1}^{T^\star} f(\mathbf{y}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}) \right] f(\boldsymbol{\theta} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{pred}}.
\end{aligned}
\tag{2.13}
$$

Since the integral that appears in Equation (2.14) is not analytically tractable, we may use Monte Carlo methods to compute it by adding extra steps to Algorithm 6. The procedure is described in Algorithm 7 and its implementation is shown in Appendix B.1.6.1.

2.3.2 Interpolation

2.3.2.1 Interpolation for observed times

Let $\underset{\sim}{\mathbf{s}}_{N+1}, \ldots, \underset{\sim}{\mathbf{s}}_{N+N^*}$ be N^* ungauged coordinates that we wish to predict in $\mathcal{D}$-space, denoting by $\mathbf{S}^* = \begin{bmatrix} \underset{\sim}{\mathbf{s}}_{N+1} & \cdots & \underset{\sim}{\mathbf{s}}_{N+N^*} \end{bmatrix}$ the $2 \times N^*$ matrix formed by the N^* ungauged sites. Suppose that $\mathbf{D}^* = \begin{bmatrix} \mathbf{D}_{N+1} & \cdots & \mathbf{D}_{N+N^*} \end{bmatrix} \sim \mathsf{N}_{2 \times N^*}(\mathbf{S}^*, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^*)$, where $\mathbf{D}_{N+n} = d(\underset{\sim}{\mathbf{s}}_{N+n}) \in \mathcal{D} \subset \mathbb{R}^2$ for all $n \in \{1, \ldots, N^*\}$ and $\mathbf{R}_d^* = \begin{bmatrix} R_{n,n'}^* \end{bmatrix}_{N^* \times N^*}$ is a matrix such that $R_{n,n'}^* = \exp\{-\psi \|\underset{\sim}{\mathbf{s}}_{N+n} - \underset{\sim}{\mathbf{s}}_{N+n'}\|^2\}$ for all $n, n' \in \{1, \ldots, N^*\}$. Define $\mathbf{R}_{\mathsf{g,u}} = \begin{bmatrix} R_{n,n'}^{\mathsf{g,u}} \end{bmatrix}_{N \times N^*}$ such that $R_{n,n'}^{\mathsf{g,u}} = \exp\{-\psi \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{N+n'}\|^2\}$ for all $n \in \{1, \ldots, N\}$ and $n' \in \{1, \ldots, N^*\}$, where $\mathbf{R}_{\mathsf{u,g}}$ is its transpose. Applying the matrix-normal theory (Quintana, 1987, Eqs. A3.2.4 and A3.2.8), from the joint distribution

$$
\begin{bmatrix} \mathbf{D} & \mathbf{D}^* \end{bmatrix} \sim \mathsf{N}_{2 \times (N+N^*)} \left(\begin{bmatrix} \mathbf{S} & \mathbf{S}^* \end{bmatrix}, \boldsymbol{\sigma}_d^2, \begin{bmatrix} \mathbf{R}_d & \mathbf{R}_{\mathsf{g,u}} \\ \mathbf{R}_{\mathsf{u,g}} & \mathbf{R}_d^* \end{bmatrix} \right),
$$

one can obtain the following expressions:

$$
\begin{aligned}
\begin{bmatrix} \mathbf{D}^\intercal \\ \{\mathbf{D}^*\}^\intercal \end{bmatrix} &\sim \mathsf{N}_{(N+N^*) \times 2} \left(\begin{bmatrix} \mathbf{S}^\intercal \\ \{\mathbf{S}^*\}^\intercal \end{bmatrix}, \begin{bmatrix} \mathbf{R}_d & \mathbf{R}_{\mathsf{g,u}} \\ \mathbf{R}_{\mathsf{u,g}} & \mathbf{R}_d^* \end{bmatrix}, \boldsymbol{\sigma}_d^2 \right), \\
\{\mathbf{D}^*\}^\intercal \mid \mathbf{D} &\sim \mathsf{N}_{N^* \times 2}(\{\mathbf{S}^*\}^\intercal + \mathbf{R}_{\mathsf{u,g}} \mathbf{R}_d^{-1} (\mathbf{D}^\intercal - \mathbf{S}^\intercal), \mathbf{R}_d^* - \mathbf{R}_{\mathsf{u,g}} \mathbf{R}_d^{-1} \mathbf{R}_{\mathsf{g,u}}, \boldsymbol{\sigma}_d^2), \\
\mathbf{D}^* \mid \mathbf{D} &\sim \mathsf{N}_{2 \times N^*}(\mathbf{S}^* + (\mathbf{D} - \mathbf{S}) \mathbf{R}_d^{-1} \mathbf{R}_{\mathsf{g,u}}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^* - \mathbf{R}_{\mathsf{u,g}} \mathbf{R}_d^{-1} \mathbf{R}_{\mathsf{g,u}}).
\end{aligned}
\tag{2.14}
$$

Algorithm 7 Monte Carlo integration to approximate $f(\mathbf{y}_{\mathsf{pred}} \mid \mathbf{y})$.

Start with K MCMC samples, where $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$ is the $j_k{}^{\text{th}}$ MCMC sample.

for $k \leftarrow 1$ to K **do**

  Compute $\mathbf{B}_{(j_k)} = \left[\ B_{n,n'}^{(j_k)}\ \right]_{N \times N}$, where $B_{n,n'}^{(j_k)} = \exp\{-\phi^{(j_k)} \| d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{n'}) \|\}$ for all $n, n' \in \{1, \ldots, N\}$.

  for $t \leftarrow 1$ to $T^\star$ **do**

    if $t \leftarrow 1$ **or** $T^\star \leftarrow 1$ **then**

      Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{p \times q}(\mathbf{G}_{T+1}\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \mathbf{W}, \boldsymbol{\Sigma}^{(j_k)})$.

    else

      Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the following distribution:

$$\mathsf{N}_{p \times q}\left(\left[\prod_{t'=t}^{1} \mathbf{G}_{T+t'}\right]\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \left\{\mathbf{W} + \sum_{t'=t}^{2}\left[\prod_{t''=t}^{t'} \mathbf{G}_{T+t''}\right]^{\mathsf{T}} \mathbf{W} \left[\prod_{t''=t}^{t'} \mathbf{G}_{T+t''}\right]\right\}, \boldsymbol{\Sigma}^{(j_k)}\right).$$

    end if

    Sample $\mathbf{y}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{N \times q}(\mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t}^{(j_k)}, V^{(j_k)} \cdot \mathbf{B}_{(j_k)}, \boldsymbol{\Sigma}^{(j_k)})$.

  end for

end for

Let $d(\underset{\sim}{\mathbf{s}}_{N+1}), \ldots, d(\underset{\sim}{\mathbf{s}}_{N+N^*})$ be the N^* corresponding predicted coordinates of $\underset{\sim}{\mathbf{s}}_{N+1}, \ldots, \underset{\sim}{\mathbf{s}}_{N+N^*}$ in $\mathcal{D}$-space, respectively. Suppose that $\mathbf{Y}_t^* \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{N^* \times q}(\mathbf{X}_t^* \boldsymbol{\beta}_t, V \cdot \mathbf{B}^*, \boldsymbol{\Sigma})$, where $\mathbf{X}_t^*$ is a fixed $N^* \times p$ matrix and $\mathbf{B}^* = \left[\begin{array}{c} B_{n,n'}^* \end{array} \right]_{N^* \times N^*}$ is a matrix such that $B_{n,n'}^* = \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_{N+n}) - d(\underset{\sim}{\mathbf{s}}_{N+n'}) \|\}$ for all $n, n' \in \{1, \ldots, N^*\}$. Define $\mathbf{B}_{\mathsf{g},\mathsf{u}} = \left[\begin{array}{c} B_{n,n'}^{\mathsf{g},\mathsf{u}} \end{array} \right]_{N \times N^*}$ such that $B_{n,n'}^{\mathsf{g},\mathsf{u}} = \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{N+n'}) \|\}$ for all $n \in \{1, \ldots, N\}$ and $n' \in \{1, \ldots, N^*\}$, where $\mathbf{B}_{\mathsf{u},\mathsf{g}}$ is its transpose. Applying the matrix-normal theory again, from the joint distribution

$$\left[\begin{array}{c} \mathbf{Y}_t \\ \mathbf{Y}_t^* \end{array} \right] \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{(N+N^*) \times q} \left(\left[\begin{array}{c} \mathbf{X}_t \boldsymbol{\beta}_t \\ \mathbf{X}_t^* \boldsymbol{\beta}_t \end{array} \right], V \cdot \left[\begin{array}{cc} \mathbf{B} & \mathbf{B}_{\mathsf{g},\mathsf{u}} \\ \mathbf{B}_{\mathsf{u},\mathsf{g}} & \mathbf{B}^* \end{array} \right], \boldsymbol{\Sigma} \right),$$

one can obtain the following expression:

$$\mathbf{Y}_t^* \mid \mathbf{Y}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{N^* \times q}(\mathbf{X}_t^* \boldsymbol{\beta}_t + \mathbf{B}_{\mathsf{u},\mathsf{g}} \mathbf{B}^{-1} (\mathbf{Y}_t - \mathbf{X}_t \boldsymbol{\beta}_t), V \cdot (\mathbf{B}^* - \mathbf{B}_{\mathsf{u},\mathsf{g}} \mathbf{B}^{-1} \mathbf{B}_{\mathsf{g},\mathsf{u}}), \boldsymbol{\Sigma}). \quad (2.15)$$

Now the collection of unknown parameters is augmented to $\boldsymbol{\theta}_{\mathsf{int}} = \{\boldsymbol{\theta}, \mathbf{D}^*\}$, with parametric space $\boldsymbol{\Theta}_{\mathsf{int}} = \boldsymbol{\Theta} \times \mathbb{R}^{2 \times N^*}$. Define $\mathbf{Y}_{\mathsf{int}} = \{\mathbf{Y}_1^*, \ldots, \mathbf{Y}_T^*\}$. For observed times, one can interpolate the response matrices using the following predictive density:

$$\begin{aligned} f(\mathbf{y}_{\mathsf{int}} \mid \mathbf{y}) &= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} f(\mathbf{y}_{\mathsf{int}}, \boldsymbol{\theta}_{\mathsf{int}} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{int}} \\ &= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} \frac{f(\mathbf{y}_{\mathsf{int}}, \boldsymbol{\theta}_{\mathsf{int}}, \mathbf{y})}{f(\boldsymbol{\theta}_{\mathsf{int}}, \mathbf{y})} \cdot \frac{f(\boldsymbol{\theta}_{\mathsf{int}}, \mathbf{y})}{f(\mathbf{y})} \partial \boldsymbol{\theta}_{\mathsf{int}} \\ &= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} f(\mathbf{y}_{\mathsf{int}} \mid \mathbf{D}^*, \boldsymbol{\theta}, \mathbf{y}) \cdot \frac{f(\mathbf{D}^* \mid \boldsymbol{\theta}, \mathbf{y}) f(\boldsymbol{\theta} \mid \mathbf{y}) f(\mathbf{y})}{f(\mathbf{y})} \partial \boldsymbol{\theta}_{\mathsf{int}} \\ &= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} \left[\prod_{t=1}^{T} f(\mathbf{y}_t^* \mid \mathbf{y}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}) \right] f(\mathbf{D}^* \mid \mathbf{D}) f(\boldsymbol{\theta} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{int}}. \end{aligned} \quad (2.16)$$

Since the integral that appears in Equation (2.16) is not analytically tractable, we may use Monte Carlo methods to compute it by adding extra steps to Algorithm 6. The procedure is described in Algorithm 8 and implemented in Appendix B.1.6.2.

Recall that the role of hyperparameters $\boldsymbol{\sigma}_d^2$ and ψ were discussed in Section 2.2.1. Notice that $\psi \to +\infty$ would not be a suitable choice because it would lead to $\mathbf{R}_{\mathsf{g},\mathsf{u}} \to \mathbf{0}_{N \times N^*}$. Consequently, $\mathbf{D}_{(j_k)}^*$ would have no dependency on $\mathbf{D}^{(j_k)}$ in Step 2 of Algorithm 8 (i.e. $\mathbf{D}_{(j_k)}^*$ would be sampled from the distribution $\mathsf{N}_{2 \times N^*}(\mathbf{S}^*, \boldsymbol{\sigma}_d^2, \mathbf{I}_{N^*})$). On the other hand, $\psi \to 0$ would lead to the case $\mathbf{R}_{\mathsf{g},\mathsf{u}} \to \mathbf{1}_{N \times N^*}$ (not useful, since $\mathbf{R}_d^*$ needs to be invertible). Morales and Vicini (2020) recommend evaluating different combinations of $\boldsymbol{\sigma}_d^2$ and ψ (related to $\mathbf{R}_d$, $\mathbf{R}_{\mathsf{g},\mathsf{u}}$ and $\mathbf{R}_d^*$). Our suggestion is to fix $\boldsymbol{\sigma}_d^2$ as the empirical covariance matrix of the gauged sites and pick $\psi > 0$ based on three criteria:

Algorithm 8 Monte Carlo integration to approximate $f(\mathbf{y}_{\mathsf{int}} \mid \mathbf{y})$.

Start with K MCMC samples and also with the hyperparameters $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$ and ψ, where the k^{th} MCMC sample is given by $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$.

1: **for** $k \leftarrow 1$ to K **do**

2: Sample $\mathbf{D}^*_{(j_k)}$ from the distribution $\mathsf{N}_{2\times N^*}(\mathbf{S}^* + (\mathbf{D}^{(j_k)} - \mathbf{S})\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}, \boldsymbol{\sigma}^2_d, \mathbf{R}^*_d - \mathbf{R}_{\mathsf{u},\mathsf{g}}\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}})$.

3: Compute the matrices $\mathbf{B}_{(j_k)} = \left[\ B^{(j_k)}_{n,n'}\ \right]_{N\times N}$, $\mathbf{B}^{(j_k)}_{\mathsf{g},\mathsf{u}} = \left[\ B^{\mathsf{g},\mathsf{u}^{(j_k)}}_{n,n'}\ \right]_{N\times N^*}$ ($\mathbf{B}^{(j_k)}_{\mathsf{u},\mathsf{g}}$ is its transpose) and $\mathbf{B}^*_{(j_k)} = \left[\ B^{*(j_k)}_{n,n'}\ \right]_{N^*\times N^*}$, where:

$$
\begin{aligned}
B^{(j_k)}_{n,n'} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\mathbf{s}_n) - d^{(j_k)}(\mathbf{s}_{n'})\|\}, && n, n' \in \{1,\ldots,N\},\\
B^{\mathsf{g},\mathsf{u}^{(j_k)}}_{n,n'} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\mathbf{s}_n) - d^{(j_k)}(\mathbf{s}_{N+n'})\|\}, && n \in \{1,\ldots,N\}, n' \in \{1,\ldots,N^*\},\\
B^{*(j_k)}_{n,n'} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\mathbf{s}_{N+n}) - d^{(j_k)}(\mathbf{s}_{N+n'})\|\}, && n, n' \in \{1,\ldots,N^*\}.
\end{aligned}
$$

4: **for** $t \leftarrow 1$ to T **do**

5: Sample $\mathbf{y}_t^{*(j_k)}$ from the following distribution:

$$\mathsf{N}_{N^*\times q}(\mathbf{X}^*_t\boldsymbol{\beta}_t^{(j_k)} + \mathbf{B}^{(j_k)}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}_{(j_k)}(\mathbf{y}_t - \mathbf{X}_t\boldsymbol{\beta}_t^{(j_k)}), V^{(j_k)}\cdot(\mathbf{B}^*_{(j_k)} - \mathbf{B}^{(j_k)}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}_{(j_k)}\mathbf{B}^{(j_k)}_{\mathsf{g},\mathsf{u}}), \boldsymbol{\Sigma}^{(j_k)}).$$

6: **end for**

7: **end for**

fast mixing chains (see Gamerman and Lopes, 2006, Example 7.10), good performance in the interpolation of the response matrices (this will be discussed in Section 2.4), and the interpretability of the estimated deformation (e.g. not too many folds).

2.3.2.2 Interpolation after last observed time

Here we wish to predict the response matrix at time $T+t$ in N^* ungauged sites of interest, where T is the last observed time and t is a positive integer. To do so, we will use the conditional distributions $\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}$ (for $t=1$ and $t \in \mathbb{N}\backslash\{1\}$) and $\mathbf{D}^* \mid \mathbf{D}$ obtained in Sections 2.3.1 and 2.3.2.1, respectively.

Let $d(\underset{\sim}{\mathbf{s}}_{N+1}), \ldots, d(\underset{\sim}{\mathbf{s}}_{N+N^*})$ be the N^* corresponding predicted coordinates of $\underset{\sim}{\mathbf{s}}_{N+1}, \ldots, \underset{\sim}{\mathbf{s}}_{N+N^*}$ in $\mathcal{D}$-space, respectively. Suppose that $\mathbf{Y}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N\times q}(\mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t}, V \cdot \mathbf{B}, \boldsymbol{\Sigma})$ and $\mathbf{Y}^*_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{N^*\times q}(\mathbf{X}^*_{T+t}\boldsymbol{\beta}_{T+t}, V \cdot \mathbf{B}^*, \boldsymbol{\Sigma})$ for $t \in \mathbb{N}$, where $\mathbf{X}_{T+t}$ and $\mathbf{X}^*_{T+t}$ are two fixed $N \times p$ and $N^* \times p$ matrices, respectively, and $\mathbf{B}^* = \left[\, B^*_{n,n'} \,\right]_{N^*\times N^*}$ is a matrix such that $B^*_{n,n'} = \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_{N+n}) - d(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}$ for all $n, n' \in \{1, \ldots, N^*\}$. Define $\mathbf{B}_{\mathsf{g},\mathsf{u}} = \left[\, B^{\mathsf{g},\mathsf{u}}_{n,n'} \,\right]_{N\times N^*}$ such that $B^{\mathsf{g},\mathsf{u}}_{n,n'} = \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_{n}) - d(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}$ for all $n \in \{1, \ldots, N\}$ and $n' \in \{1, \ldots, N^*\}$, where $\mathbf{B}_{\mathsf{u},\mathsf{g}}$ is its transpose. Applying the matrix-normal theory (Quintana, 1987; Gupta and Nagar, 2000), from the joint distribution

$$\begin{bmatrix} \mathbf{Y}_{T+t} \\ \mathbf{Y}^*_{T+t} \end{bmatrix} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{(N+N^*)\times q}\left(\begin{bmatrix} \mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t} \\ \mathbf{X}^*_{T+t}\boldsymbol{\beta}_{T+t} \end{bmatrix}, V \cdot \begin{bmatrix} \mathbf{B} & \mathbf{B}_{\mathsf{g},\mathsf{u}} \\ \mathbf{B}_{\mathsf{u},\mathsf{g}} & \mathbf{B}^* \end{bmatrix}, \boldsymbol{\Sigma} \right),$$

one can obtain the following expression:

$$\mathbf{Y}^*_{T+t} \mid \mathbf{Y}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{N^*\times q}(\boldsymbol{\mu}_{\mathbf{Y}^*_{T+t}|\mathbf{Y}_{T+t},\boldsymbol{\beta}_{T+t},V,\phi,\mathbf{D},\mathbf{D}^*,\boldsymbol{\Sigma}}, V\cdot(\mathbf{B}^* - \mathbf{B}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}\mathbf{B}_{\mathsf{g},\mathsf{u}}), \boldsymbol{\Sigma}), \quad (2.17)$$

where

$$\begin{aligned} \boldsymbol{\mu}_{\mathbf{Y}^*_{T+t}|\mathbf{Y}_{T+t},\boldsymbol{\beta}_{T+t},V,\phi,\mathbf{D},\mathbf{D}^*,\boldsymbol{\Sigma}} &= \mathrm{E}[\mathbf{Y}^*_{T+t} \mid \mathbf{Y}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}] \\ &= \mathbf{X}^*_{T+t}\boldsymbol{\beta}_{T+t} + \mathbf{B}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}(\mathbf{Y}_{T+t} - \mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t}). \end{aligned}$$

Let $\mathbf{Y}_{\mathsf{aug}} = \{\mathbf{Y}^*_{T+1}, \ldots, \mathbf{Y}^*_{T+T^\star}\}$ be the collection of $T^\star$ matrix-variate responses to forecast after the last observed time in N^* ungauged sites of interest. Now the collection of unknown parameters is augmented to $\boldsymbol{\theta}_{\mathsf{aug}} = \{\boldsymbol{\theta}, \mathbf{D}^*, \boldsymbol{\beta}_{\mathsf{pred}}, \mathbf{Y}_{\mathsf{pred}}\}$, where $\mathbf{D}^* = \begin{bmatrix} \mathbf{D}_{N+1} & \cdots & \mathbf{D}_{N+N^*} \end{bmatrix}$, $\boldsymbol{\beta}_{\mathsf{pred}} = \{\boldsymbol{\beta}_{T+1}, \ldots, \boldsymbol{\beta}_{T+T^\star}\}$ and $\mathbf{Y}_{\mathsf{pred}} = \{\mathbf{Y}_{T+1}, \ldots, \mathbf{Y}_{T+T^\star}\}$ is treated as a collection of unknown quantities, with parametric space $\boldsymbol{\Theta}_{\mathsf{aug}} = \boldsymbol{\Theta} \times \mathbb{R}^{2\times N^*} \times \left(\prod_{t=1}^{T^\star} \mathbb{R}^{p\times q}\right) \times \left(\prod_{t=1}^{T^\star} \mathbb{R}^{N\times q}\right)$. One can forecast

and interpolate responses simultaneously using the following predictive density:

$$
\begin{aligned}
f(\mathbf{y}_{\mathsf{aug}} \mid \mathbf{y}) &= \int\limits_{\boldsymbol{\Theta}_{\mathsf{aug}}} f(\mathbf{y}_{\mathsf{aug}}, \boldsymbol{\theta}_{\mathsf{aug}} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int\limits_{\boldsymbol{\Theta}_{\mathsf{aug}}} \frac{f(\mathbf{y}_{\mathsf{aug}}, \boldsymbol{\theta}_{\mathsf{aug}}, \mathbf{y})}{f(\boldsymbol{\theta}_{\mathsf{aug}}, \mathbf{y})} \cdot \frac{f(\boldsymbol{\theta}_{\mathsf{aug}}, \mathbf{y})}{f(\mathbf{y})} \partial \boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int\limits_{\boldsymbol{\Theta}_{\mathsf{aug}}} f(\mathbf{y}_{\mathsf{aug}} \mid \boldsymbol{\theta}_{\mathsf{aug}}, \mathbf{y}) f(\boldsymbol{\theta}_{\mathsf{aug}} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int\limits_{\boldsymbol{\Theta}_{\mathsf{aug}}} f(\mathbf{y}_{\mathsf{aug}} \mid \mathbf{y}_{\mathsf{pred}}, \boldsymbol{\beta}_{\mathsf{pred}}, \mathbf{D}^*, \boldsymbol{\theta}) f(\mathbf{y}_{\mathsf{pred}} \mid \boldsymbol{\beta}_{\mathsf{pred}}, \boldsymbol{\theta}) f(\boldsymbol{\beta}_{\mathsf{pred}} \mid \boldsymbol{\theta}) f(\mathbf{D}^* \mid \mathbf{D}) f(\boldsymbol{\theta} \mid \mathbf{y}) \partial \boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int\limits_{\boldsymbol{\Theta}_{\mathsf{aug}}} \Bigg\{ \Bigg[\prod_{t=1}^{T^\star} f(\mathbf{y}^*_{T+t} \mid \mathbf{y}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}) f(\mathbf{y}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\
&\qquad\qquad f(\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}) \Bigg] f(\mathbf{D}^* \mid \mathbf{D}) f(\boldsymbol{\theta} \mid \mathbf{y}) \Bigg\} \partial \boldsymbol{\theta}_{\mathsf{aug}}. \qquad (2.18)
\end{aligned}
$$

Since the integral that appears in Equation (2.18) is not analytically tractable, we may use Monte Carlo methods to compute it by adding extra steps to Algorithm 6. The procedure is described in Algorithm 9 and implemented in Appendix B.1.6.2.

2.4 Model comparison

In this work, we are interested in analyzing spatiotemporal data by comparing the predictive performance of the anisotropic model ($\mathcal{M}_{\mathrm{A}}$) given in Equation (2.3) with that of its analogous isotropic model ($\mathcal{M}_{\mathrm{I}}$), which is described as follows:

$$
\begin{aligned}
\mathbf{Y}_t \mid \boldsymbol{\beta}_t, V, \phi, \boldsymbol{\Sigma} &\sim \mathsf{N}_{N \times q}(\mathbf{X}_t \boldsymbol{\beta}_t, V \cdot \mathbf{B}, \boldsymbol{\Sigma}),\ t \in \{1, \ldots, T\}, \\
\mathbf{B} &= \left[\ B_{n,n'}\ \right]_{N \times N}, \\
B_{n,n'} &= \begin{cases} \exp\{-\phi \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases}, \\
\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{p \times q}(\mathbf{G}_t \boldsymbol{\beta}_{t-1}, V \cdot \mathbf{W}, \boldsymbol{\Sigma}),\ t \in \{1, \ldots, T\}, \\
\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{p \times q}(\mathbf{M}_0, V \cdot \mathbf{C}_0, \boldsymbol{\Sigma}), \\
V &\sim \mathsf{IG}(a_V, b_V), \\
\boldsymbol{\Sigma} &\sim \mathsf{IW}_q(a_{\boldsymbol{\Sigma}}, \mathbf{b}_{\boldsymbol{\Sigma}}), \\
\phi &\sim \mathsf{G}(a_\phi, b_\phi).
\end{aligned} \qquad (2.19)
$$

Algorithm 9 Monte Carlo integration to approximate $f(\mathbf{y}_{\mathsf{aug}} \mid \mathbf{y})$.

Start with K MCMC samples and also with the hyperparameters $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$ and ψ, where the k^{th} MCMC sample is given by $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$.

1: **for** $k \leftarrow 1$ to K **do**

2: Sample $\mathbf{D}^*_{(j_k)}$ from the distribution $\mathsf{N}_{2\times N^*}(\mathbf{S}^* + (\mathbf{D}^{(j_k)} - \mathbf{S})\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^* - \mathbf{R}_{\mathsf{u},\mathsf{g}}\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}})$.

3: Compute the matrices $\mathbf{B}_{(j_k)} = \left[\ B_{n,n'}^{(j_k)}\ \right]_{N\times N}$, $\mathbf{B}_{\mathsf{g},\mathsf{u}}^{(j_k)} = \left[\ B_{n,n'}^{\mathsf{g},\mathsf{u}^{(j_k)}}\ \right]_{N\times N^*}$ ($\mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}$ is its transpose) and $\mathbf{B}^*_{(j_k)} = \left[\ B_{n,n'}^{*(j_k)}\ \right]_{N^*\times N^*}$, where:

$$
\begin{aligned}
B_{n,n'}^{(j_k)} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{n'})\|\}, && n, n' \in \{1,\ldots,N\},\\
B_{n,n'}^{\mathsf{g},\mathsf{u}^{(j_k)}} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}, && n \in \{1,\ldots,N\}, n' \in \{1,\ldots,N^*\},\\
B_{n,n'}^{*(j_k)} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n}) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}, && n, n' \in \{1,\ldots,N^*\}.
\end{aligned}
$$

4: **for** $t \leftarrow 1$ to $T^\star$ **do**

5: **if** $t \leftarrow 1$ **or** $T^\star \leftarrow 1$ **then**

6: Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{p\times q}(\mathbf{G}_{T+1}\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \mathbf{W}, \boldsymbol{\Sigma}^{(j_k)})$.

7: **else**

8: Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the following distribution:

$$
\mathsf{N}_{p\times q}\left(\left[\prod_{t'=t}^{1}\mathbf{G}_{T+t'}\right]\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \left\{\mathbf{W} + \sum_{t'=t}^{2}\left[\prod_{t''=t}^{t'}\mathbf{G}_{T+t''}\right]^{\mathsf{T}}\mathbf{W}\left[\prod_{t''=t}^{t'}\mathbf{G}_{T+t''}\right]\right\}, \boldsymbol{\Sigma}^{(j_k)}\right).
$$

9: **end if**

10: Sample $\mathbf{y}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{N\times q}(\mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t}^{(j_k)}, V^{(j_k)} \cdot \mathbf{B}_{(j_k)}, \boldsymbol{\Sigma}^{(j_k)})$.

11: Sample $\mathbf{y}_{T+t}^{*(j_k)}$ from the following distribution:

$$
\mathsf{N}_{N^*\times q}(\mathbf{X}^*_{T+t}\boldsymbol{\beta}_{T+t}^{(j_k)} + \mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}\mathbf{B}_{(j_k)}^{-1}(\mathbf{y}_{T+t}^{(j_k)} - \mathbf{X}_{T+t}\boldsymbol{\beta}_{T+t}^{(j_k)}), V^{(j_k)} \cdot (\mathbf{B}^*_{(j_k)} - \mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}\mathbf{B}_{(j_k)}^{-1}\mathbf{B}_{\mathsf{g},\mathsf{u}}^{(j_k)}), \boldsymbol{\Sigma}^{(j_k)}).
$$

12: **end for**

13: **end for**

In this section, we will also review the following measures to select models: Deviance Information Criterion (DIC) and Predictive Mean Squared Error (PMSE). Also, we will review the Empirical Coverage Probability (ECP) and the Interval Score (IS). These four quantities will be used to compare the anisotropic model ($\mathcal{M}_{\text{A}}$) with its analogous isotropic model ($\mathcal{M}_{\text{I}}$).

The *Deviance Information Criterion* (DIC) is a metric for comparing Bayesian models (Spiegelhalter et al., 2002; Gelman et al., 2013). To present it, it is first necessary to introduce some notation. Let $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$ be the $j_k{}^{\text{th}}$ MCMC sample of $\boldsymbol{\theta}$, where $k \in \{1, \ldots, K\}$, K is size of the MCMC chains and $\{j_1, \ldots, j_K\}$ is a set of indices. Define $\bar{\boldsymbol{\theta}} = \{\bar{\boldsymbol{\beta}}_0, \bar{\boldsymbol{\beta}}_1, \ldots, \bar{\boldsymbol{\beta}}_T, \bar{V}, \bar{\phi}, \bar{\mathbf{D}}, \bar{\boldsymbol{\Sigma}}\}$, where $\bar{\boldsymbol{\beta}}_t = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\beta}_t^{(j_k)}$ for $t \in \{0, 1, \ldots, T\}$, $\bar{V} = \frac{1}{K}\sum_{k=1}^{K} V^{(j_k)}$, $\bar{\phi} = \frac{1}{K}\sum_{k=1}^{K} \phi^{(j_k)}$, $\bar{\mathbf{D}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{D}^{(j_k)}$ and $\bar{\boldsymbol{\Sigma}} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{\Sigma}^{(j_k)}$. According to Paulino et al. (2018, Eq. 8.37) and also from the likelihood function given in Equation (2.5), we have the following expression for the DIC statistic:

$$\begin{aligned}
\text{DIC} &= 2\,\text{E}_{\boldsymbol{\theta}|\mathbf{Y}=\mathbf{y}}[-2\ln f(\mathbf{y} \mid \boldsymbol{\theta})] - \{-2\ln f(\mathbf{y} \mid \text{E}[\boldsymbol{\theta} \mid \mathbf{Y} = \mathbf{y}])\} \qquad (2.20)\\
&\approx -4 \cdot \left\{\frac{1}{K}\sum_{k=1}^{K} \ln f(\mathbf{y} \mid \boldsymbol{\theta}^{(j_k)})\right\} + 2\ln f(\mathbf{y} \mid \bar{\boldsymbol{\theta}})\\
&= -\frac{4}{K}\sum_{k=1}^{K}\sum_{t=1}^{T} \ln f(\mathbf{y}_t \mid \boldsymbol{\beta}_t^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}) + 2\sum_{t=1}^{T} \ln f(\mathbf{y}_t \mid \bar{\boldsymbol{\beta}}_t, \bar{V}, \bar{\phi}, \bar{\mathbf{D}}, \bar{\boldsymbol{\Sigma}}).
\end{aligned}$$

The *Predictive Mean Squared Error* (PMSE) measures the mean squared difference between the predicted value and the observed value and is averaged over all N^* ungauged locations (Shen and Gelfand, 2019; Morales et al., 2022). In our work, this metric is given by

$$\text{PMSE} = \frac{1}{N^* q T}\sum_{n=1}^{N^*}\sum_{i=1}^{q}\sum_{t=1}^{T}(\bar{Y}_{N+n,i,t} - Y_{N+n,i,t})^2, \qquad (2.21)$$

where, in the Bayesian context, $\bar{Y}_{N+n,i,t}$ is the predictive posterior mean (i.e. use the samples $\mathbf{Y}_t^{*(j_1)} = \begin{bmatrix} Y_{N+n,i,t}^{(j_1)} \end{bmatrix}_{N^*\times q}, \ldots, \mathbf{Y}_t^{*(j_K)} = \begin{bmatrix} Y_{N+n,i,t}^{(j_K)} \end{bmatrix}_{N^*\times q}$ obtained in Algorithm 8 to compute $\bar{Y}_{N+n,i,t} = \frac{1}{K}\sum_{k=1}^{K} Y_{N+n,i,t}^{(j_k)}$).

The *Empirical Coverage Probability* (ECP) is the proportion of times that the predictive interval contains the observed value (Banerjee et al., 2014, Chapter 5). Here we consider the range from the 2.5^{th} to the 97.5^{th} percentile, say $\hat{Y}_{N+n,i,t}^{[\alpha/2]}$ and $\hat{Y}_{N+n,i,t}^{[1-\alpha/2]}$ for $\alpha = 0.05$. For some ungauged site $n \in \{1, \ldots, N^*\}$ and response variable $i \in \{1, \ldots, q\}$, we have the following expression:

$$\text{ECP}_{n,i}^{[\alpha]} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{1}_{\left[\hat{Y}_{N+n,i,t}^{[\alpha/2]}, \hat{Y}_{N+n,i,t}^{[1-\alpha/2]}\right]}(Y_{N+n,i,t}). \qquad (2.22)$$

Winkler (1972) proposed a measure to calculate the accuracy of the predictive interval for any given model, known as *Interval Score* (IS). The IS is defined as the length of the interval plus a penalty if the observation is outside the interval. In this work, the IS for each period is given by

$$\mathrm{IS}^{[\alpha]}_{N+n,i,t} = \begin{cases} \left(\hat{Y}^{[1-\alpha/2]}_{N+n,i,t} - \hat{Y}^{[\alpha/2]}_{N+n,i,t}\right), & Y_{N+n,i,t} \in \left[\hat{Y}^{[\alpha/2]}_{N+n,i,t}, \hat{Y}^{[1-\alpha/2]}_{N+n,i,t}\right] \\ \left(\hat{Y}^{[1-\alpha/2]}_{N+n,i,t} - \hat{Y}^{[\alpha/2]}_{N+n,i,t}\right) + \dfrac{2 \cdot \left\{\hat{Y}^{[\alpha/2]}_{N+n,i,t} - Y_{N+n,i,t}\right\}}{\alpha}, & Y_{N+n,i,t} < \hat{Y}^{[\alpha/2]}_{N+n,i,t} \\ \left(\hat{Y}^{[1-\alpha/2]}_{N+n,i,t} - \hat{Y}^{[\alpha/2]}_{N+n,i,t}\right) + \dfrac{2 \cdot \left\{Y_{N+n,i,t} - \hat{Y}^{[1-\alpha/2]}_{N+n,i,t}\right\}}{\alpha}, & Y_{N+n,i,t} > \hat{Y}^{[1-\alpha/2]}_{N+n,i,t} \end{cases}$$

and the average IS is given by

$$\mathrm{IS}^{[\alpha]}_{N+n,i} = \frac{1}{T}\sum_{t=1}^{T} \mathrm{IS}^{[\alpha]}_{N+n,i,t}\,. \tag{2.23}$$

The model with the lowest DIC is estimated to be the model that would best predict a replicate dataset which has the same structure as that currently observed one (Tsiko, 2015, Page 5). Models with smaller PMSE are preferred for interpolation (Shen and Gelfand, 2019, Sec. 4). A model that has a narrow predictive interval with high ECP is preferred; therefore, a model with a smaller IS is better (Rostami-Tabar and Rendon-Sanchez, 2021, Sec. 3.4.2).

2.5 Simulation studies

2.5.1 First simulation study – Retrieving model parameters

In this simulation study, we want to evaluate whether Algorithm 6 is able to correctly retrieve the parameters. We consider $q = 2$ response variables, three sample sizes given by $T \in \{10, 100, 1000\}$, and $p = 2$ regression coefficients per time and response variable. We simulate the process Y at $N = 17$ points in the unit square (i.e. $\mathcal{S} = [0,1]^2$). Figure 2.1 shows the geographic region of interest and highlights which points are used as anchor points ($\underset{\sim}{s}_1$ and $\underset{\sim}{s}_2$) and the other non-anchor points ($\underset{\sim}{s}_3, \ldots, \underset{\sim}{s}_{17}$). These seventeen points are used to fit the model. We generate data from the following anisotropic scheme:

1. Generate $\mathbf{D} = \begin{bmatrix} \mathbf{D}_{1:2} & \mathbf{D}_{3:17} \end{bmatrix}_{2\times 17}$ proceeding as follows:
   (a) Do $\mathbf{D}_{1:2} = \mathbf{S}_{1:2}$ (i.e. $d(\underset{\sim}{s}_1) = \underset{\sim}{s}_1$ and $d(\underset{\sim}{s}_2) = \underset{\sim}{s}_2$) to fix the locations of two sites in $\mathcal{D}$-space, where $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{1:2} & \mathbf{S}_{3:17} \end{bmatrix}_{2\times 17}$.
   (b) Apply Equation (2.14) to obtain $\mathbf{D}_{3:17} \mid \mathbf{D}_{1:2} = \mathbf{S}_{1:2}$ and generate $\mathbf{D}_{3:17}$ from the distribution $\mathsf{N}_{2\times 15}(\mathbf{S}_{3:17}, \mathrm{diag}\{0.500, 0.500\}, \mathbf{R}_{3:17} - \mathbf{R}_{3:17,1:2}\mathbf{R}^{-1}_{1:2}\mathbf{R}_{1:2,3:17})$, where $\mathbf{R}_{1:2}$ (2×2),

$\mathbf{R}_{1:2,3:17}$ (2×15), $\mathbf{R}_{3:17,1:2}$ (15×2) and $\mathbf{R}_{3:17}$ (15×15) are blocks of the following matrix:

$$\mathbf{R}_d = \left[\ \exp\{-\tilde{\psi}\cdot\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2\}\ \right]_{17\times 17} = \begin{bmatrix} \mathbf{R}_{1:2} & \mathbf{R}_{1:2,3:17} \\ \mathbf{R}_{3:17,1:2} & \mathbf{R}_{3:17} \end{bmatrix}.$$

As the greatest square of the Euclidean distance between two points in Figure 2.1 is $\|\underset{\sim}{\mathbf{s}}_1-\underset{\sim}{\mathbf{s}}_2\|^2 = \|\underset{\sim}{\mathbf{s}}_5-\underset{\sim}{\mathbf{s}}_{15}\|^2 = 1^2+1^2 = 2$, here $\tilde{\psi} = -2\ln(0.05)/\left(\max\limits_{n,n'\in\{1,\ldots,17\}}\|\underset{\sim}{\mathbf{s}}_n-\underset{\sim}{\mathbf{s}}_{n'}\|^2\right) \simeq 3$.

2. Set $V = 0.4$, $\phi = 0.1$ and $\mathbf{\Sigma} = \left[\begin{smallmatrix} 1.0 & 0.7 \\ 0.7 & 1.0 \end{smallmatrix}\right]$ as true parameters, generate $\boldsymbol{\beta}_0$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{0}_{2\times 2}, V\cdot\mathbf{I}_2, \mathbf{\Sigma})$ and compute $\mathbf{B} = \left[\ B_{n,n'}\ \right]_{17\times 17}$, where, for all $n, n' \in \{1, 2, \ldots, 17\}$, we have:

$$B_{n,n'} = \begin{cases} \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'})\|\}, & n \neq n' \\ 1, & n = n' \end{cases}.$$

3. Using one of the three sample sizes (i.e. $T \in \{10, 100, 1000\}$), for $t = 1, \ldots, T$ do the following:

   (a) Generate $\boldsymbol{\beta}_t$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{G}_t\boldsymbol{\beta}_{t-1}, V\cdot\mathbf{W}, \mathbf{\Sigma})$, where $\mathbf{G}_t = \mathbf{I}_2$ and $\mathbf{W} = \mathbf{I}_2$.

   (b) Generate $\mathbf{X}_t = [\,{}^{1}_{U_{1,t}}\ {}^{1}_{U_{2,t}}\ \cdots\ {}^{1}_{U_{17,t}}\,]^{\mathsf{T}}$ (17×2), where $U_{1,t}, \ldots, U_{17,t}$ are seventeen random numbers between zero and one.

   (c) Generate $\mathbf{Y}_t$ from the distribution $\mathsf{N}_{17\times 2}(\mathbf{X}_t\boldsymbol{\beta}_t, V\cdot\mathbf{B}, \mathbf{\Sigma})$.

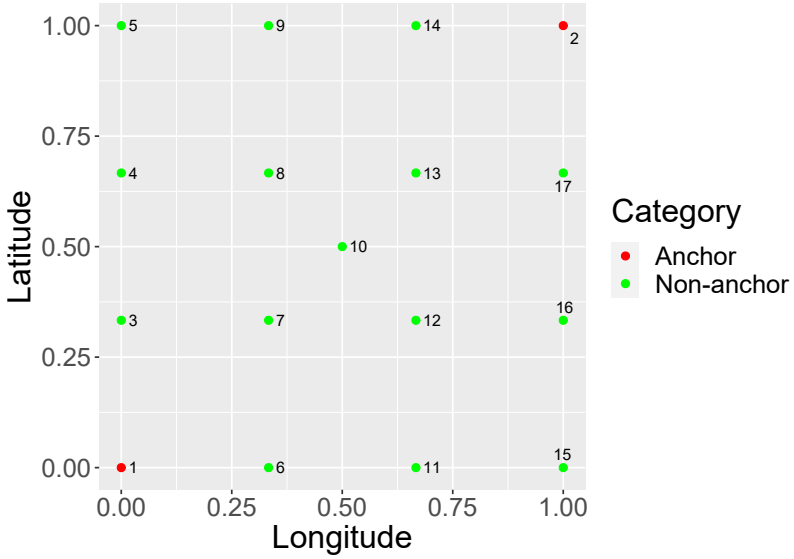


Figure 2.1: Geographic region of interest ($\mathcal{S}$) of the first simulation study in Chapter 2.

To run Algorithm 6, we use the fixed matrices $\mathbf{W}$, $\mathbf{G}_1, \ldots, \mathbf{G}_T$, $\mathbf{X}_1, \ldots, \mathbf{X}_T$ and $\mathbf{Y}_1, \ldots, \mathbf{Y}_T$ generated on 3(a)-(c) for a given $T \in \{10, 100, 1000\}$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\mathbf{\Sigma}} = 0.001$, $\mathbf{b}_{\mathbf{\Sigma}} = 0.001\cdot\mathbf{I}_2$, $a_\phi = 0.001$, $b_\phi = 0.001$,

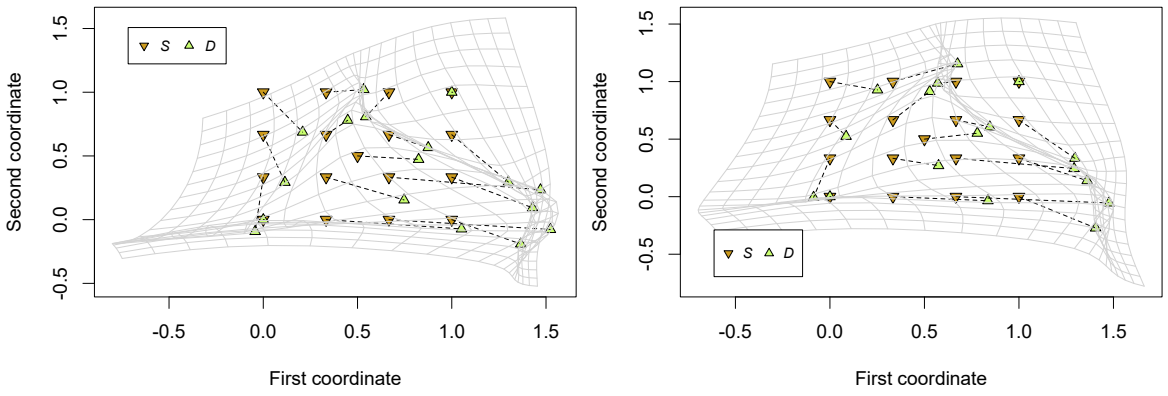
$\mathbf{M}_0 = \mathbf{0}_{2\times 2}$, $\mathbf{C}_0 = \mathbf{I}_2$, and $\boldsymbol{\sigma}_d^2$ as the empirical covariance matrix of the gauged sites. We choose $\psi = 5.0$ after a few tries. In Appendix B.1.3, we present the code written in Python for the data generation scheme related to the first simulation study and specifying these hyperparameters.

Algorithm 6 was run 11000 times for all three sample sizes, having presented convergence after $J \approx 1000$ iterations. We applied the slice sampler in its 6^{th} step. To avoid autocorrelation in the chains, we form a sample of size $K = 1000$ of the posterior distribution of the parameters by systematically sampling every ten iterations (i.e. $j_1 = 1001, j_2 = 1011, \ldots, j_{1000} = 10991$).

The true deformation and the estimated deformations for the three sample sizes are shown in Figure 2.2, where we visually notice that the true deformation is not well estimated if the sample size is small ($T = 10$). Figures 2.3(a), 2.3(b), 2.3(c) and 2.3(d) show that the posterior distributions of the parameters ϕ, $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$ and $V \cdot \Sigma_{2,2}$ do not recover the corresponding true values, which is exemplified only to the case $T = 100$. This happens because ϕ, V and $\boldsymbol{\Sigma} = \left[\Sigma_{i,i'} \right]_{2\times 2}$ are not identifiable, as discussed in Section 2.2.3. Figure 2.4 shows that the posterior distribution of $\phi \cdot V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ recover the true values even when the sample size is small. This finding is important because it was not necessary to elicit an informative prior distribution for ϕ, such as $\phi \sim \mathsf{G}(a_\phi b_\phi, b_\phi)$ for a given $\mathrm{E}[\phi] = a_\phi$ and a "large" $b_\phi > 0$ (Schmidt and Gelfand, 2003), commonly specified in some works in this area. However, we point out that prior distributions for ϕ based on Euclidean distances may be necessary to calibrate this parameter in more realistic geographic regions than the unit square[5]. Figures 2.5, 2.6, 2.7 and 2.8 show that the posterior distributions of $\beta_{0,1,t}, \beta_{0,2,t}, \beta_{1,1,t}, \beta_{1,2,t}$ $(t = 0, 1, \ldots, T)$ estimate well their true values for $T \in \{10, 100, 1000\}$, respectively.

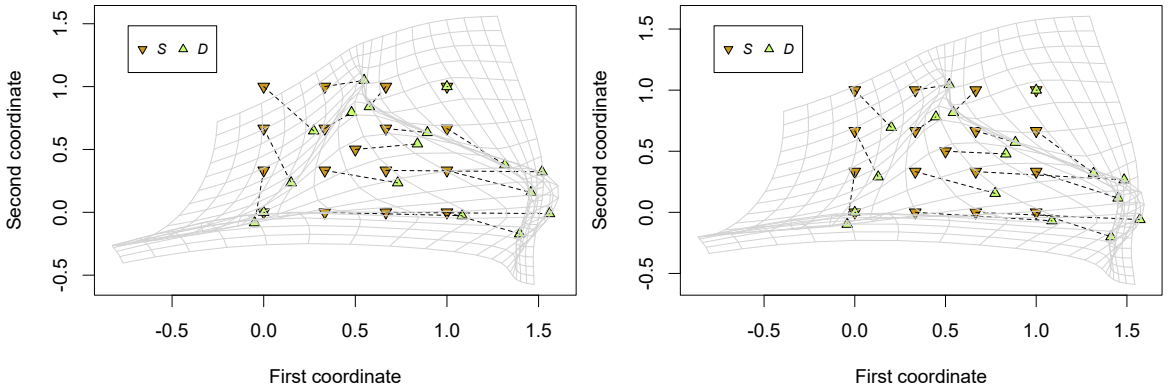
Due to the use of the slice sampler, the acceptance rate of each component $D_{m,n}$, $3 \leq n \leq 17$, is equal to 100%. The acceptance rates of the parameter ϕ are equal to 45.55%, 44.07% and 46.86%, respectively for the sample sizes $T \in \{10, 100, 1000\}$. The trace plots for the posterior distributions of $\phi \cdot V \cdot \Sigma_{i,i'}$ and $\mathbf{D}$ from this simulation study are available in Appendix C.1.1.

[5]See, for example, Fonseca and Steel (2011, Sec. 4) and Morales et al. (2022, Sec. 2.1).



(a) True deformation.

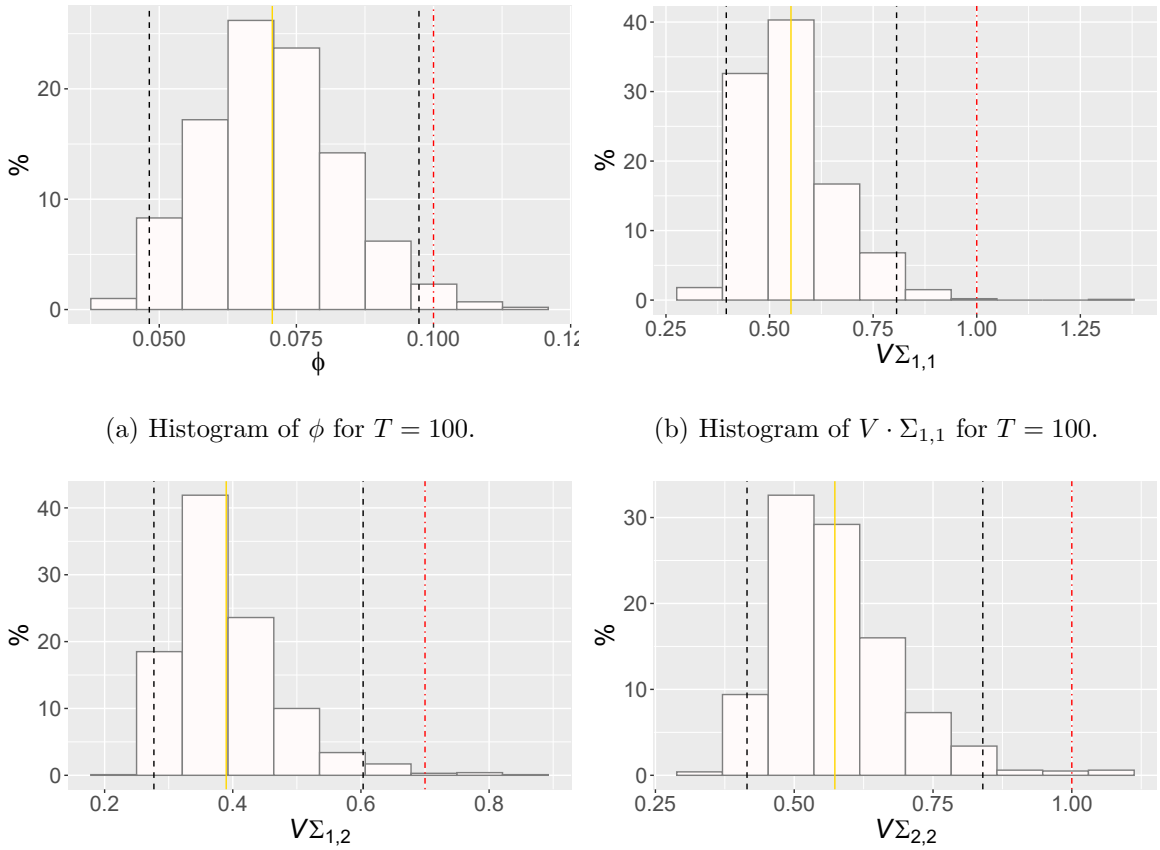
(b) Estimated deformation with $T = 10$.



(c) Estimated deformation with $T = 100$.

(d) Estimated deformation with $T = 1000$.

Figure 2.2: True deformation and posterior means of $\mathbf{D}$ for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2.



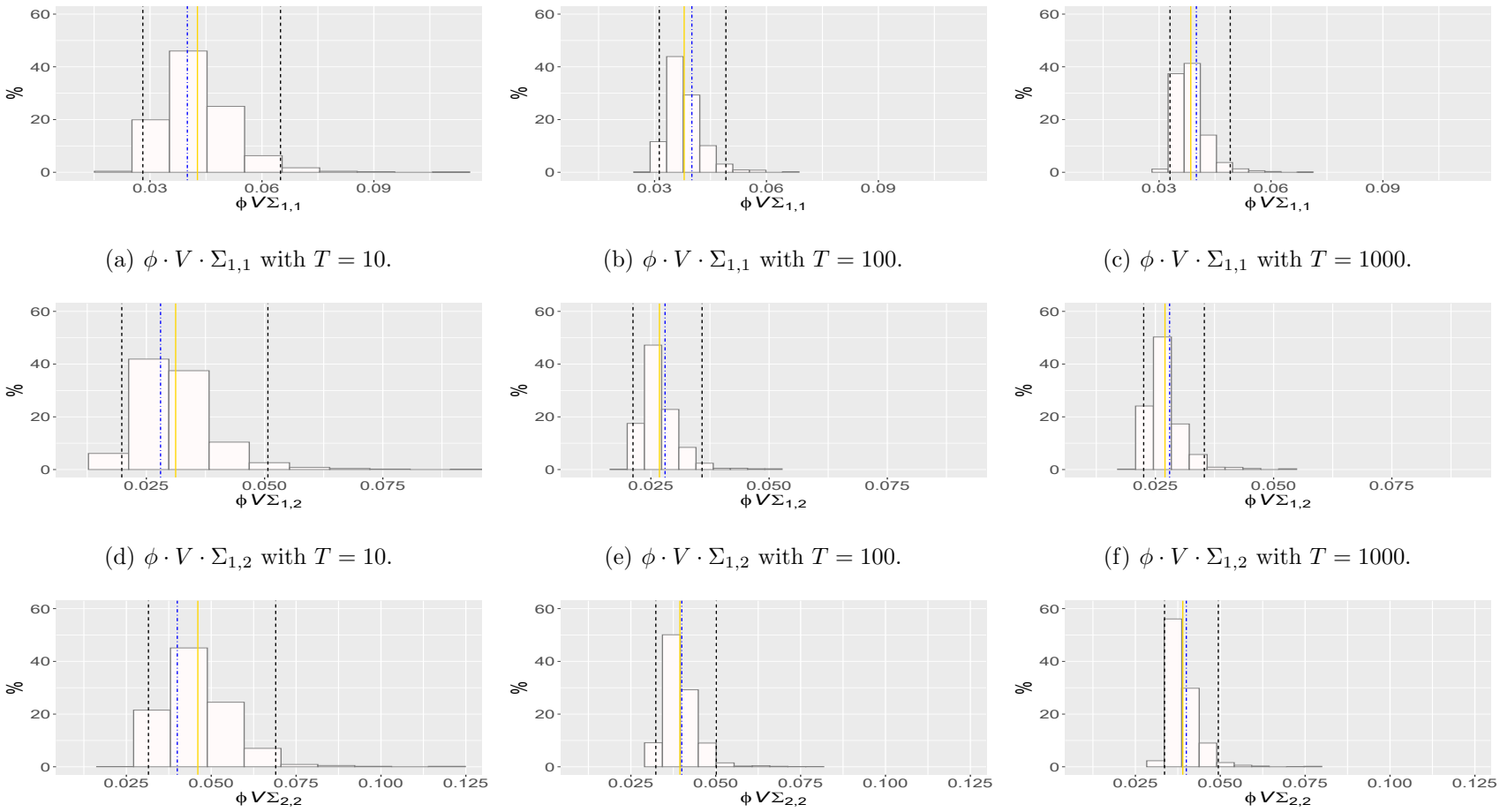
(a) Histogram of ϕ for $T = 100$.

(b) Histogram of $V \cdot \Sigma_{1,1}$ for $T = 100$.

(c) Histogram of $V \cdot \Sigma_{1,2}$ for $T = 100$.

(d) Histogram of $V \cdot \Sigma_{2,2}$ for $T = 100$.

Figure 2.3: Histograms of the posterior distribution of ϕ, $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$ and $V \cdot \Sigma_{2,2}$ for $T = 100$, resulting from the first simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).



(a) $\phi \cdot V \cdot \Sigma_{1,1}$ with $T = 10$.

(b) $\phi \cdot V \cdot \Sigma_{1,1}$ with $T = 100$.

(c) $\phi \cdot V \cdot \Sigma_{1,1}$ with $T = 1000$.

(d) $\phi \cdot V \cdot \Sigma_{1,2}$ with $T = 10$.

(e) $\phi \cdot V \cdot \Sigma_{1,2}$ with $T = 100$.

(f) $\phi \cdot V \cdot \Sigma_{1,2}$ with $T = 1000$.

(g) $\phi \cdot V \cdot \Sigma_{2,2}$ with $T = 10$.

(h) $\phi \cdot V \cdot \Sigma_{2,2}$ with $T = 100$.

(i) $\phi \cdot V \cdot \Sigma_{2,2}$ with $T = 1000$.

Figure 2.4: Histograms of the posterior distribution of $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

After several attempts, we found that the collection of parameters $\boldsymbol{\theta} = \{V, \boldsymbol{\Sigma}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, \phi, \mathbf{D}\}$ of the model described by Equation (2.3), where $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T\}$, is only well estimated with the specification of non-informative prior distributions when the full conditional distributions of all components – Equations (2.7), (2.8), (2.9), (2.10) and (2.11) – directly involve the likelihood function given in Equation (2.5). For this reason, we treat $\boldsymbol{\sigma}_d^2$ as a hyperparameter and $\mathbf{W}$ as fixed in this work.

We also run Algorithm 6 applying the Metropolis-Hastings algorithm instead of slice sampler to sample $\mathbf{D}$ in its 6^{th} step. We consider the case $T = 100$, under the same specifications, for comparative purposes. The acceptance rate of each component of the parameter $\mathbf{D}$ is given by $\begin{bmatrix} 0.00 & 0.00 & 33.08 & 48.27 & 47.13 & 52.68 & 47.02 & 26.53 & 48.51 & 31.97 & 42.75 & 38.94 & 31.35 & 25.61 & 44.20 & 35.91 & 41.40 \\ 0.00 & 0.00 & 31.08 & 52.75 & 49.54 & 55.75 & 48.71 & 35.5 & 41.66 & 30.95 & 38.50 & 38.01 & 29.20 & 29.15 & 47.58 & 32.20 & 44.18 \end{bmatrix}\%$. Figure 2.9 shows the estimated deformation and the trace plots of the chains of the parameters $D_{1,3}$ and $D_{2,3}$ obtained with both the slice sampler and the Metropolis-Hastings algorithms, noticing that the estimated deformations are very similar. Table 2.1 shows the autocorrelation for the chains of $D_{1,3}$ and $D_{2,3}$ at lags 0, 1, 5, 10 and 50, by algorithm. The corresponding autocorrelation plot is shown in Figure C.8. The use of the slice sampler algorithm generates chains for $D_{1,3}$ and $D_{2,3}$ with less autocorrelation and with a processing time of 23.6 minutes, while the use of the Metropolis-Hastings algorithm generates chains for $D_{1,3}$ and $D_{2,3}$ with a more noticeable autocorrelation and with processing time of 5.4 minutes[6]. Although the slice sampling method is slower, it allows the use of much smaller chains due to much smaller autocorrelation.

Table 2.1: Autocorrelation for the chains of the parameters $D_{1,3}$ and $D_{2,3}$ at lags 0, 1, 5, 10 and 50, obtained through the slice sampling (SS) and Metropolis-Hastings (MH) algorithms with $T = 100$, resulting from the first simulation study in Chapter 2.

Number	$D_{1,3}$		$D_{2,3}$	
of lags	**SS**	**MH**	**SS**	**MH**
0	1.00000000	1.000000000	1.00000000	1.000000000
1	-0.01678581	0.395769527	0.04442386	0.316242578
5	0.01983039	0.069915080	-0.01470370	0.010094051
10	0.02399235	0.035563727	0.03934883	0.012484223
50	0.01552206	0.003160147	-0.01666782	-0.000651269

[6]The algorithms were run on a computer with the following configuration: (i) Processor: Intel® Core™ i7-6500U CPU @ 2.50 GHz; (ii) Installed RAM: 16 GB.

Thus, through this simulation study we conclude that Algorithm 6 is suitable to correctly retrieve the parameters of the model given in Equation (2.3) for $T \geq 100$.

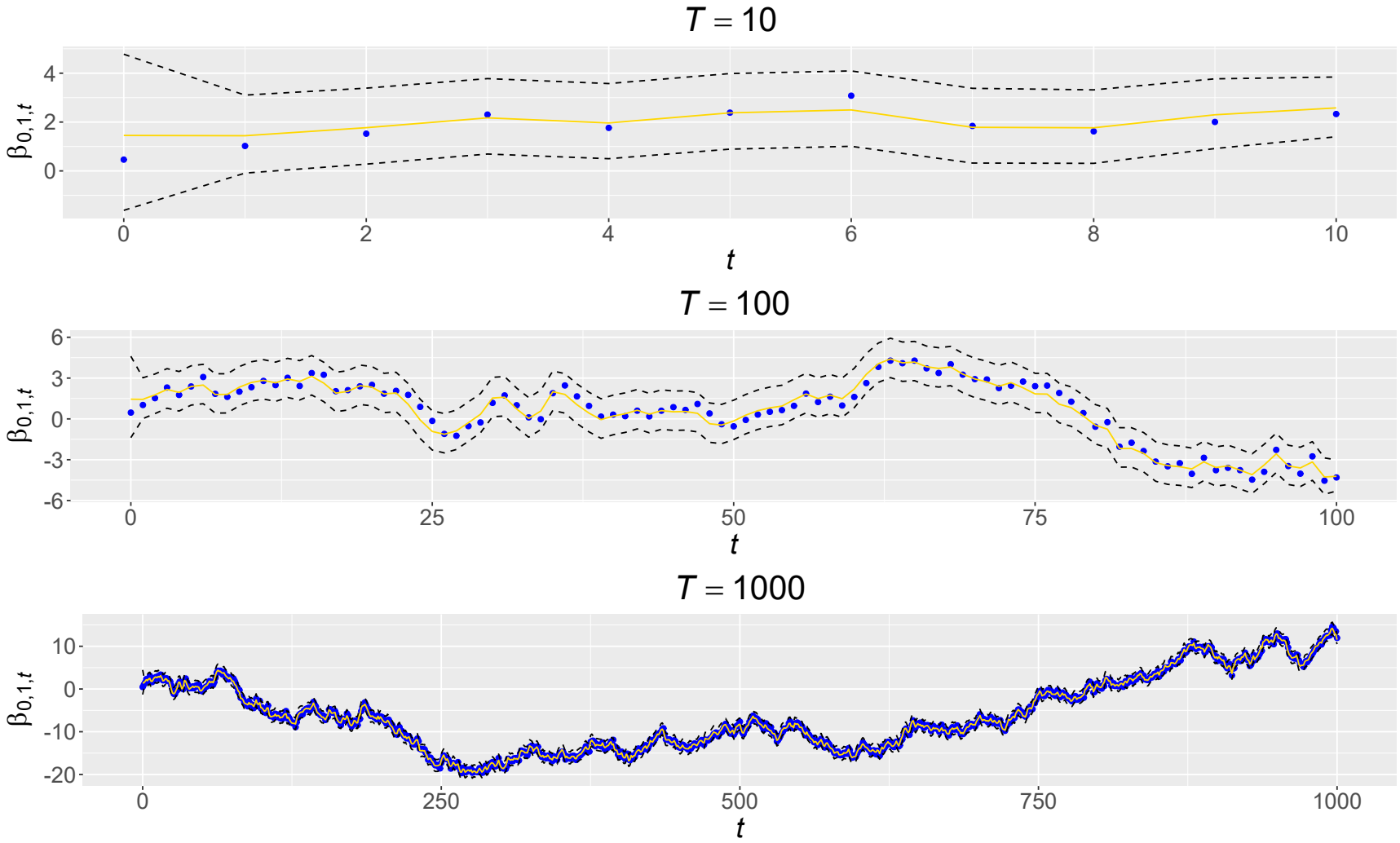


Figure 2.5: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, T\}$ and $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

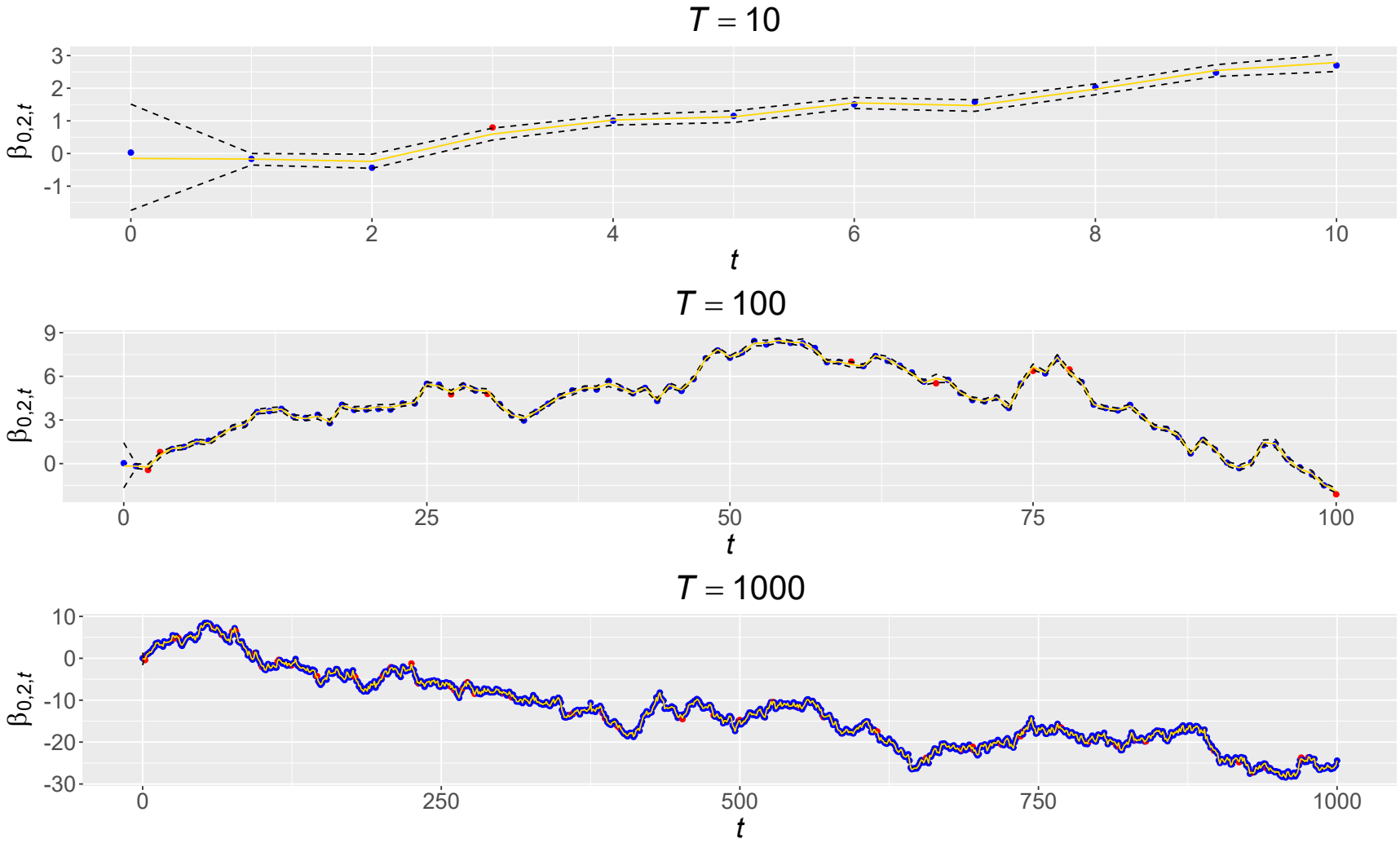


Figure 2.6: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, T\}$ and $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

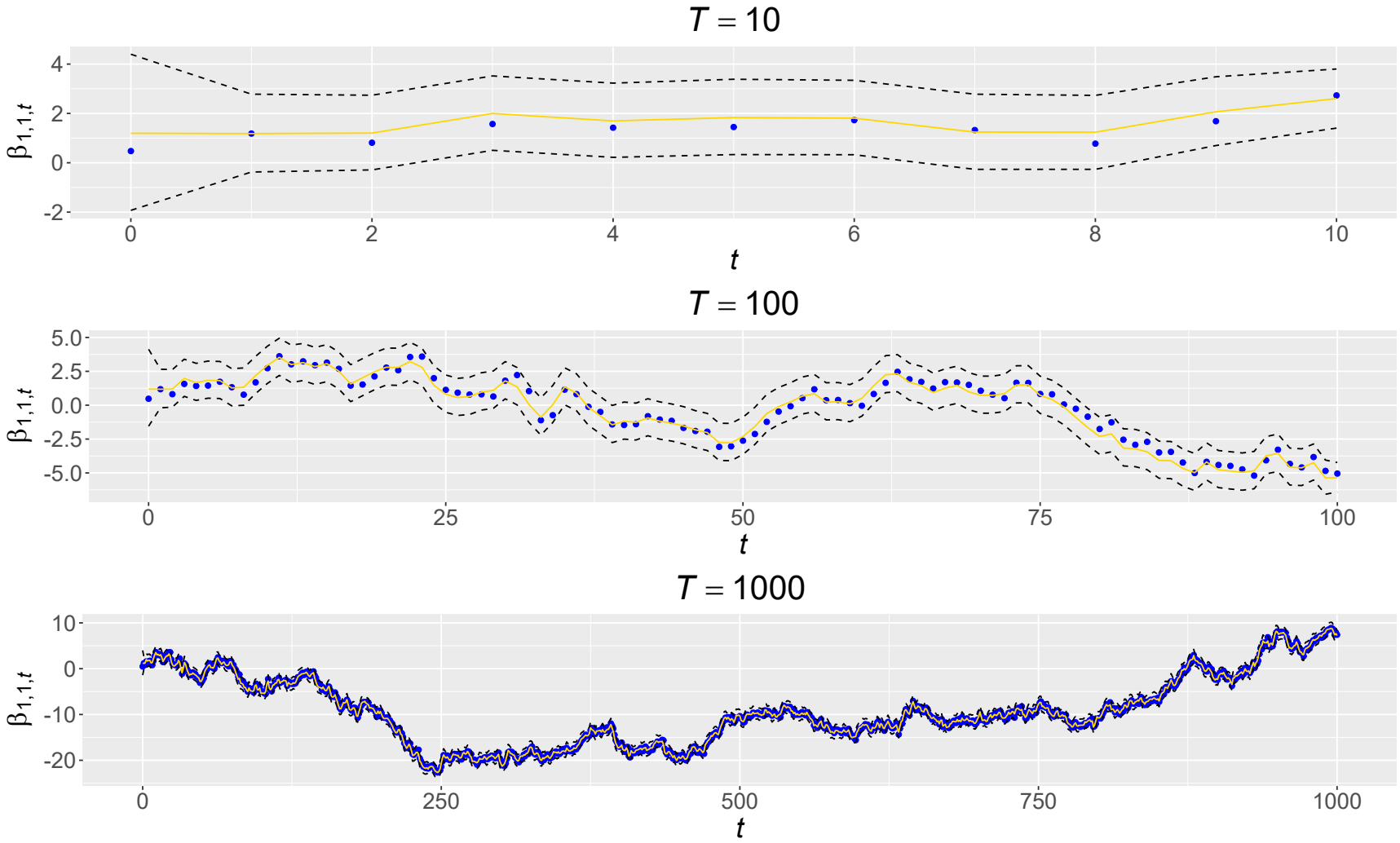


Figure 2.7: Line chart of the posterior distribution of $\beta_{1,1,t}$ for $t \in \{0, 1, \ldots, T\}$ and $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

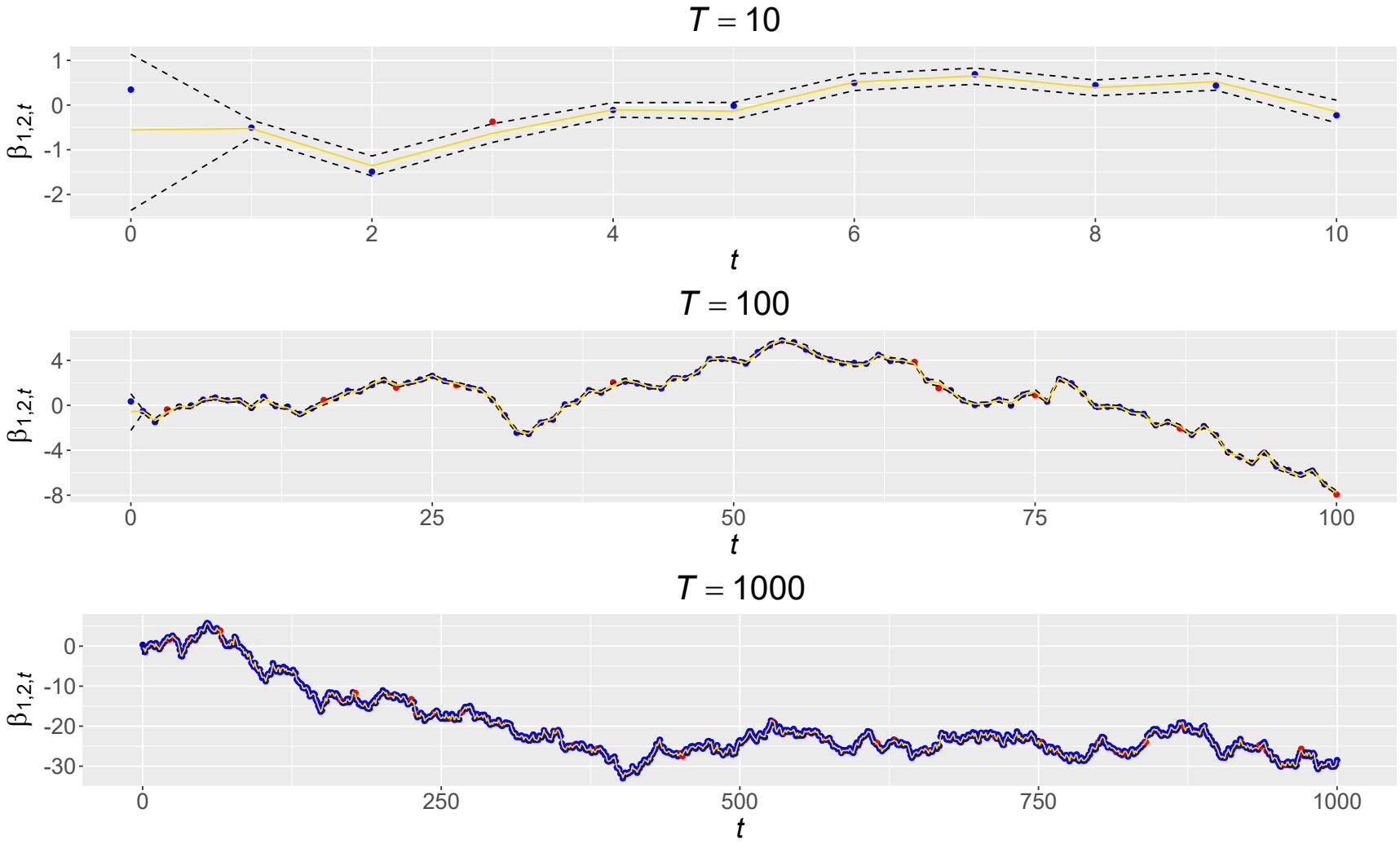
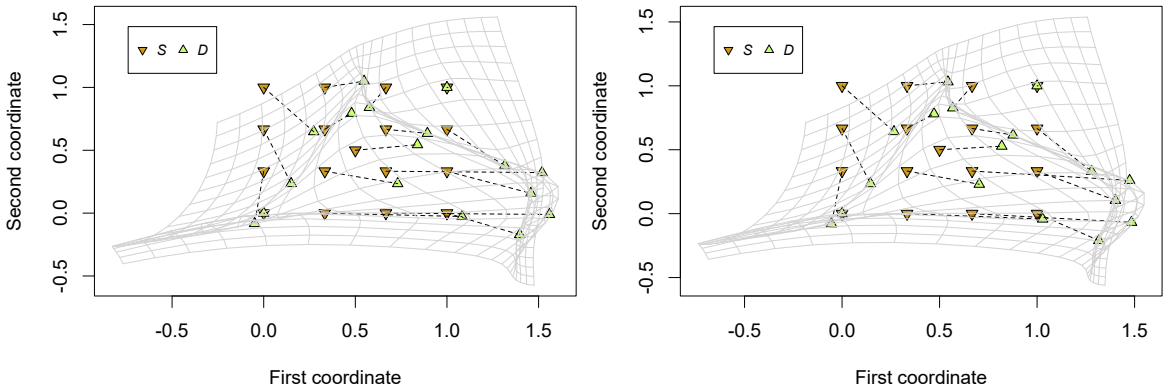
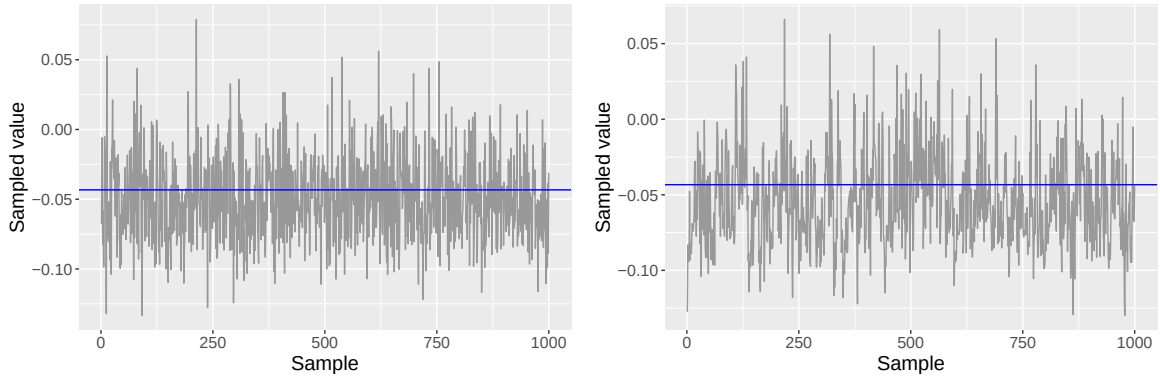


Figure 2.8: Line chart of the posterior distribution of $\beta_{1,2,t}$ for $t \in \{0, 1, \ldots, T\}$ and $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).



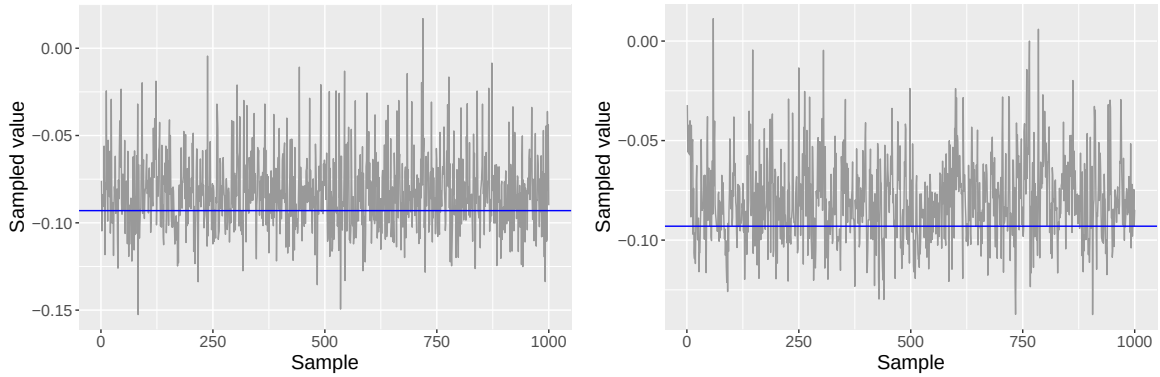
(a) Estimated deform. w/ the slice sampler.

(b) Estimated deform. w/ the MH algorithm.



(c) $D_{1,3}$ with the slice sampler.

(d) $D_{1,3}$ with the MH algorithm.



(e) $D_{2,3}$ with the slice sampler.

(f) $D_{2,3}$ with the MH algorithm.

Figure 2.9: Estimated deformation and trace plot of the deformation parameter $\underset{\sim}{\mathbf{d}}_3 = (D_{1,3}, D_{2,3})$ for $T = 100$, by algorithm (the slice sampler and the Metropolis-Hastings or MH), resulting from the first simulation study in Chapter 2. The corresponding true value is represented by the blue solid line and the true deformation is shown in Figure 3.2(a)

2.5.2 Second simulation study – To use or not to use spatial deformation to model anisotropic data?

In this second simulation study, we will generate anisotropic data to study the predictive performance (forecasting and interpolation) of the anisotropic model ($\mathcal{M}_{\text{A}}$) given in Equation (2.3) and compare its results with those obtained through an analogous isotropic model ($\mathcal{M}_{\text{I}}$) given in Equation (2.19).

We prefer not to generate spatial deformations from the distribution $\mathsf{N}_{2\times N}(\mathbf{S}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d)$, as this would give a natural advantage to the anisotropic model ($\mathcal{M}_{\text{A}}$). To make a fairer comparison between the two models, we will generate data with geometric anisotropy. A geometrically anisotropic covariance function of a second-order stationary random field is obtained from a linear transformation of the separation vector (i.e. $\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'}$) of an isotropic covariance function (De Iaco et al., 2020, Sec. 2). Maity and Sherman (2012, Sec. 4) generate anisotropic data by imposing a spatial structure of the following exponential form:

$$\exp\{-\phi\sqrt{(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})^\intercal \mathbf{A}(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})}\}. \tag{2.24}$$

Based on Diggle and Ribeiro-Jr. (2007, Sec. 3.7) and Shen and Gelfand (2019, Sec. 2.2), we will work with a 2×2 matrix $\mathbf{A}$ that can be written in the factorized form $\mathbf{A} = \boldsymbol{\Lambda}^\intercal\boldsymbol{\Lambda}$ for some square matrix $\boldsymbol{\Lambda}$. Thus, we will use the structure given in Equation (2.24) to obtain deterministic deformations as follows:

$$\begin{aligned}
\exp\{-\phi\sqrt{(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})^\intercal \mathbf{A}(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})}\} &= \exp\{-\phi\sqrt{(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})^\intercal \boldsymbol{\Lambda}^\intercal\boldsymbol{\Lambda}(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})}\} \\
&= \exp\{-\phi\sqrt{[\boldsymbol{\Lambda}(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})]^\intercal[\boldsymbol{\Lambda}(\underline{\mathbf{s}}_n - \underline{\mathbf{s}}_{n'})]}\} \\
&= \exp\{-\phi\sqrt{[\boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_n - \boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_{n'}]^\intercal[\boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_n - \boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_{n'}]}\} \\
&= \exp\{-\phi\sqrt{[d(\underline{\mathbf{s}}_n) - d(\underline{\mathbf{s}}_{n'})]^\intercal[d(\underline{\mathbf{s}}_n) - d(\underline{\mathbf{s}}_{n'})]}\} \\
&= \exp\{-\phi\|d(\underline{\mathbf{s}}_n) - d(\underline{\mathbf{s}}_{n'})\|\},
\end{aligned}$$

where $d(\underline{\mathbf{s}}_n) = \boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_n$ and $d(\underline{\mathbf{s}}_{n'}) = \boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_{n'}$ are linear transformations[7] of the locations.

We based on the work of House and Keyser (2016, Appx. C.3) to specify the matrix $\boldsymbol{\Lambda}$. We can write the following matrix product if we want to scale an object up to a new size, shear the object

[7] It can be generalized as affine transformations of the form $d(\underline{\mathbf{s}}_n) = \boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_n + \underline{\mathbf{u}}$ and $d(\underline{\mathbf{s}}_{n'}) = \boldsymbol{\Lambda}\cdot\underline{\mathbf{s}}_{n'} + \underline{\mathbf{u}}$ for any translation vector $\underline{\mathbf{u}} \in \mathbb{R}^2$, but Algorithm 6 is not able to retrieve the vector $\underline{\mathbf{u}}$.

to a new shape[8], and rotate it:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \xi_x & 0 \\ 0 & \xi_y \end{bmatrix} \cdot \begin{bmatrix} 1 & \zeta_x \\ \zeta_y & 1 \end{bmatrix} \cdot \begin{bmatrix} \cos\eta & -\sin\eta \\ \sin\eta & \cos\eta \end{bmatrix},$$

where ξ_x and ξ_y scale the x and y coordinates of a point, ζ_x is a horizontal shear factor, ζ_y is a vertical shear factor, and η is an angle of counterclockwise rotation around the origin.

We consider $q = 2$ response variables, $p = 2$ regression coefficients per time and response variable, and $T + T^\star = 110$ periods of time, where the first $T = 100$ times are used to fit the model and the last $T^\star = 10$ times are generated to study the forecasting performance. We simulate the process Y at $N = 16$ equally spaced points in the unit square and, jointly, also simulate the process Y^* at $N^* = 3$ not equally spaced points in the unit square. We are interested in interpolating $\mathbf{Y}_t^*$ using the anisotropic and isotropic models given by Equations (2.3) and (2.19) after observing $\mathbf{Y}_t$, and compare to their real generated values. Figure 2.10 shows the geographic region and highlights the anchor points ($\underset{\sim}{\mathbf{s}}_1$ and $\underset{\sim}{\mathbf{s}}_2$), the non-anchor points ($\underset{\sim}{\mathbf{s}}_3, \ldots, \underset{\sim}{\mathbf{s}}_{16}$) and the sites considered as ungauged to study the interpolation performance ($\underset{\sim}{\mathbf{s}}_{17}$, $\underset{\sim}{\mathbf{s}}_{18}$ and $\underset{\sim}{\mathbf{s}}_{19}$). We generate data from the following anisotropic scheme:

1. Generate $\mathbf{D} = \begin{bmatrix} \mathbf{D}_{1:2} & \mathbf{D}_{3:16} \end{bmatrix}_{2\times 16}$ and $\mathbf{D}^* = \begin{bmatrix} \mathbf{D}_{17} & \mathbf{D}_{18} & \mathbf{D}_{19} \end{bmatrix}_{2\times 3}$ proceeding as follows:
   (a) Do $\mathbf{D}_{1:2} = \mathbf{S}_{1:2}$ (i.e. $d(\underset{\sim}{\mathbf{s}}_1) = \underset{\sim}{\mathbf{s}}_1$ and $d(\underset{\sim}{\mathbf{s}}_2) = \underset{\sim}{\mathbf{s}}_2$) to fix the locations of two sites in $\mathcal{D}$-space (i.e. $\xi_x = \xi_y = 1$ and $\eta = \zeta_x = \zeta_y = 0$), where $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{1:2} & \mathbf{S}_{3:16} \end{bmatrix}_{2\times 16}$.
   (b) Do $\underset{\sim}{\mathbf{d}}_n = d(\underset{\sim}{\mathbf{s}}_n) = \boldsymbol{\Lambda} \cdot \underset{\sim}{\mathbf{s}}_n$ for $n \in \{3, \ldots, 16, 17, 18, 19\}$ to generate deterministic deformations that induce geometric anisotropy, where:

$$\begin{aligned} \boldsymbol{\Lambda} &= \begin{bmatrix} 1 & 0 \\ 0 & 3 \end{bmatrix} \cdot \begin{bmatrix} 1.00 & 0.05 \\ 0.00 & 1.00 \end{bmatrix} \cdot \begin{bmatrix} \cos\left(-\frac{\pi}{6}\right) & -\sin\left(-\frac{\pi}{6}\right) \\ \sin\left(-\frac{\pi}{6}\right) & \cos\left(-\frac{\pi}{6}\right) \end{bmatrix} \\ &= \begin{bmatrix} 0.8410254 & 0.54330127 \\ -1.5000000 & 2.59807621 \end{bmatrix}. \end{aligned}$$

2. Set $V = 1.0$, $\phi = 0.5$ and $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 1.0 & 0.8 \\ 0.8 & 1.0 \end{smallmatrix}\right]$ as true parameters, generate $\boldsymbol{\beta}_0$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{0}_{2\times 2}, V \cdot \mathbf{I}_2, \boldsymbol{\Sigma})$ and compute the block matrix $\mathbf{B}^{\mathsf{aug}} = \left[B^{\mathsf{aug}}_{n,n'}\right]_{19\times 19} = \begin{bmatrix} \mathbf{B} & \mathbf{B}_{\mathsf{g,u}} \\ \mathbf{B}_{\mathsf{u,g}} & \mathbf{B}^* \end{bmatrix}$, where $\mathbf{B}$ (16×16), $\mathbf{B}_{\mathsf{g,u}}$ (16×3), $\mathbf{B}_{\mathsf{u,g}}$ (3×16) and $\mathbf{B}^*$ (3×3) are four sub-matrices and, for all

[8] See Lay et al. (2022, Sec. 1.9) for a review about shears from a geometric point of view. Twiss and Moores (2007, Chapter 12) discuss geometric aspects of shear in the context of geology.

$n, n' \in \{1, \ldots, 16, 17, 18, 19\}$, $B^{\mathsf{aug}}_{n,n'}$ is a generic entry given by:

$$B^{\mathsf{aug}}_{n,n'} = \begin{cases} \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'}) \|\}, & n \neq n' \\ 1, & n = n' \end{cases}.$$

These values were chosen to have $\phi \cdot V \cdot \Sigma_{1,1} = \phi \cdot V \cdot \Sigma_{2,2} = \phi$, with the aim of overcoming the problem of lack of identifiability. This way, it will be possible to study the estimation of ϕ and $V \cdot \Sigma_{i,i'}$ separately and by model ($\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$).

3. For $t = 1, \ldots, 100, 101, \ldots, 110$ do the following:

   (a) Generate $\boldsymbol{\beta}_t$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{G}_t \boldsymbol{\beta}_{t-1}, V \cdot \mathbf{W}, \boldsymbol{\Sigma})$, where $\mathbf{G}_t = \mathbf{I}_2$ and $\mathbf{W} = \mathbf{I}_2$.

   (b) Generate $\mathbf{X}_t = [\,{}^{1}_{U_{1,t}}\ {}^{1}_{U_{2,t}}\ \cdots\ {}^{1}_{U_{16,t}}\,]^{\intercal}$ (16×2) and $\mathbf{X}^*_t = [\,{}^{1}_{U_{17,t}}\ {}^{1}_{U_{18,t}}\ {}^{1}_{U_{19,t}}\,]^{\intercal}$ (3×2), where $U_{1,t}, \ldots, U_{19,t}$ are nineteen random numbers between zero and one.

   (c) Generate $\mathbf{Y}_t$ and $\mathbf{Y}^*_t$ jointly from the following distribution:

$$\mathsf{N}_{19\times 2}\left(\begin{bmatrix} \mathbf{X}_t \\ \mathbf{X}^*_t \end{bmatrix} \cdot \boldsymbol{\beta}_t, V \cdot \mathbf{B}^{\mathsf{aug}}, \boldsymbol{\Sigma} \right).$$

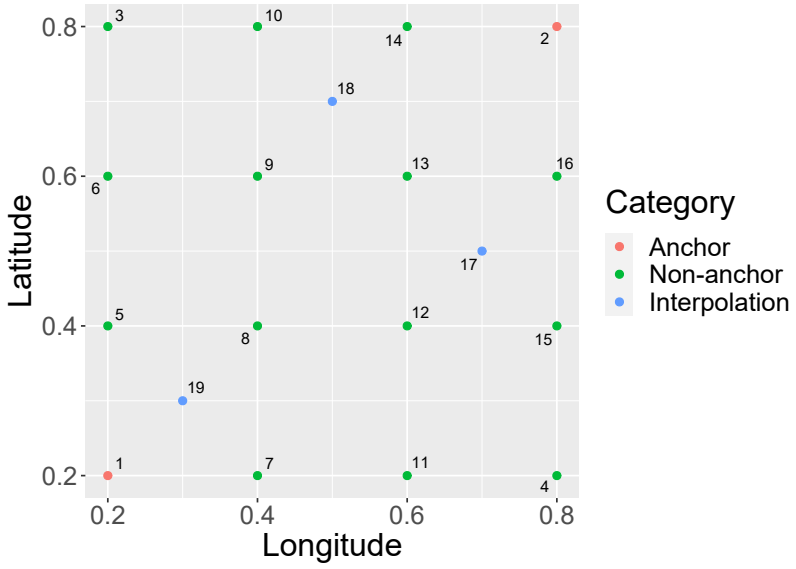


Figure 2.10: Geographic region ($\mathcal{S}$) of the second simulation study in Chapter 2.

To run Algorithm 6, we use the fixed matrices $\mathbf{W}$, $\mathbf{G}_t$, $\mathbf{X}_t$ and $\mathbf{Y}_t$ generated on 3(a)-(c) for $t = 1, \ldots, 100$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\boldsymbol{\Sigma}} = 0.001$, $\mathbf{b}_{\boldsymbol{\Sigma}} = 0.001 \cdot \mathbf{I}_2$, $a_\phi = 0.001$, $b_\phi = 0.001$, $\mathbf{M}_0 = \mathbf{0}_{2\times 2}$, $\mathbf{C}_0 = \mathbf{I}_2$, and $\boldsymbol{\sigma}^2_d$ as the empirical

covariance matrix of the gauged sites. We choose $\psi = 15.0$ after a few tries. In Appendix B.1.4, we present a code implemented in Python for the data generation scheme related to the second simulation study.

Algorithm 6 was run for 20000 iterations, having converged after $J \approx 5000$ iterations. We applied the slice sampler in its 6^{th} step. The same configuration was adopted for sampling from the analogous isotropic model. To avoid autocorrelation in the chains, we form a sample of size $K = 1000$ of the posterior distribution of the parameters by systematically sampling every 15 iterations (i.e. $j_1 = 5001, j_2 = 5016, \ldots, j_{1000} = 19986$). The acceptance rates of the parameter ϕ are equal to 44.80% and 44.04%, respectively for the models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$. The trace plots for the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$, ϕ and $\mathbf{D}$ from this simulation study are available in Appendix C.1.2.

True and estimated deformations are shown in Figure 2.11, where visually it can be seen that most of the true deformation points were well estimated[9]. Through the samples $\mathbf{D}^{(j_1)}, \ldots, \mathbf{D}^{(j_{1000})}$, the entries of matrix $\mathbf{\Lambda}$ can be recovered by applying the solution described in Appendix A.4 – it can be verified in Figure 2.12. Figures 2.13(a)-2.13(f), 2.14, 2.15, 2.16 and 2.17 show that the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ and $\beta_{0,1,t}, \beta_{0,2,t}, \beta_{1,1,t}, \beta_{1,2,t}$ $(t = 0, 1, \ldots, 100)$ estimate well their true values for $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, respectively. In addition, Figures 2.13(g) and 2.13(h) show that the parameter ϕ is well estimated only for $\mathcal{M}_{\text{A}}$.

Table 2.2 shows some metrics for comparing models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$. Based on the DIC and PMSE statistics, $\mathcal{M}_{\text{A}}$ fits the data better than $\mathcal{M}_{\text{I}}$. Figures 2.18 and 2.19 show that the forecast performance of both models are similar at the gauged site $\underset{\sim}{\mathbf{s}}_1$, for both response variables. To facilitate visual comparison of the interpolations of each model, we divide the time series into two periods: $t \in \{1, \ldots, 55\}$ and $t \in \{56, \ldots, 110\}$. Figures 2.20, 2.21, 2.22 and 2.23 show that the interpolation performance of model $\mathcal{M}_{\text{A}}$ is superior to the one of model $\mathcal{M}_{\text{I}}$ at the ungauged site $\underset{\sim}{\mathbf{s}}_{18}$, for both response variables, noting that their credibility intervals are narrower, which is in agreement with the results of the ECP and IS statistics. The histograms of interpolated values at the second ungauged site and for time $t = 10$ are in Figures 2.24(a), 2.24(b), 2.24(c) and 2.24(d), for both response variables and both models, and it shows that the results are more accurate for the anisotropic model.

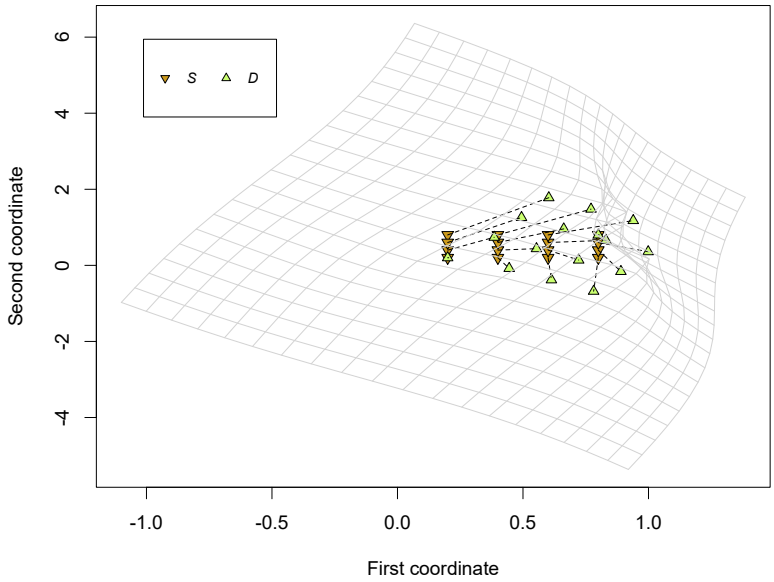
Thus, through this simulation study we concluded that for analyzing data exhibiting geometric anisotropy, as per our data generation method, model $\mathcal{M}_{\text{A}}$ (with spatial deformation) makes more

[9]Figures C.10(a), C.10(g) and C.11(a) show that only the first coordinates of points $\underset{\sim}{\mathbf{d}}_3$, $\underset{\sim}{\mathbf{d}}_6$ and $\underset{\sim}{\mathbf{d}}_{10}$ (i.e. $D_{1,3}$, $D_{1,6}$ and $D_{1,10}$) were not well estimated, respectively.

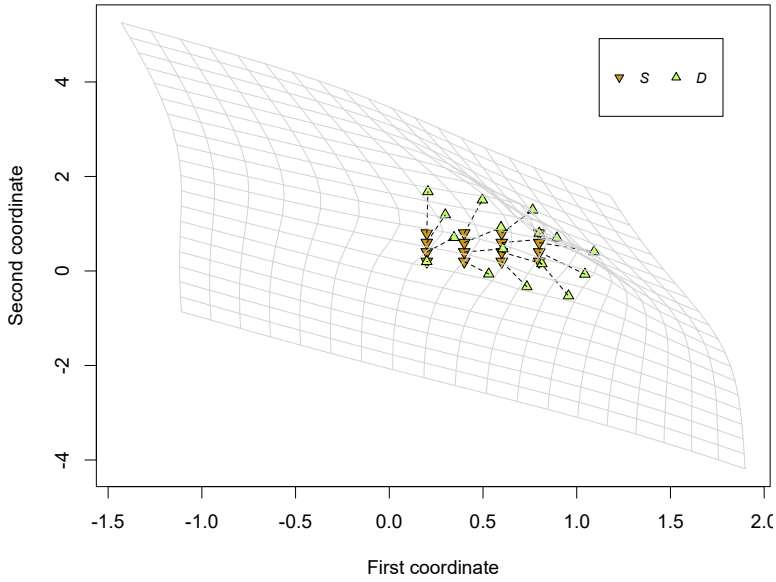
assertive interpolations than model $\mathcal{M}_{\mathrm{I}}$ (without spatial deformation).

Table 2.2: Metrics for model comparison (DIC, PMSE, ECP and IS by response variable and ungauged site), from the second simulation study in Chapter 2.

Metric	$\mathcal{M}_{\mathrm{A}}$		$\mathcal{M}_{\mathrm{I}}$	
DIC	3666.0		4157.8	
PMSE	0.1076		0.1350	
ECP (%)	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{17}$	96.0	98.0	98.0	100.0
$\underset{\sim}{\mathbf{s}}_{18}$	96.0	96.0	96.0	96.0
$\underset{\sim}{\mathbf{s}}_{19}$	96.0	93.0	95.0	91.0
IS	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{17}$	0.0360	0.0345	0.0408	0.0376
$\underset{\sim}{\mathbf{s}}_{18}$	0.0347	0.0394	0.0432	0.0430
$\underset{\sim}{\mathbf{s}}_{19}$	0.0405	0.0462	0.0435	0.0496

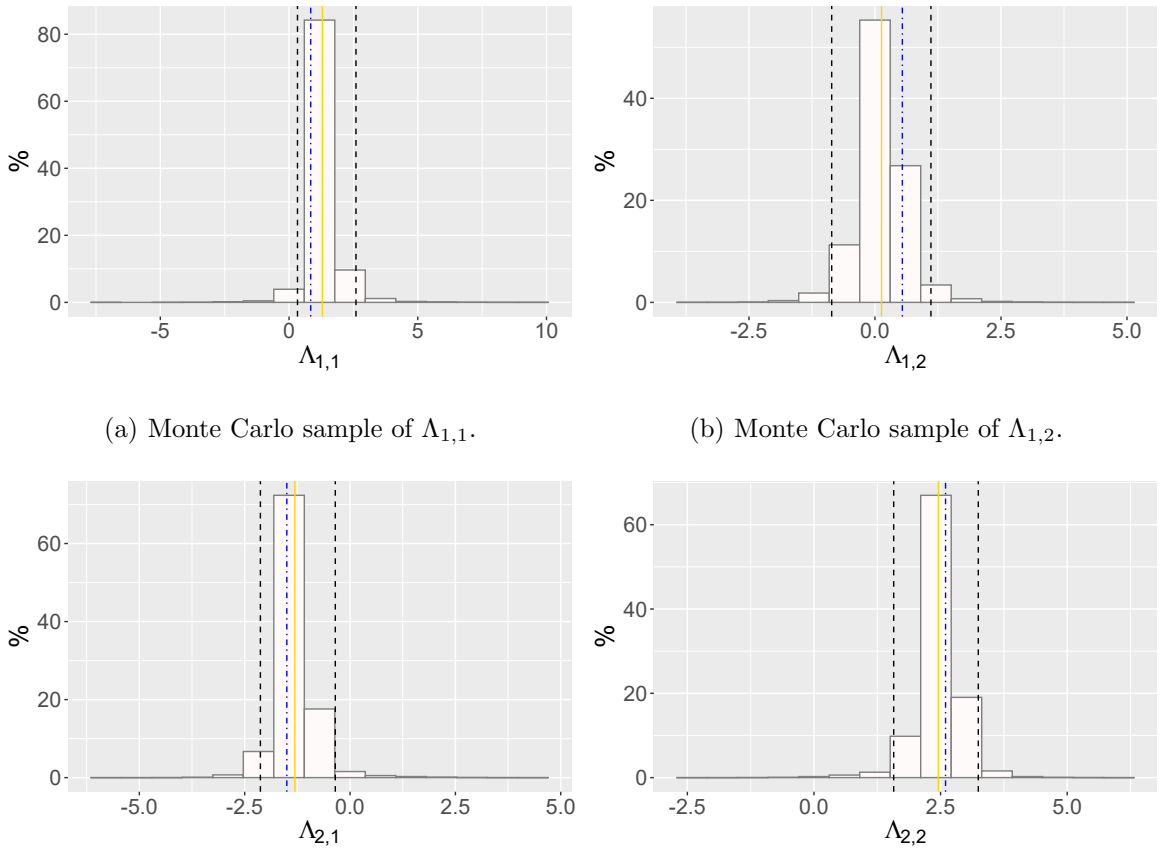


(a) True deformation.



(b) Estimated deformation.

Figure 2.11: True deformation and posterior means of **D**, resulting from the second simulation study in Chapter 2.



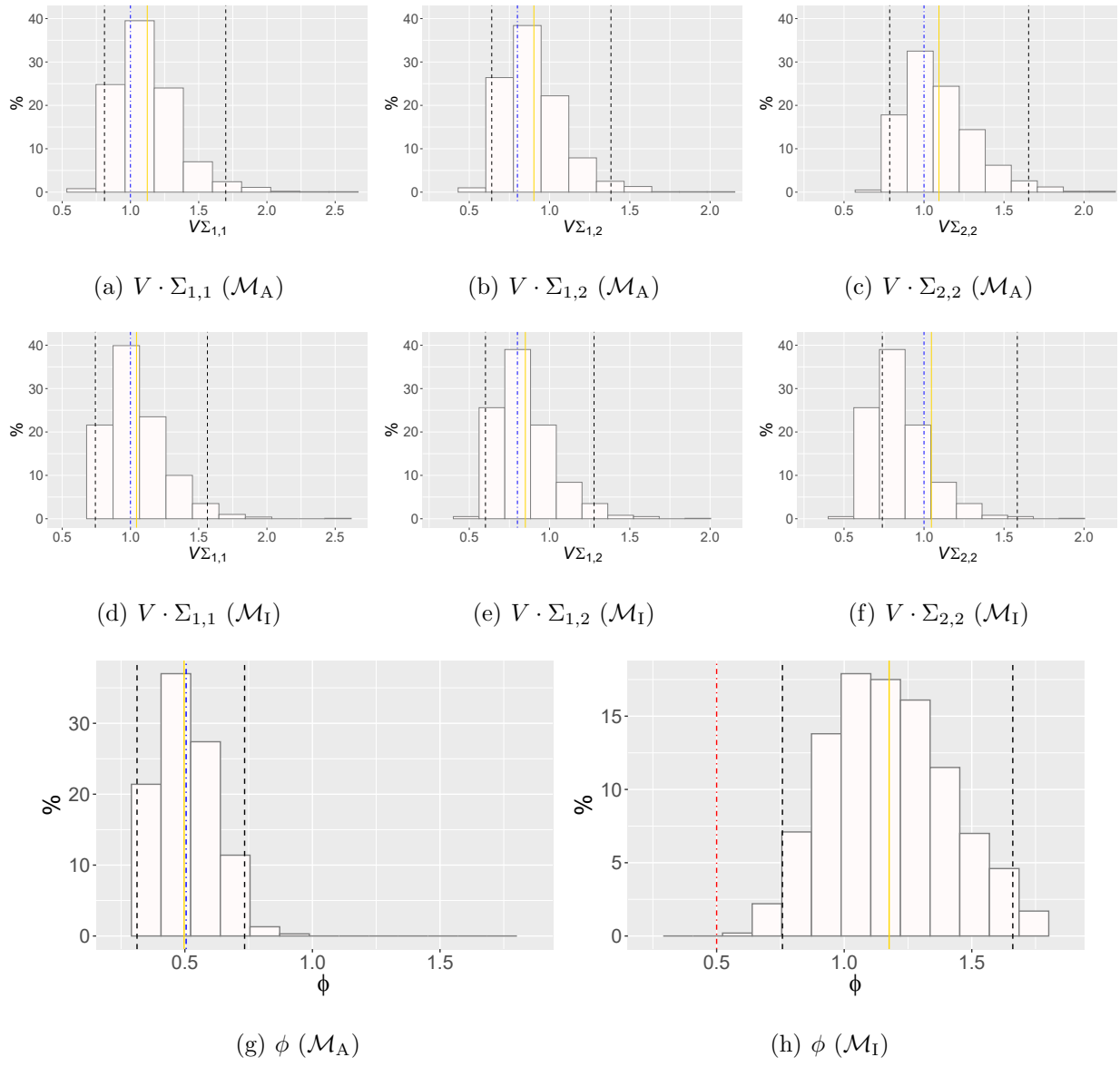
(a) Monte Carlo sample of $\Lambda_{1,1}$.

(b) Monte Carlo sample of $\Lambda_{1,2}$.

(c) Monte Carlo sample of $\Lambda_{2,1}$.

(d) Monte Carlo sample of $\Lambda_{2,2}$.

Figure 2.12: Histograms of the Monte Carlo samples of the parameters $\Lambda_{1,1}$, $\Lambda_{1,2}$, $\Lambda_{2,1}$ and $\Lambda_{2,2}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).



(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$)

(g) ϕ ($\mathcal{M}_{\mathrm{A}}$) (h) ϕ ($\mathcal{M}_{\mathrm{I}}$)

Figure 2.13: Histograms of the posterior distributions of $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ and ϕ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

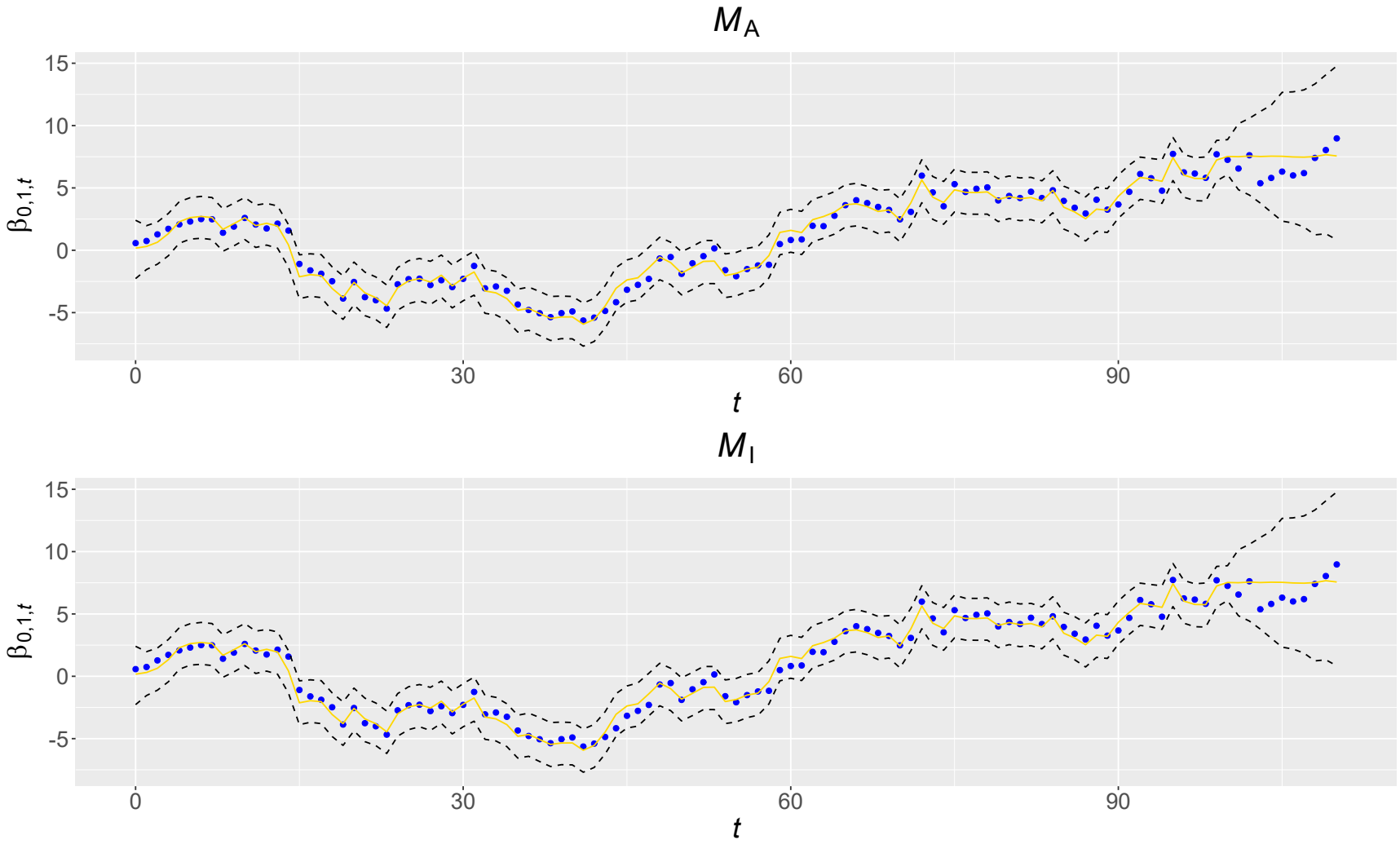


Figure 2.14: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

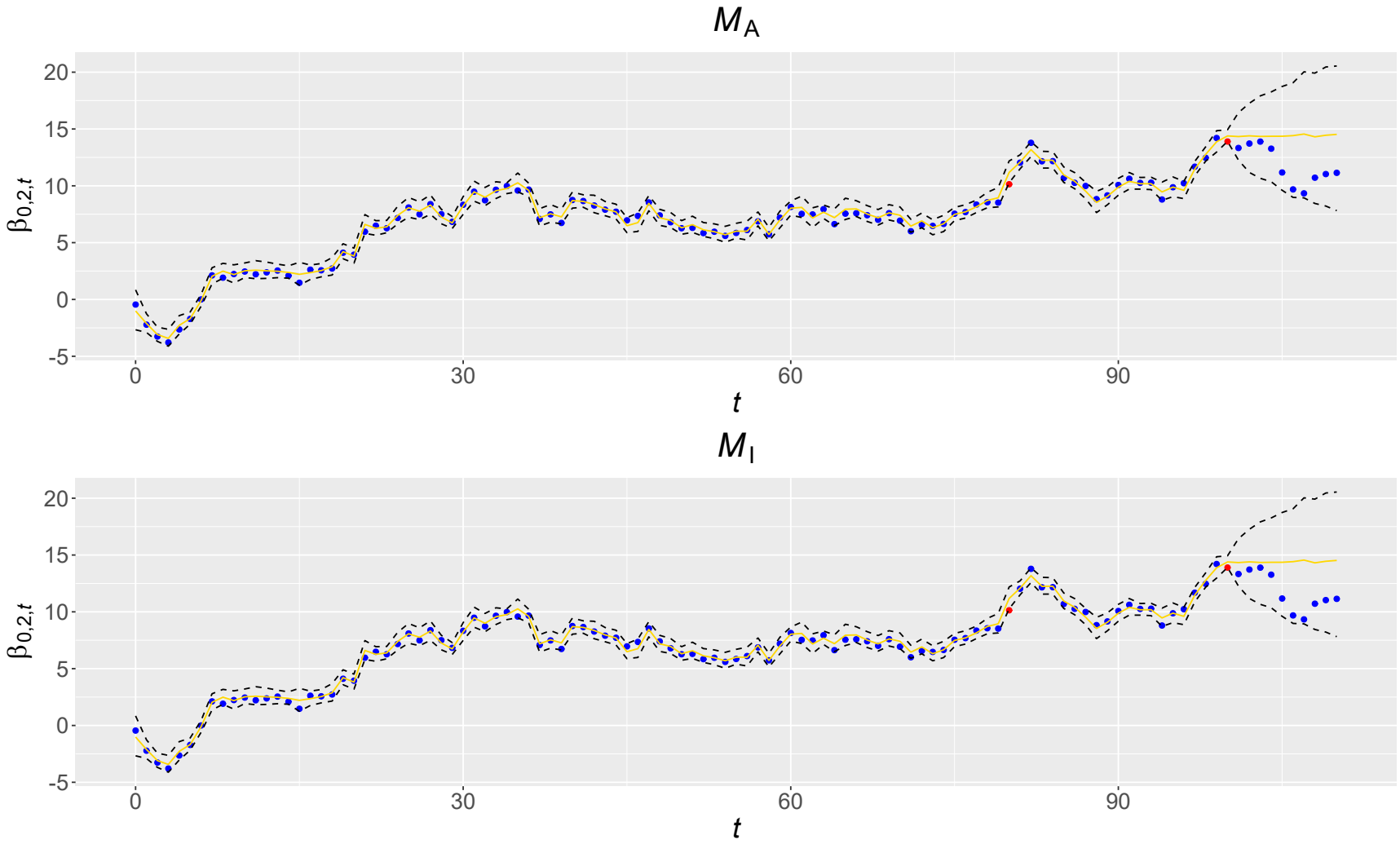


Figure 2.15: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

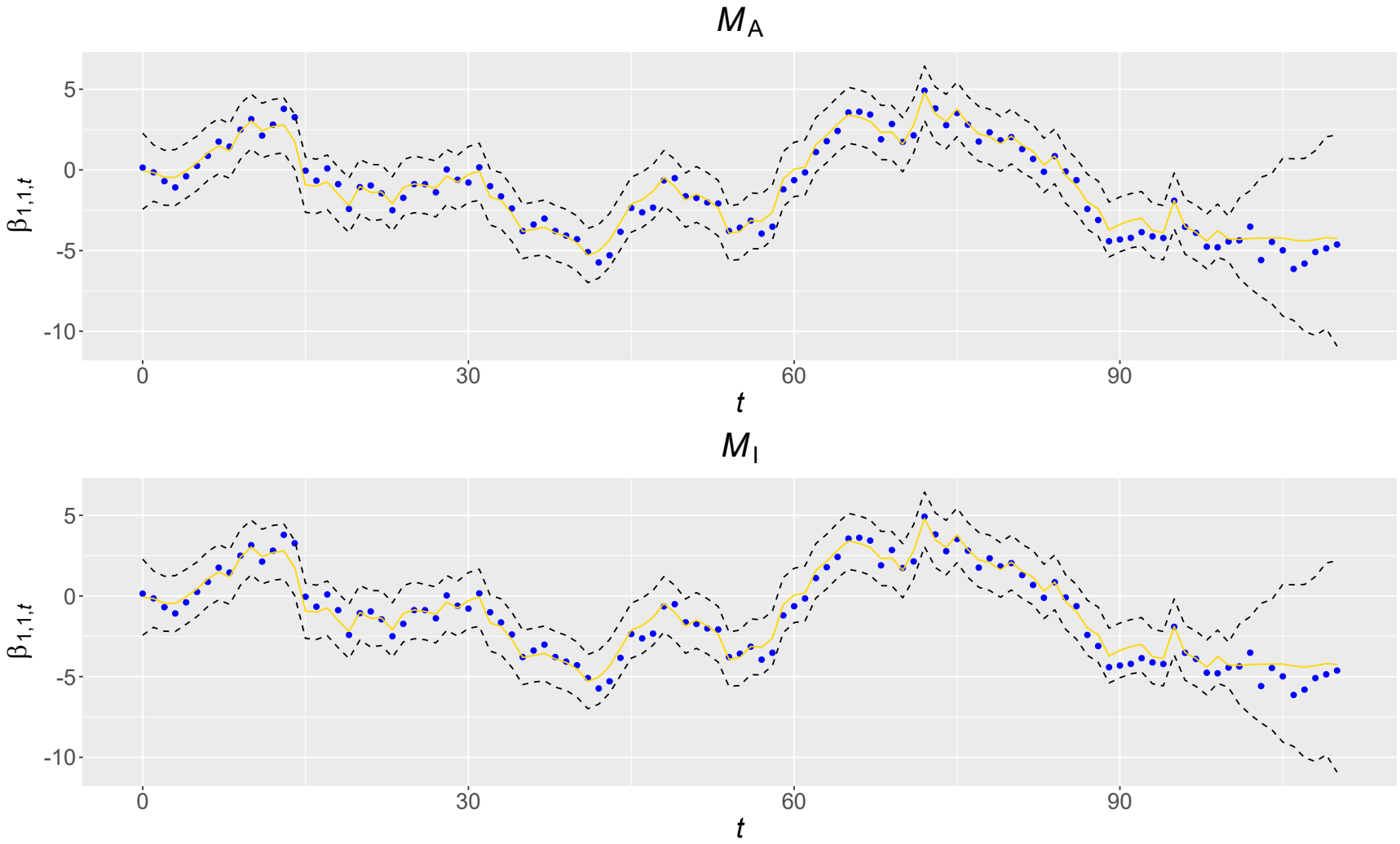


Figure 2.16: Line chart of the posterior distribution of $\beta_{1,1,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

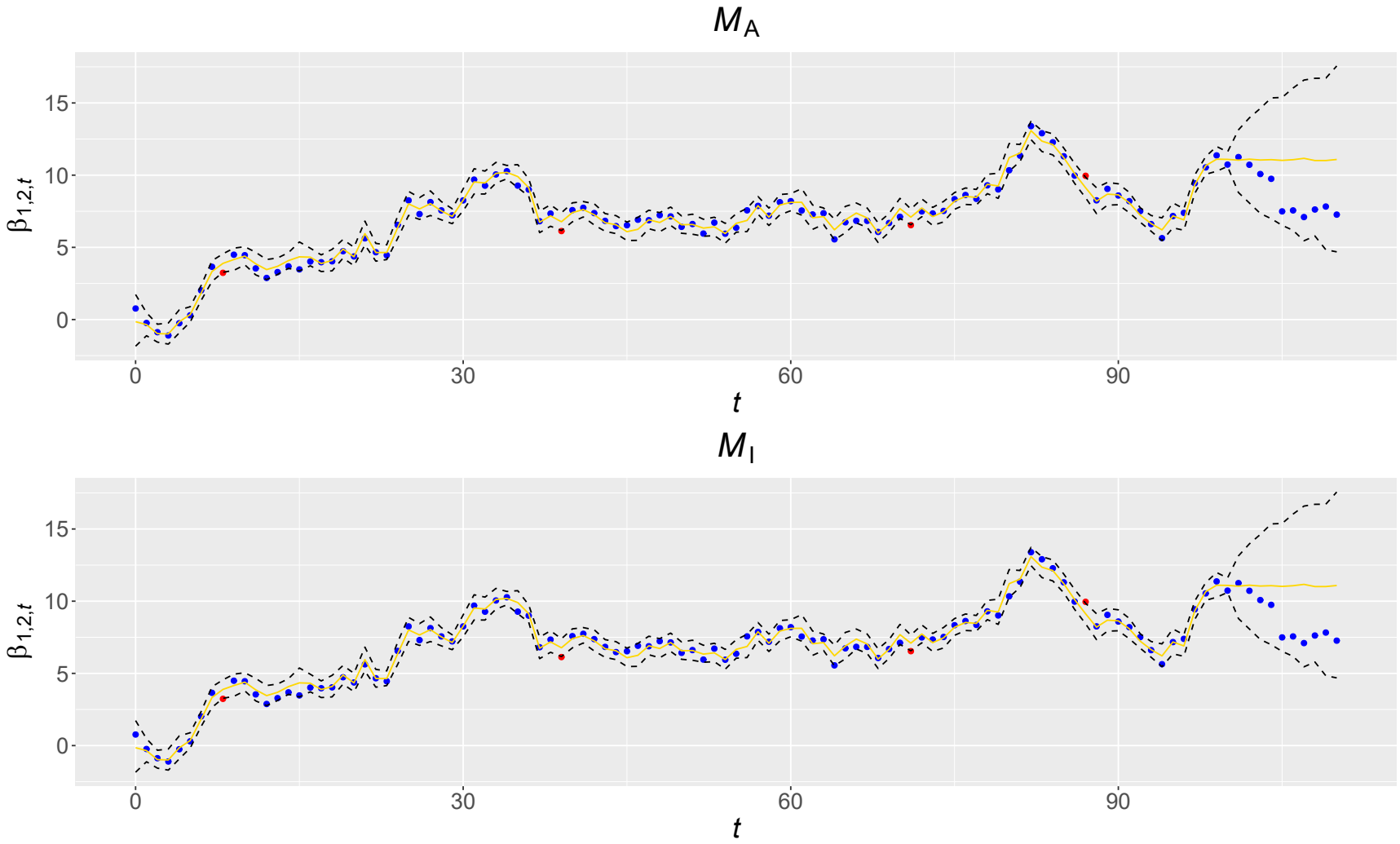


Figure 2.17: Line chart of the posterior distribution of $\beta_{1,2,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

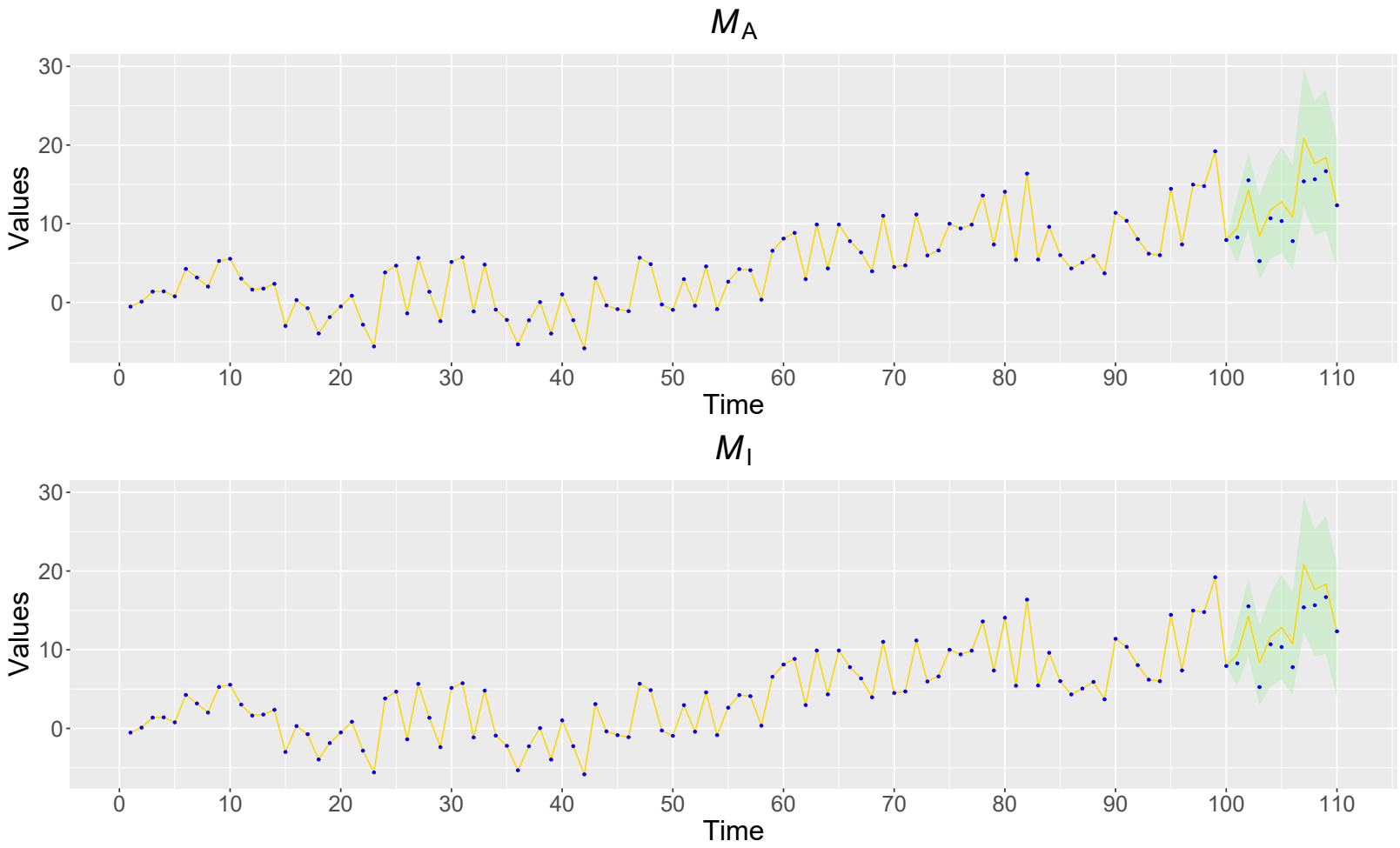


Figure 2.18: Line chart of $Y_{n,i,t}$ for $n = 1$, $i = 1$ and $t \in \{1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). For $t \leq 100$, the solid gold line connects the true values; for $t > 100$, it connects the posterior means.

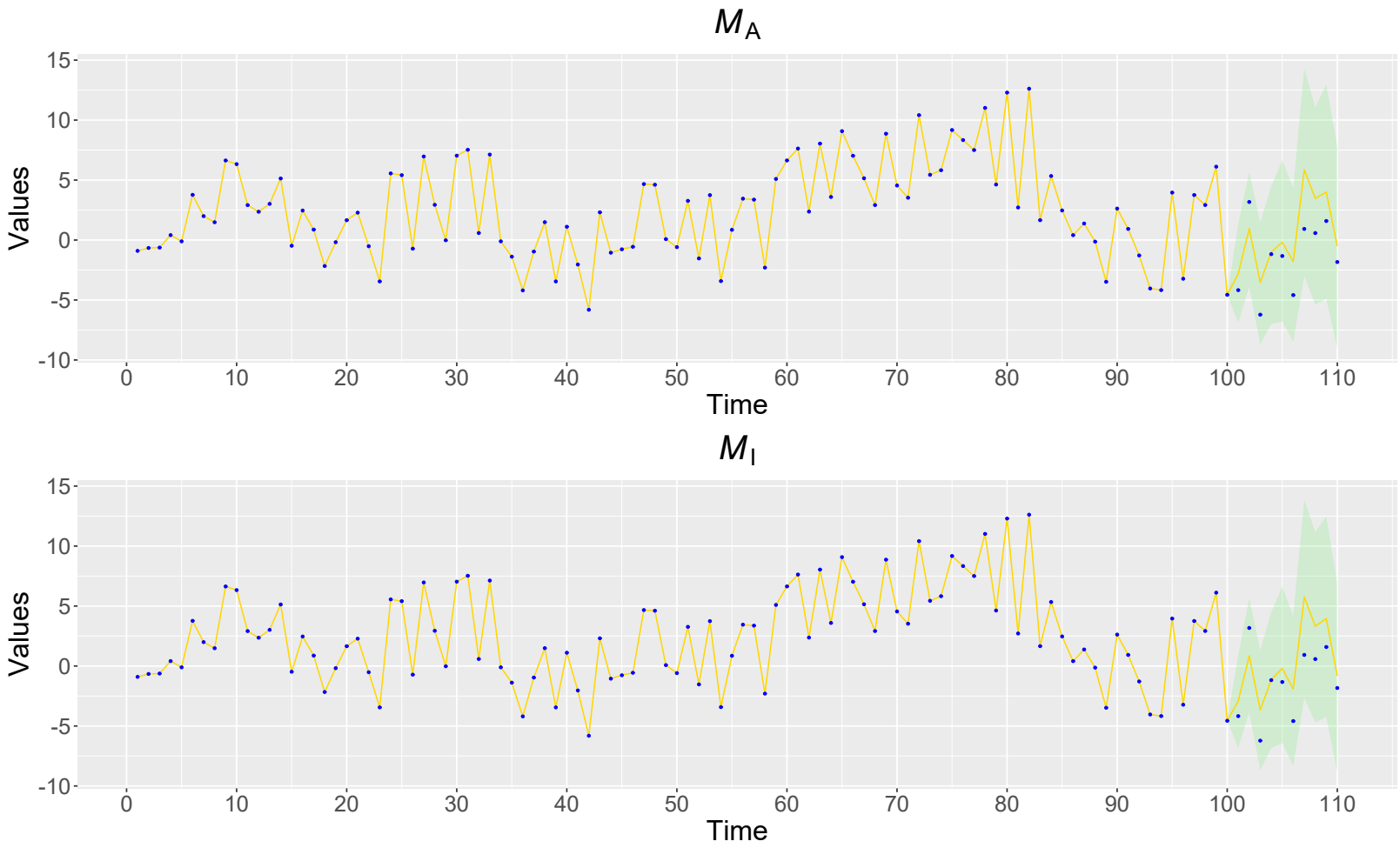


Figure 2.19: Line chart of $Y_{n,i,t}$ for $n = 1$, $i = 2$ and $t \in \{1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). For $t \leq 100$, the solid gold line connects the true values; for $t > 100$, it connects the posterior means.

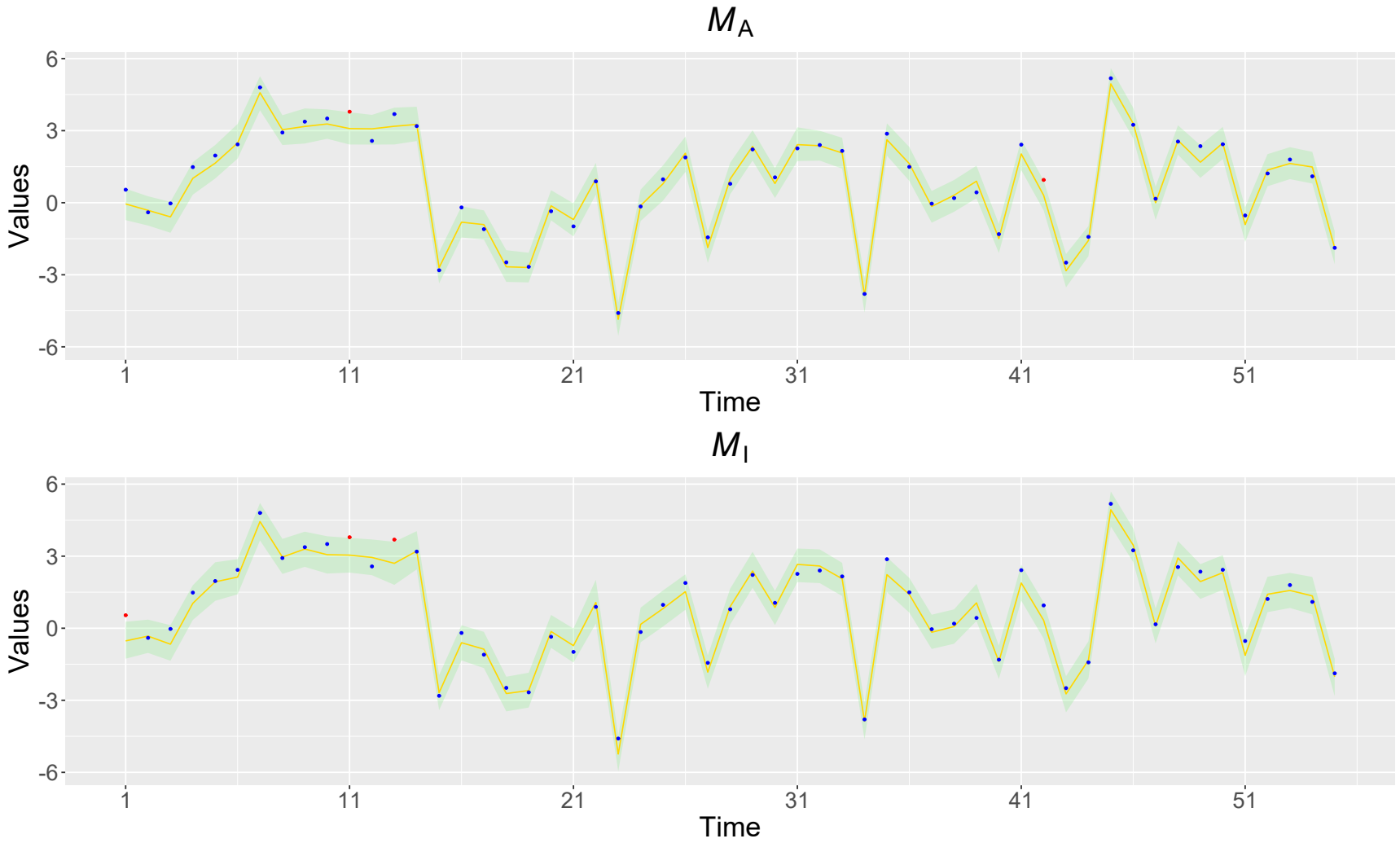


Figure 2.20: Line chart of the interpolated values of $Y_{16+n,i,t}$ for $n = 2$, $i = 1$ and $t \in \{1, \ldots, 55\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

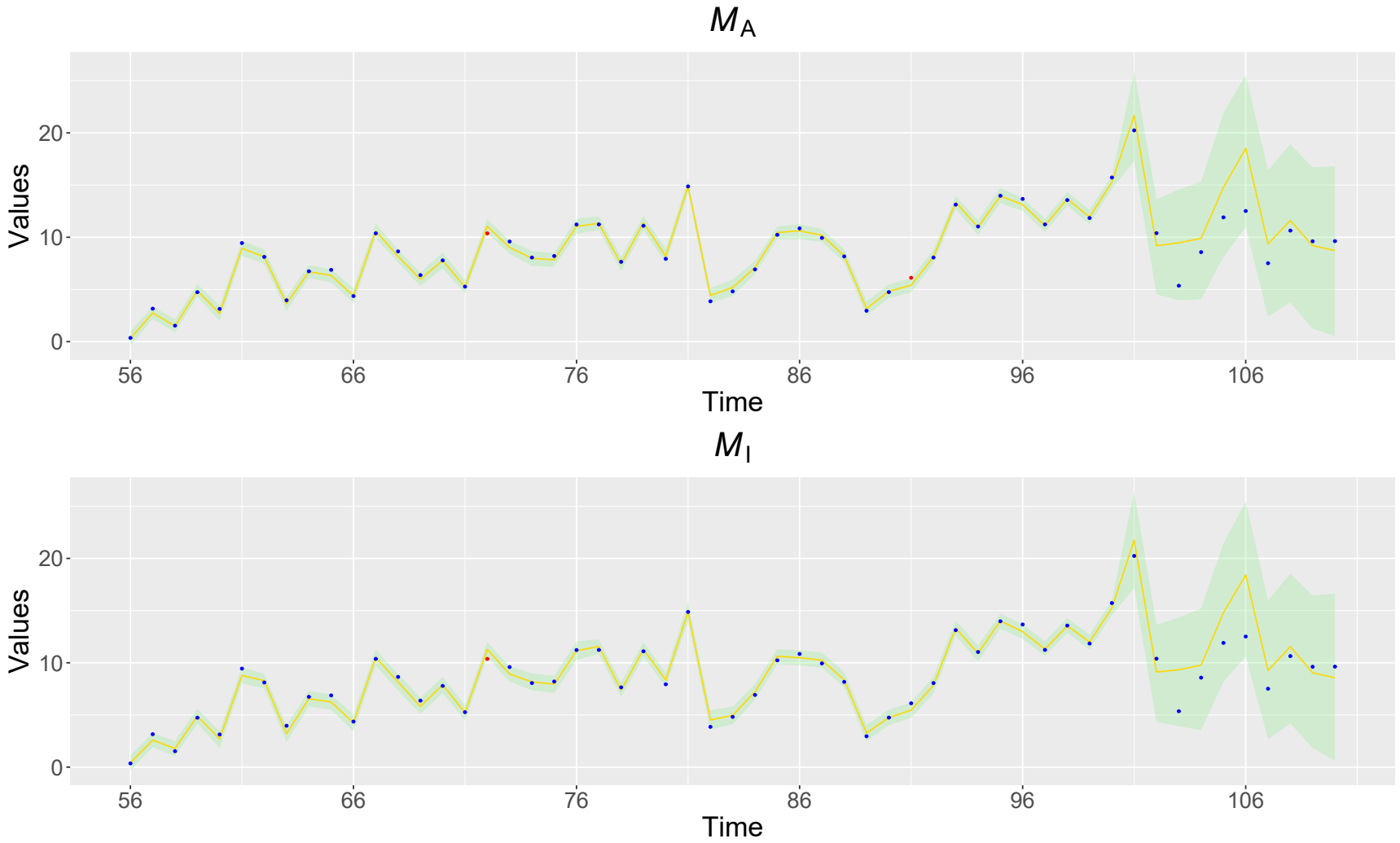


Figure 2.21: Line chart of the interpolated values of $Y_{16+n,i,t}$ for $n = 2$, $i = 1$ and $t \in \{56, \ldots, 110\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

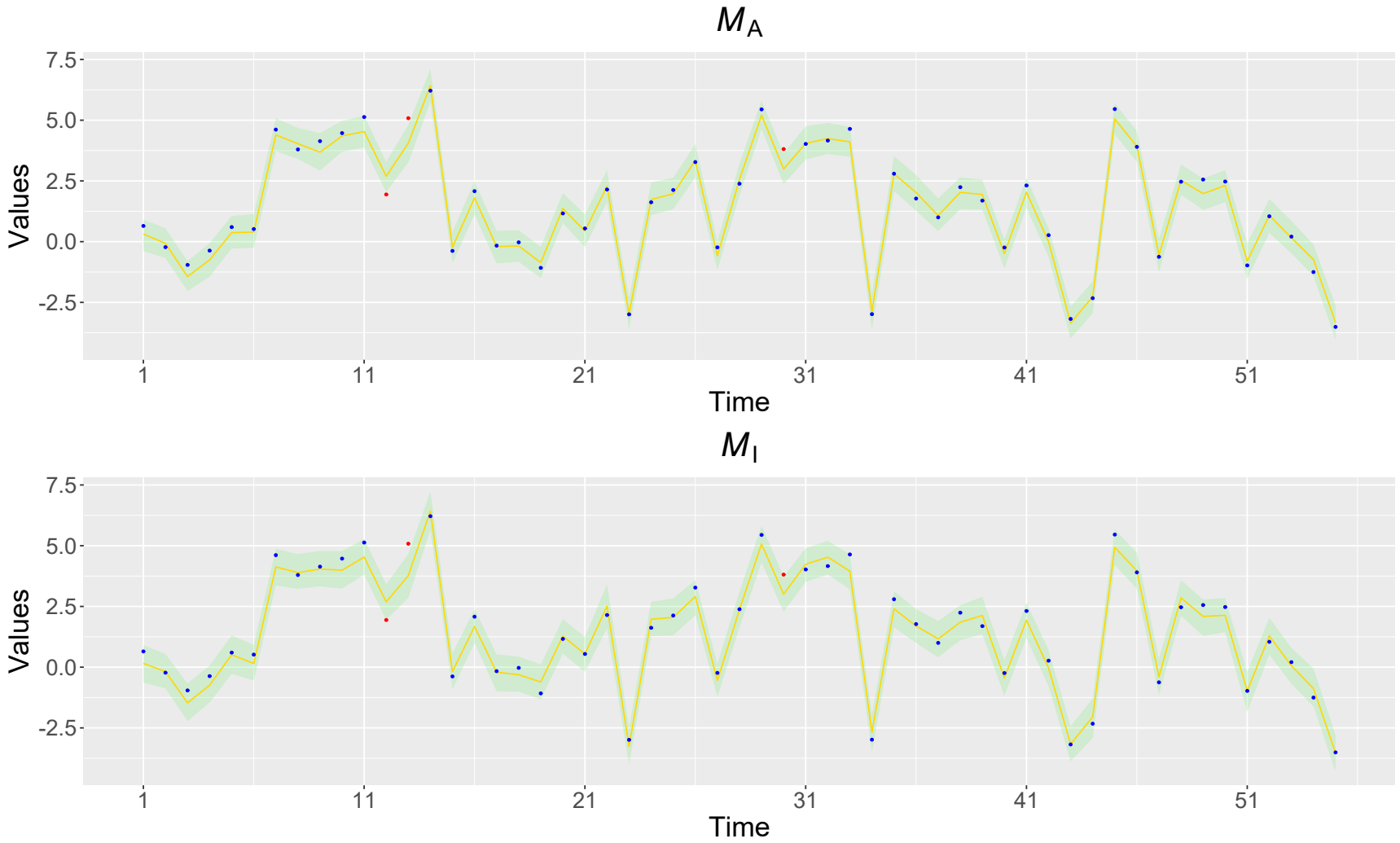


Figure 2.22: Line chart of the interpolated values of $Y_{16+n,i,t}$ for $n = 2$, $i = 2$ and $t \in \{1, \ldots, 55\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

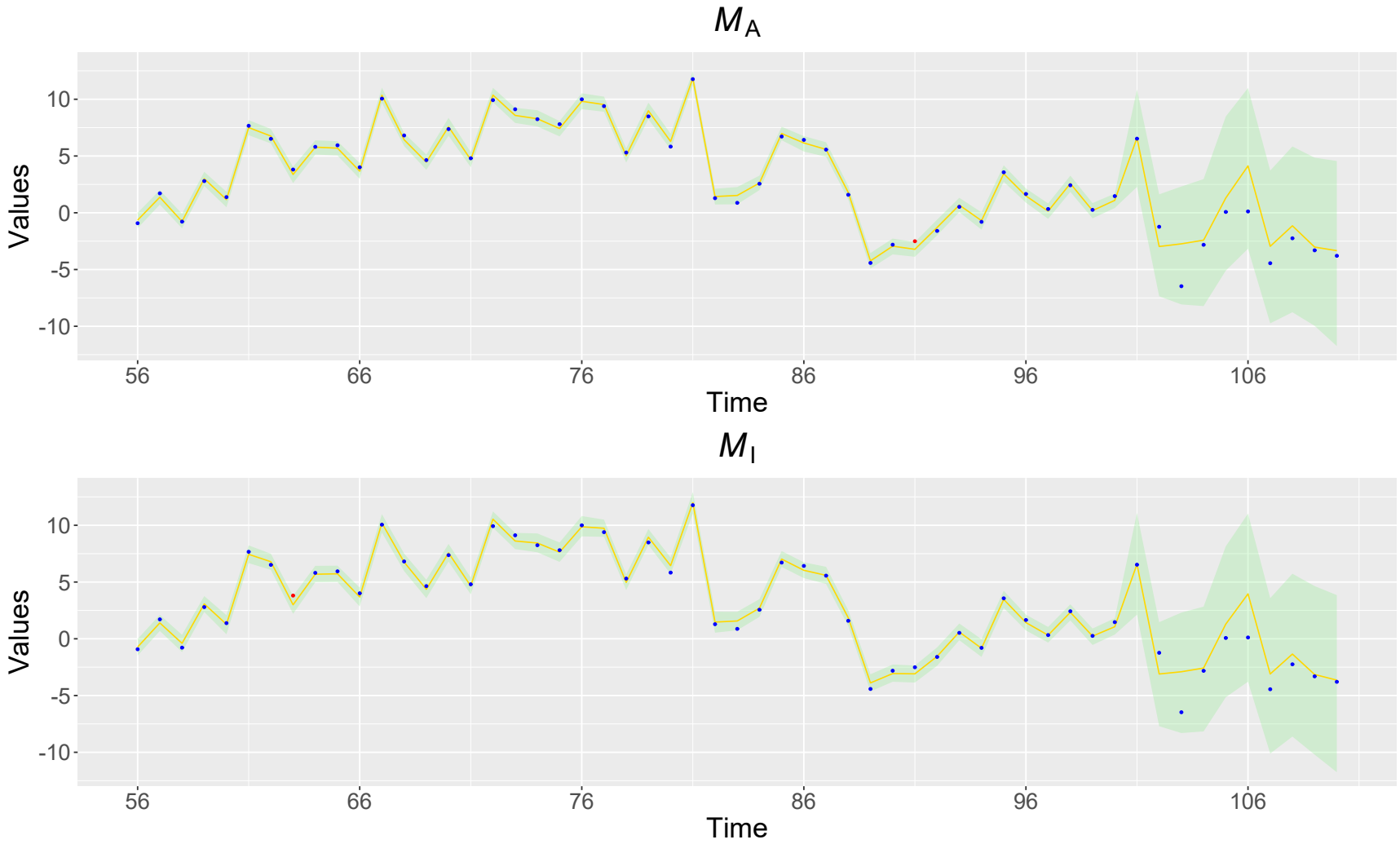
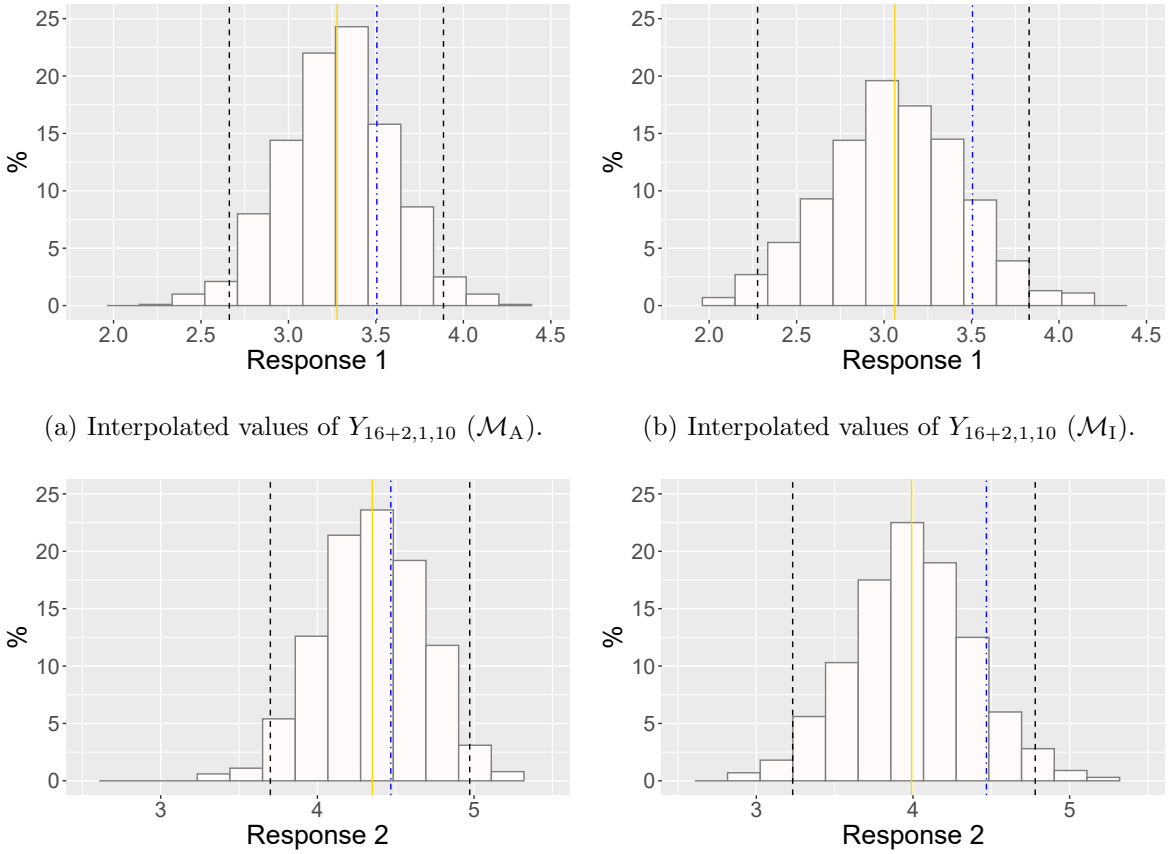


Figure 2.23: Line chart of the interpolated values of $Y_{16+n,i,t}$ for $n = 2$, $i = 2$ and $t \in \{56, \ldots, 110\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).



(a) Interpolated values of $Y_{16+2,1,10}$ ($\mathcal{M}_{\mathrm{A}}$).

(b) Interpolated values of $Y_{16+2,1,10}$ ($\mathcal{M}_{\mathrm{I}}$).

(c) Interpolated values of $Y_{16+2,2,10}$ ($\mathcal{M}_{\mathrm{A}}$).

(d) Interpolated values of $Y_{16+2,2,10}$ ($\mathcal{M}_{\mathrm{I}}$).

Figure 2.24: Histograms of the interpolated values of $Y_{16+2,i,10}$ for $i \in \{1, 2\}$ and for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

2.6 Illustrative example

The proposed model ($\mathcal{M}_{\mathrm{A}}$) and its isotropic version ($\mathcal{M}_{\mathrm{I}}$), given by Equations (2.3) and (2.19), respectively, were applied to a dataset on the concentration levels of the following $q = 2$ air quality indices: Response 1 – Ozone (O_3) and Response 2 – Sulfur Dioxide (SO_2). They were measured in the State of New York, United States (US), having been obtained at twelve monitoring stations that measure both pollutants (stations that measure only one of the two indices were not considered),

collected hourly between January 1st, 2021 and December 31st, 2021. We computed the daily median, working with 365 days. The source of the dataset is the US Environmental Protection Agency (URL: https://www.epa.gov/outdoor-air-quality-data).

Of the $12 \cdot 365 = 4380$ observations for each variable, we found that there were 1.67% missing values in Response 1 and 0.73% missing values in Response 2. Triantafyllopoulos (2021, Sec. 5.6) recommends imputing missing values from an application by regression and moving average methods, before analyzing the dataset through dynamic models. In this illustrative example, missing values were imputed via the R package `imputeTS` (Moritz and Bartz-Beielstein, 2017). Due to spatial heterogeneity, the following strategy was adopted for imputation. Given a response variable i and a site $\underset{\sim}{\mathbf{s}}_n$, we use all $T'_{n,i}$ responses that were observed (say $Y_{n,i,t'_1}, \ldots, Y_{n,i,t'_{T'_{n,i}}}$) to impute all remaining $T''_{n,i} = T - T'_{n,i}$ missing observations (say $Y_{n,i,t''_1}, \ldots, Y_{n,i,t''_{T''_{n,i}}}$).

We used $N = 10$ sites and the first $T = 360$ days to fit the model, storing $N^* = 2$ sites to investigate the interpolation performance (Figure 2.25(c) highlights these sites on the map) and the last $T^\star = 5$ days to study the forecasting performance. We also worked with no explanatory variables ($p = 1$), considering $\mathbf{W} = \mathbf{I}_1$ and, for $t \in \{1, \ldots, 360\}$, $\mathbf{G}_t = \mathbf{I}_1$, $\mathbf{X}_t = \mathbf{1}_{10}$ and $\mathbf{X}^*_t = \mathbf{1}_2$.

To run Algorithm 6, we use the fixed matrices $\mathbf{W}$, $\mathbf{G}_t$, $\mathbf{X}_t$ and $\mathbf{Y}_t$ for $t = 1, \ldots, 360$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\boldsymbol{\Sigma}} = 0.001$, $\mathbf{b}_{\boldsymbol{\Sigma}} = 0.001 \cdot \mathbf{I}_2$, $a_\phi = 0.001$, $b_\phi = 0.001$, $\mathbf{M}_0 = \mathbf{0}_{1\times 2}$, $\mathbf{C}_0 = \mathbf{I}_1$, and $\boldsymbol{\sigma}^2_d$ as the empirical covariance matrix of the gauged sites. We choose $\psi = 0.50$ after a few tries.

Algorithm 6 was run 120000 times, having converged after $J \approx 20000$ iterations. We applied the slice sampler in its 6th step. The same configuration was adopted for sampling from the analogous isotropic model. To avoid autocorrelation in the chains, we form a sample of size $K = 1000$ of the posterior distribution of the parameters by systematically sampling every 100 iterations (i.e. $j_1 = 20001, j_2 = 20101, \ldots, j_{1000} = 119901$). The acceptance rates of the parameter ϕ are equal to 43.03% and 42.91%, respectively for the models $\mathcal{M}_\mathrm{A}$ and $\mathcal{M}_\mathrm{I}$. The trace plots for the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$, ϕ and $\mathbf{D}$ from this illustrative example are available in Appendix C.1.3.

The geographic region map and its corresponding estimated deformation[10] are shown in Figure 2.25. The deformed map has some folds and points with more shrinking distances. For models $\mathcal{M}_\mathrm{A}$ and $\mathcal{M}_\mathrm{I}$, Figure 2.26 shows the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ and ϕ, while the posterior distributions of $\beta_{0,1,t}$ and $\beta_{0,2,t}$ $(t = 0, 1, \ldots, 360)$ are shown respectively

[10]The way we obtain the borders of the estimated deformed map is explained in Appendix A.5.

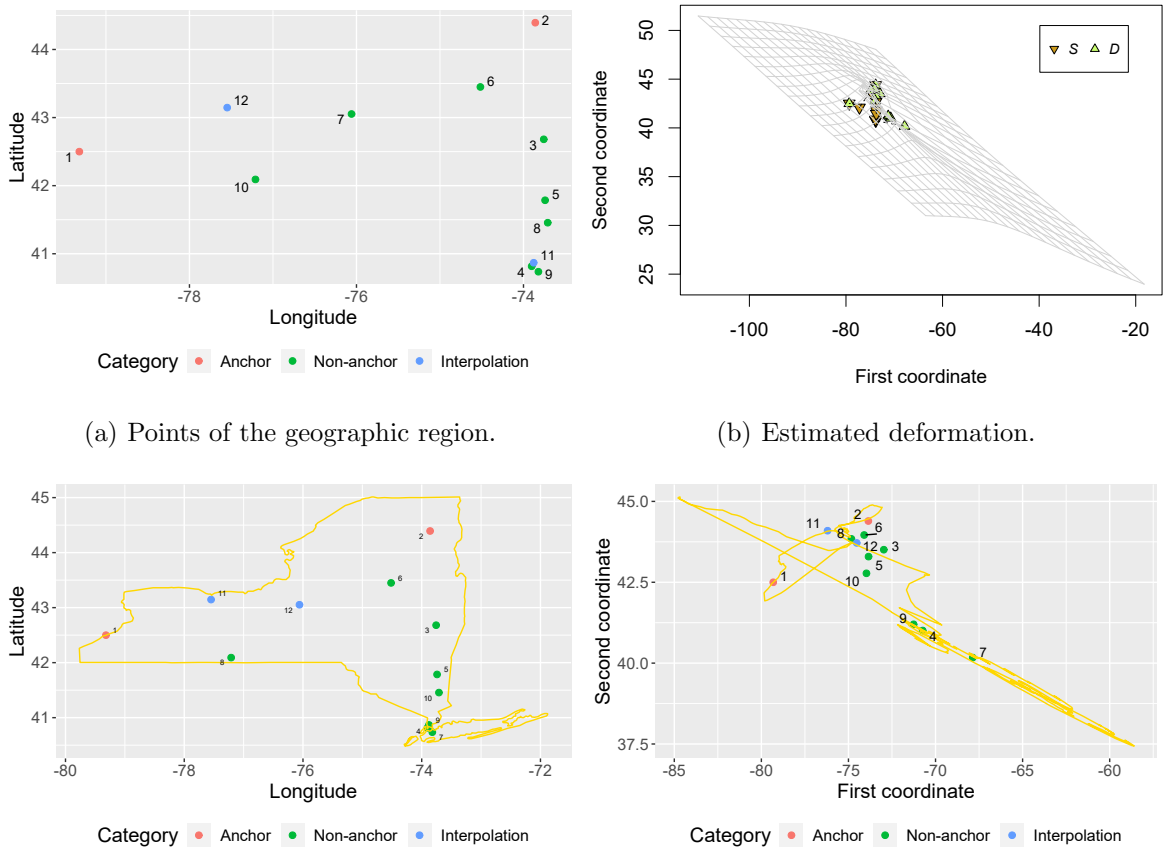
in Figures 2.27 and 2.28. It can be noted that the posterior means of ϕ change a lot depending on the model, as well as that model $\mathcal{M}_{\mathrm{A}}$ provided wider credibility intervals than model $\mathcal{M}_{\mathrm{I}}$ for parameters $\beta_{0,1,t}$ and $\beta_{0,2,t}$.

Table 2.3 shows some metrics for comparing models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$. Based on the DIC and PMSE statistics, $\mathcal{M}_{\mathrm{A}}$ fits the data better than $\mathcal{M}_{\mathrm{I}}$. Figures 2.29 and 2.30 show that the forecast performance considering model $\mathcal{M}_{\mathrm{I}}$ is better than when considering model $\mathcal{M}_{\mathrm{A}}$ at the gauged site $\underset{\sim}{\mathbf{s}}_1$ for both response variables. To facilitate visual comparison of interpolations by model, we divided the time series into two intervals: 1-182 days and 183-365 days. Figures 2.31, 2.32, 2.33 and 2.34 show that the interpolation performance of the model $\mathcal{M}_{\mathrm{A}}$ is superior than the model $\mathcal{M}_{\mathrm{I}}$ at the ungauged site $\underset{\sim}{\mathbf{s}}_{12}$ for the second response variable, while the results for the first response variable are similar. From the figures and also from the ECP and IS statistics, it can be seen that model $\mathcal{M}_{\mathrm{A}}$ has narrower credible intervals and also greater coverage of true values.

Thus, through this illustrative example we concluded that, to analyze real data with some anisotropy, model $\mathcal{M}_{\mathrm{A}}$ (with spatial deformation) makes more assertive interpolation than model $\mathcal{M}_{\mathrm{I}}$ (without spatial deformation) and less assertive forecasts than model $\mathcal{M}_{\mathrm{I}}$.

Table 2.3: Metrics for model comparison (DIC, PMSE, ECP and IS) by response variable and ungauged site, from the illustrative example in Chapter 2.

Metric	$\mathcal{M}_{\mathrm{A}}$		$\mathcal{M}_{\mathrm{I}}$	
DIC	-19134.6		-17733.2	
PMSE	0.0446		0.0560	
ECP (%)	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{11}$	99.2	99.4	97.2	99.4
$\underset{\sim}{\mathbf{s}}_{12}$	98.9	100.0	97.8	99.7
IS	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{11}$	0.00066	0.06537	0.00070	0.08432
$\underset{\sim}{\mathbf{s}}_{12}$	0.00063	0.05813	0.00065	0.08365



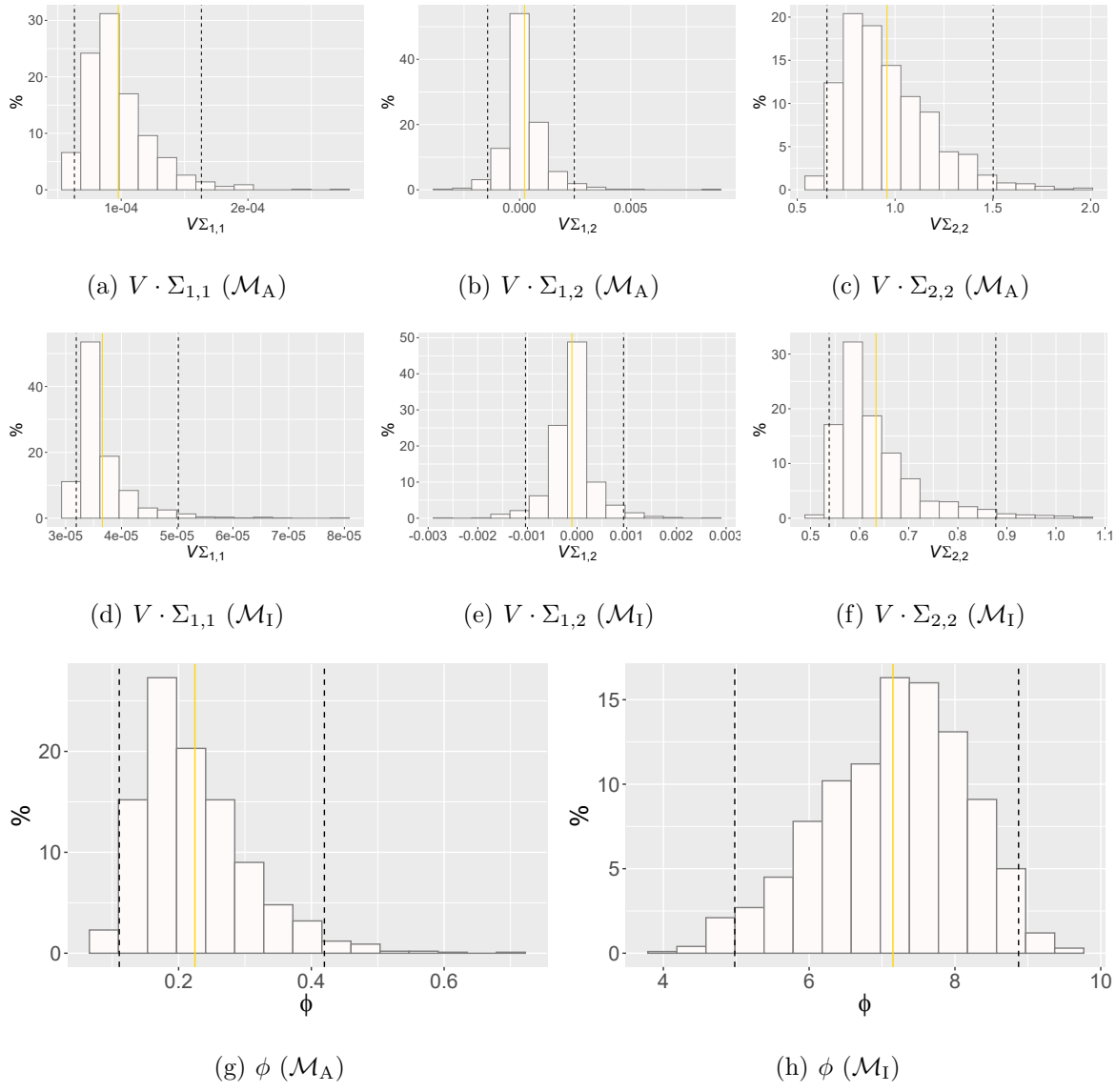
(a) Points of the geographic region.

(b) Estimated deformation.

(c) Geographic region map.

(d) Estimated deformed map.

Figure 2.25: Geographic region map (State of New York, US), estimated deformed map and their sites, resulting from the illustrative example in Chapter 2.



(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$)

(g) ϕ ($\mathcal{M}_{\mathrm{A}}$) (h) ϕ ($\mathcal{M}_{\mathrm{I}}$)

Figure 2.26: Histograms of the posterior distributions of $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ and ϕ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

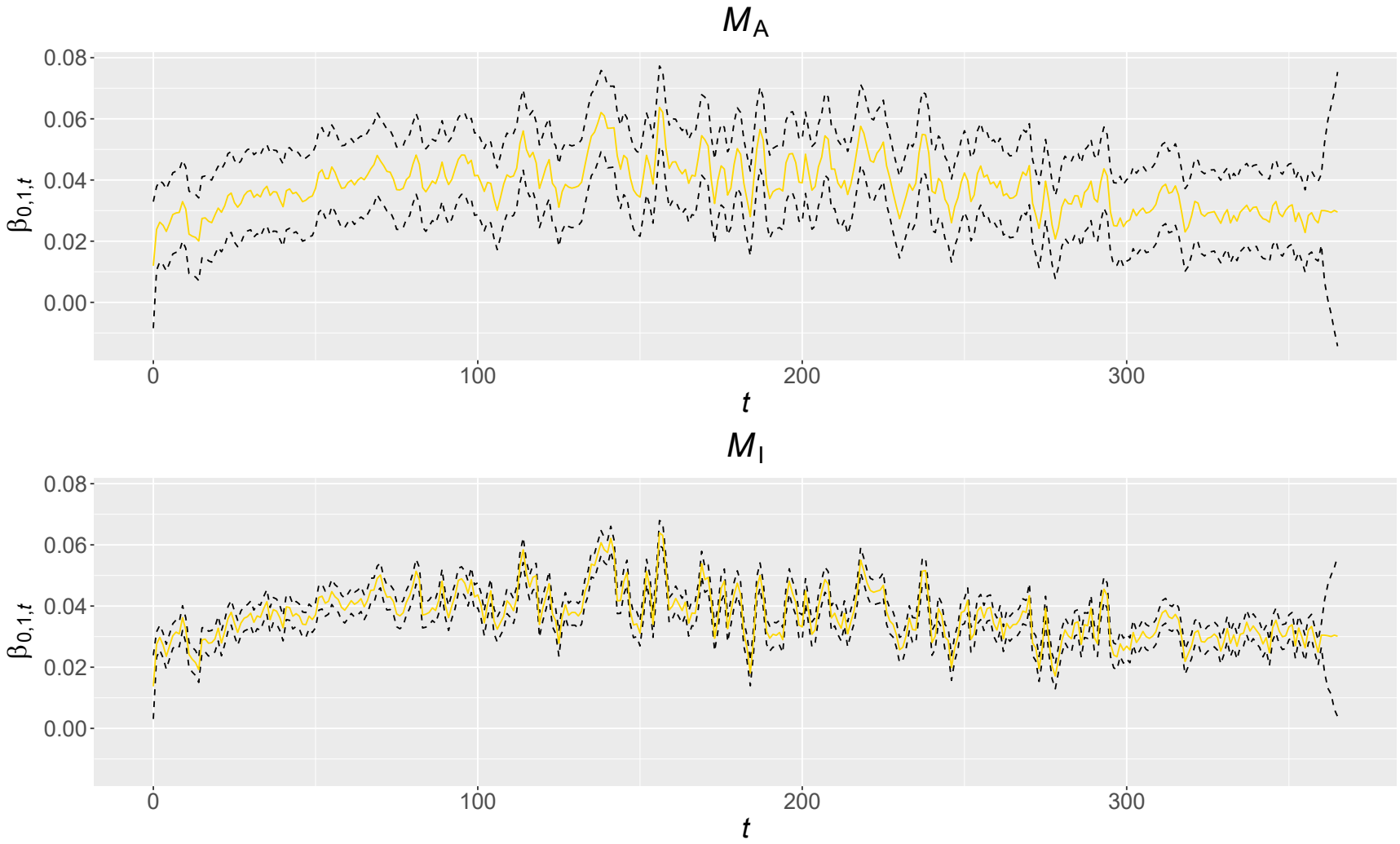


Figure 2.27: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, 360\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

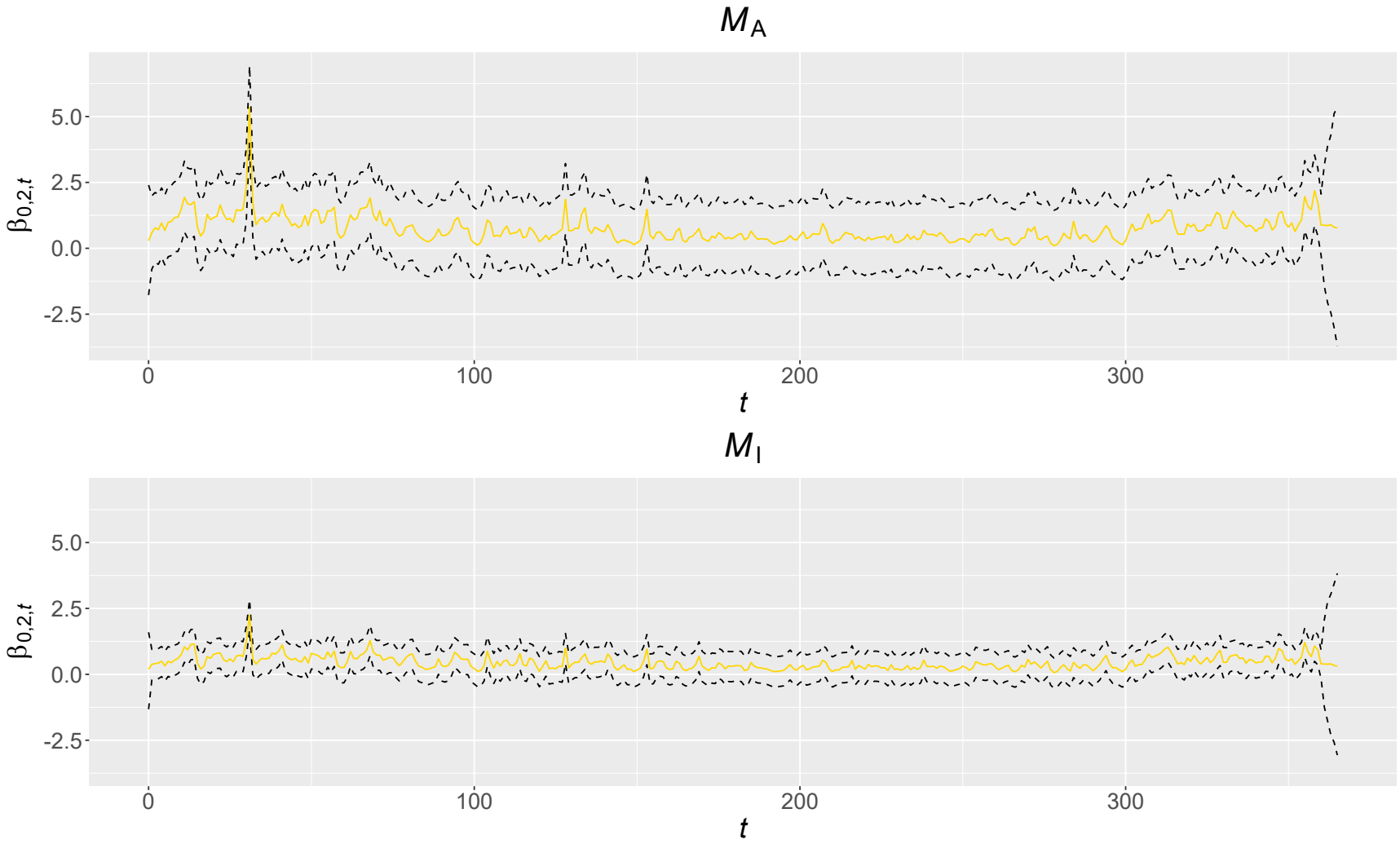


Figure 2.28: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, 360\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

Figure 2.29: Line chart of $Y_{n,i,t}$ for $n = 1$, $i = 1$ and $t \in \{1, \ldots, 365\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). For $t \leq 360$, the solid gold line connects the true values; for $t > 360$, it connects the posterior means.

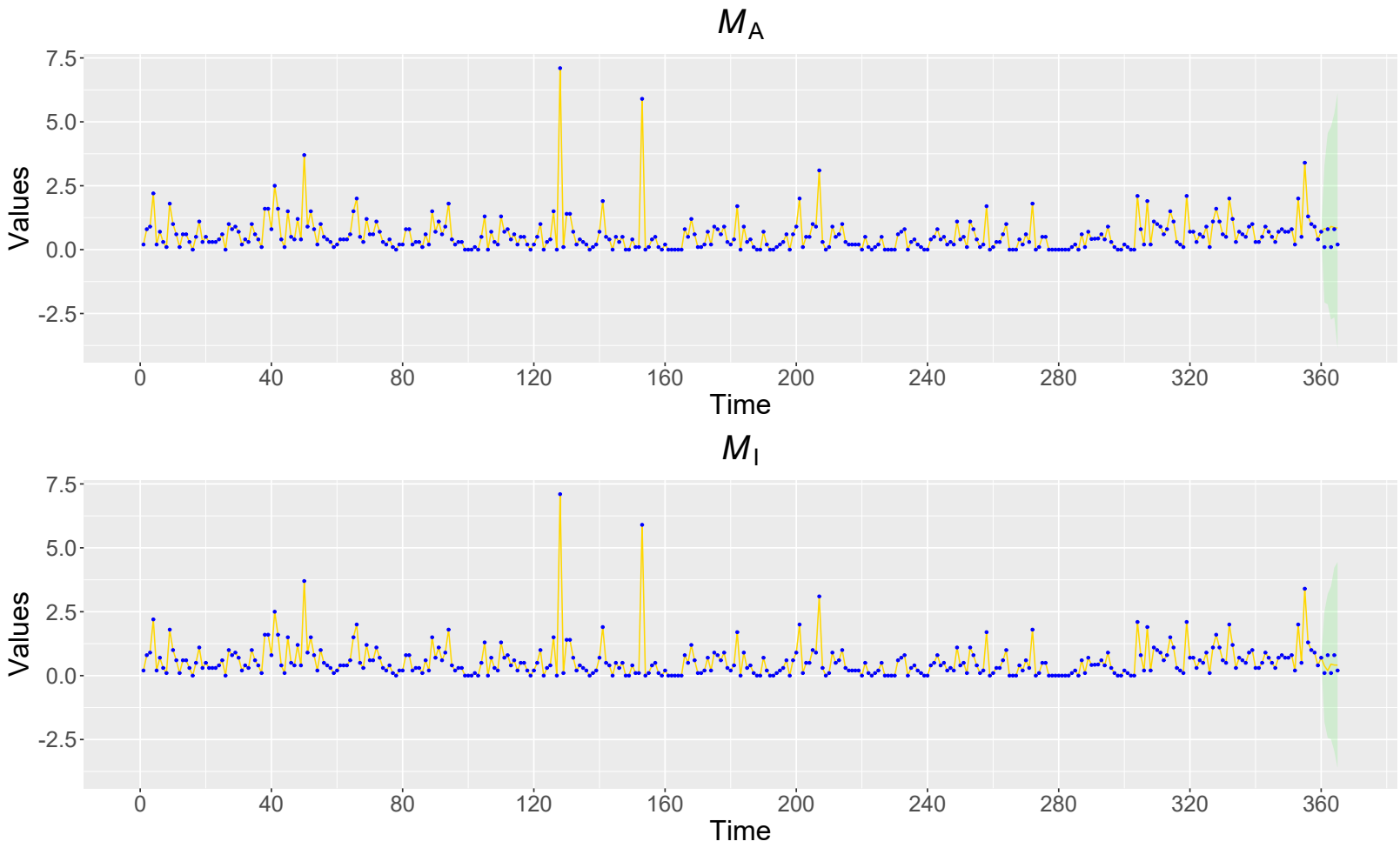


Figure 2.30: Line chart of $Y_{n,i,t}$ for $n = 1$, $i = 2$ and $t \in \{1, \ldots, 365\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). For $t \leq 360$, the solid gold line connects the true values; for $t > 360$, it connects the posterior means.

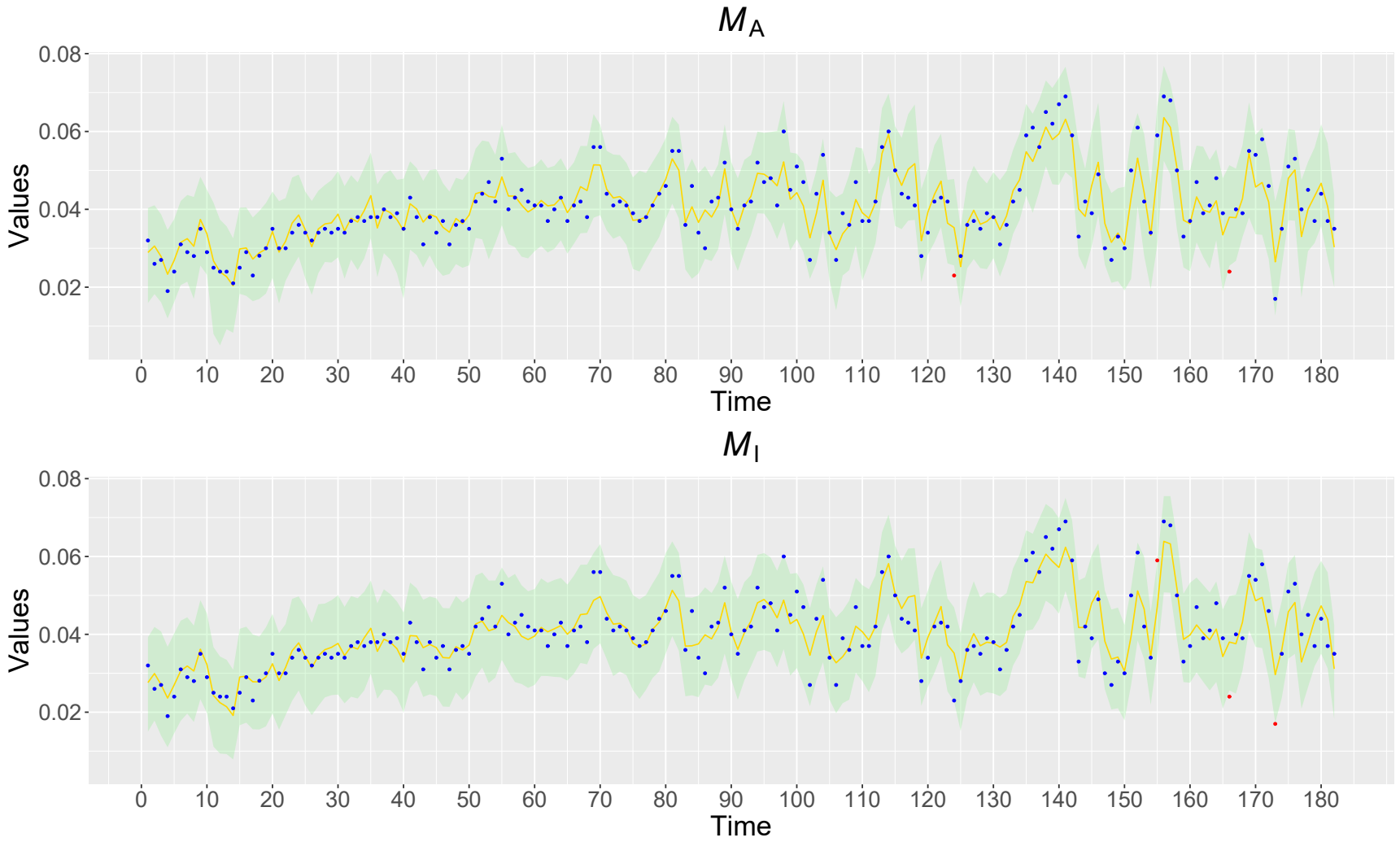


Figure 2.31: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 1$ and $t \in \{1, \ldots, 182\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

Figure 2.32: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 1$ and $t \in \{183, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

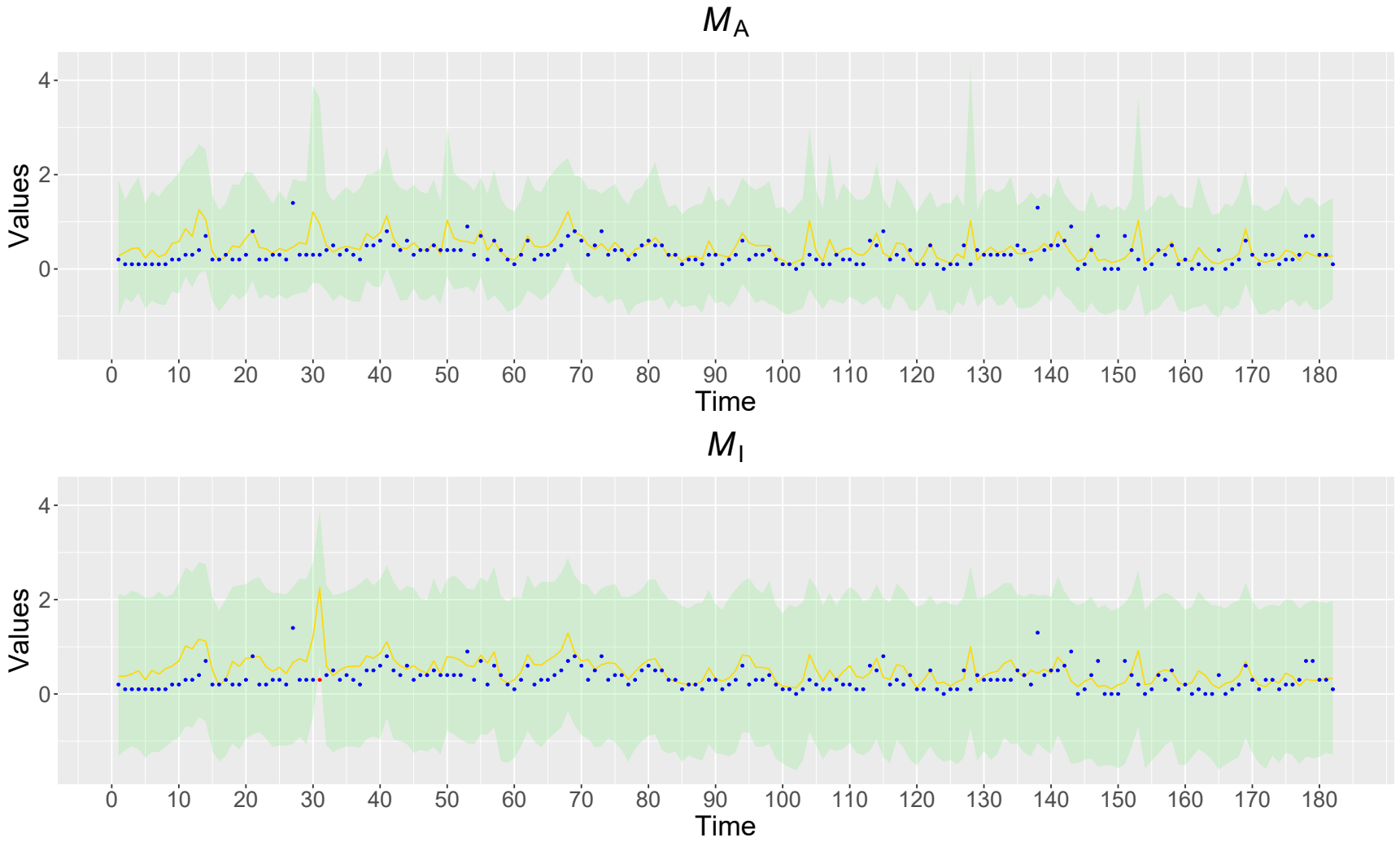


Figure 2.33: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 2$ and $t \in \{1, \ldots, 182\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

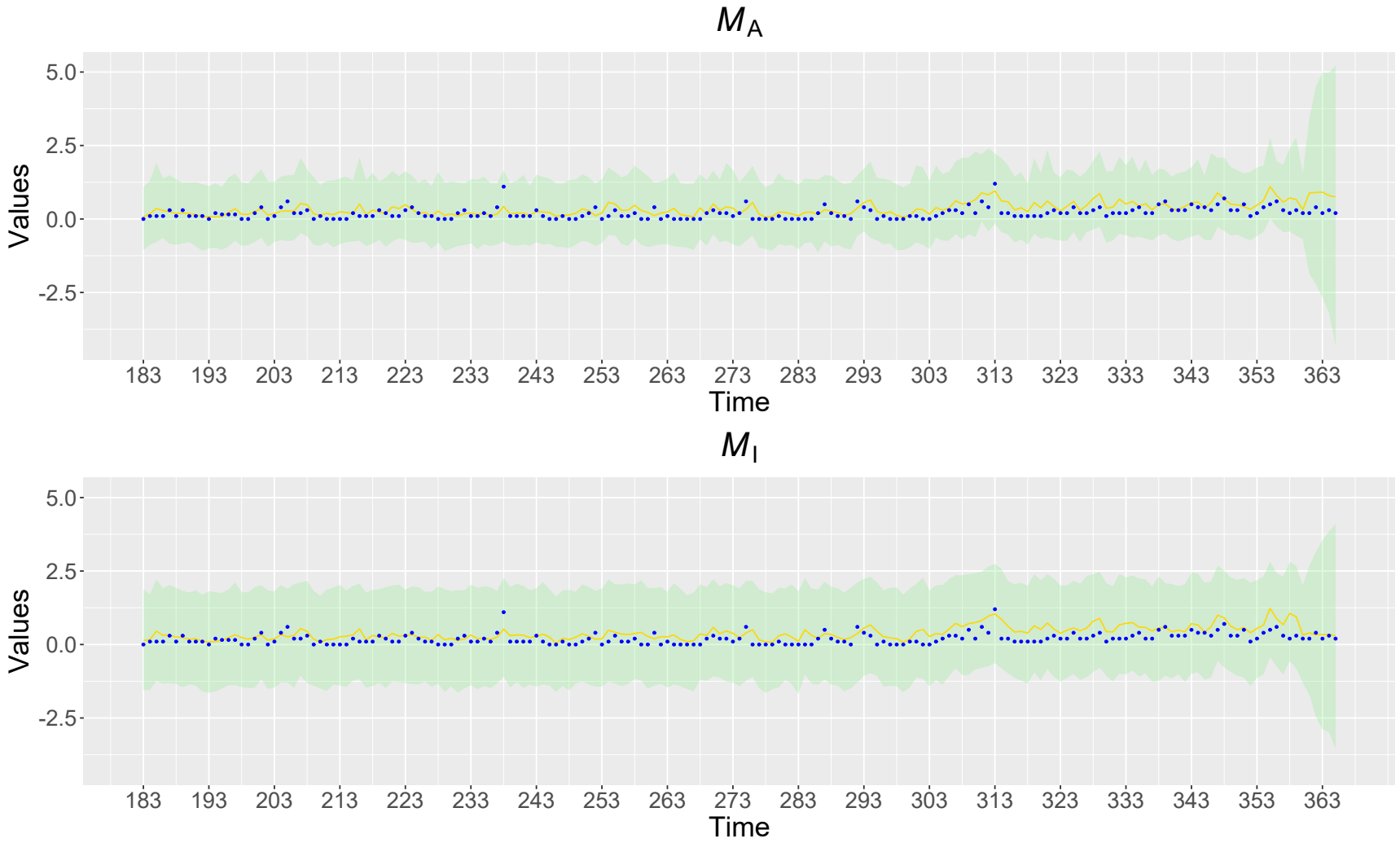


Figure 2.34: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 2$ and $t \in \{183, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 2. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

2.7 Final considerations

In this chapter, we proposed a model for more than one variable measured at monitoring stations in a continuous space at discrete times. It was assumed that all variables share the same spatial dependence structure, which can incorporate anisotropy through spatial deformation. We also consider the hypotheses that spatial dependence does not vary over time and that the variables are normally distributed.

Through simulation and also with an illustrative example, we show that the incorporation of spatial deformation provides better interpolations in some anisotropic scenarios. These results were obtained with the specification of non-informative prior distributions, except for the hyperparameters $\boldsymbol{\sigma}_d^2$ and ψ related to $\mathbf{D}$. There is currently no known ideal value for ψ – there are only a few suggestions in the literature –, but it may be necessary to test different values for this positive quantity. Some choices can greatly reduce the value of the PMSE statistic, resulting in a low ECP as a consequence. In this way, the appropriate fixation of ψ requires different attempts that involve aspects of algorithmic convergence and the joint study of some metrics.

Since several works on spatial deformation under the Bayesian paradigm estimate the coordinates in $\mathcal{D}$-space applying the traditional Metropolis-Hastings algorithm, or with some variation in the method (Damian et al., 2001; Schmidt and O'Hagan, 2003; Damian et al., 2003; Bruno et al., 2009; Schmidt et al., 2011; Morales et al., 2013; Morales and Vicini, 2020; Morales et al., 2022), an important contribution of this work concerns the computational aspects. Here we proposed the implementation of the slice sampler as a useful alternative for the estimation of $\mathbf{D}$, having obtained satisfactory results in terms of fast convergence and with the ease of not needing to define a tuning matrix of variances. In our experience, we found that the burning period of the deformation chains is shorter when using the slice sampler especially for the analysis of real data sets. Table 2.4 shows the processing times recorded in the two simulation studies and in the illustrative example, where one can see the performance of Algorithm 6 (hybrid MCMC) that we proposed. In the first simulation study, it is noted that the time increased tenfold when increasing T from 100 to 1000. In the second simulation study and in the illustrative example, it is verified that the time of the anisotropic model is at least eight times greater than that of the isotropic model. As the longest case took less than 3 hours, we understand that the processing times obtained are reasonable for practical use. However, the use of approximate methods such as variational inference may be necessary when the number of observed time periods (T) and gauged sites (N) are high.

Due to the difficulty of choosing a reasonable value for ψ in certain circumstances, Algorithm

6 can be used with the Metropolis-Hastings algorithm in its sixth step to save a little time during the search. After the choice of ψ is made, we suggest running Algorithm 6 using the slice sampler in its sixth step to obtain less autocorrelated chains.

A limitation of the proposed model concerns the lack of identifiability of the parameters ϕ, V and $\Sigma_{i,i'}$. Although it is possible to estimate the product given by $\phi \cdot V \cdot \Sigma_{i,i'}$ well, this quantity has no intuitive interpretation. For the illustrative example, whose true parameter values are unknown, we present the a posteriori estimates of ϕ and $V \cdot \Sigma_{i,i'}$ separately. Both must be analyzed with the following warning: the values found for ϕ may be overestimated and the values for $V \cdot \Sigma_{i,i'}$ may be underestimated, and vice versa.

The missing values of the response variables of the illustrative example were imputed with the help of an R package. Our next step in Chapter 3 is to develop the proposed model so that it is capable of making imputations via MCMC methods.

Table 2.4: Processing time, given in minutes, recorded in the simulation studies and in the illustrative example in Chapter 2 (Sections 2.5 and 2.6), by model ($\mathcal{M}_{\text{A}}$ or $\mathcal{M}_{\text{I}}$), number of observed time periods (T), number of gauged sites (N), number of response variables (q), number of regression coefficients per time (p) and number of iterations (J). Algorithm 6 was run using the slice sampling (SS) or Metropolis-Hastings (MH) to sample $\mathbf{D}$ in its 6^{th} step.

Description	**Model**	T	N	q	p	J	**Time (min.)**
First simulation study	$\mathcal{M}_{\text{A}}$ w/ SS	10	17	2	2	11000	8.04
First simulation study	$\mathcal{M}_{\text{A}}$ w/ SS	100	17	2	2	11000	23.63
First simulation study	$\mathcal{M}_{\text{A}}$ w/ MH	100	17	2	2	11000	5.41
First simulation study	$\mathcal{M}_{\text{A}}$ w/ SS	1000	17	2	2	11000	104.14
Second simulation study	$\mathcal{M}_{\text{A}}$ w/ SS	100	16	2	2	20000	24.87
Second simulation study	$\mathcal{M}_{\text{I}}$	100	16	2	2	20000	2.79
Illustrative example	$\mathcal{M}_{\text{A}}$ w/ SS	360	10	2	1	120000	172.88
Illustrative example	$\mathcal{M}_{\text{I}}$	360	10	2	1	120000	23.55

The algorithms were run on a computer with the following configuration: (i) Processor: Intel® Core™ i7-6500U CPU @ 2.50 GHz; (ii) Installed RAM: 16 GB.

Chapter 3

Bayesian modelling of incomplete response matrices

3.1 Introduction

Multivariate response measurements taken at various sites and times are rarely complete. In Chapter 2 we present an illustrative example in which we imputed some missing values through an R package. Our objective for this chapter is to perform the imputation of missing values simultaneously with the estimation of the unknown model parameters.

Guttman and Menzefricke (1983) presented a way to make Bayesian inference in multivariate regression with missing observations on the response variable, considering a pattern in which the response matrix can be written as a matrix block consisting of three sub-matrices of observed values and a sub-matrix of missing values. Corradi and Guagnano (1993) presented a method for estimating missing data in the common components dynamic linear model, having observations described by vectors whose entries represent values of different response variables that were measured at a fixed time. Triantafyllopoulos (2008) worked with the problem of imputation of missing values in matrix-normal Bayesian dynamic model, considering that some (or all) columns of the response matrices are formed only by missing values and that the empty columns of the response matrices are not the same for all times, but with the limitation of not having shown applications of the proposed methodology in real datasets. In this chapter, we will work with a scenario that considers that the total of missing values in the response matrices vary in each row and column.

To analyze four pollutants with missing values dispersed across distinct sites and time periods, Gamerman et al. (2022) assumed that the missing-data mechanism is completely random and

applied the spatiotemporal model developed in their work to estimate the missing data by borrowing information from different variables. In this work, we consider the same assumption of random missing-data generation mechanism of Gamerman et al. (2022).

Since the distribution of the response variable is part of the model, it is possible to predict the missing values by computing their predictive distribution (Gómez-Rubio, 2020). Working with spatial deformation, Morales (2010, Sec. 2.3) present the predictive distribution of missing values to a spatiotemporal model for multivariate responses.

To handle missing values in the matrix-variate case, we will vectorize each $N \times q$ response matrix to obtain response vectors of length Nq. After that, these Nq entries will be partitioned into sub-vectors of empty and non-empty components, as done by Morales (2010, Sec. 2.3) in the case $q = 1$. The main methodological distinction with respect to the cited work is that we will impute the missing data while performing the inference procedure on the model parameters, instead of imputing them through the predictive density.

Inspired by Berliner (1996, Sec. 1.2), we will use a casual way of representing some quantities to show the reader the differences between Chapters 2 and 3. In Chapter 2, we are interested in obtaining $[\texttt{Parameters} \mid \texttt{Complete data}]$ in a context where there are no missing values or where existing missing values are imputed before modelling. In Chapter 3, we will consider the decomposition $[\texttt{Complete data}] = [\texttt{Observed data}, \texttt{Missing data}]$ and, therefore, we became interested in obtaining $[\texttt{Parameters}, \texttt{Missing data} \mid \texttt{Observed data}]$. Since the data augmentation method can be viewed as iterative multiple imputation (Gelman et al., 2013, Sec. 18.2) and this procedure is a particular case of the Gibbs sampler, the following update cycle can be intuitively formulated to sample from this joint density:

- Sample from $[\texttt{Missing data} \mid \texttt{Parameters}, \texttt{Observed data}]$;
- Sample from $[\texttt{Parameters} \mid \texttt{Missing data}, \texttt{Observed data}] \sim [\texttt{Parameters} \mid \texttt{Complete data}]$.

For predicting unobserved data (e.g. future data to forecast and data from an ungauged site to interpolate), one can think about this through the following informal representation:

$$\begin{aligned}[\texttt{Unobs. data}, \texttt{Mis. data}, \texttt{Parameters} \mid \texttt{Obs. data}] &= [\texttt{Unobs. data} \mid \texttt{Parameters}, \texttt{Cpl. data}] \\ &\times [\texttt{Parameters}, \texttt{Mis. data} \mid \texttt{Obs. data}].\end{aligned}$$

This chapter is structured as follows. Section 3.2 presents a way to adapt the statistical model presented in Equation (2.3) to model matrix-variate responses with missing values. Section 3.3

presents strategies to make forecasting and interpolation. In Section 3.4 we present some criteria for comparing models. Simulation studies are discussed in Section 3.5. An illustrative example is presented in Section 3.6. Lastly, in Section 3.7 we present some considerations.

3.2 Formulation

3.2.1 Modelling incomplete response matrices

Consider a $N \times q$ response matrix $\mathbf{Y}_t$ with N_t^{mis} empty entries and N_t^{obs} observed non-empty entries such that $N_t^{\mathsf{obs}} + N_t^{\mathsf{mis}} = Nq$ for any time $t \in \mathcal{T} = \{1, \ldots, T\}$, where Nq is the number of entries in $\mathbf{Y}_t$. In general, we have three cases for the t^{th} response matrix:

1. **No Missing Values (NMV):** $N_t^{\mathsf{obs}} = Nq$ and $N_t^{\mathsf{mis}} = 0$ (i.e. $\mathbf{Y}_t$ is fully available at time t: all q response variables were measured at all N sites).

2. **Some Missing Values (SMV):** $N_t^{\mathsf{obs}}, N_t^{\mathsf{mis}} \in \{1, \ldots, Nq - 1\}$ (i.e. $\mathbf{Y}_t$ is partially available at time t: there is at least one measurement in some of the N sites, but not all, for some of the q response variables).

3. **Only Missing Values (OMV):** $N_t^{\mathsf{obs}} = 0$ and $N_t^{\mathsf{mis}} = Nq$ (i.e. $\mathbf{Y}_t$ is not available at time t: none of the q response variables were measured at any of the N sites).

The set of times is of the form $\mathcal{T} = \{1, \ldots, T\} = \mathcal{T}^{\mathsf{NMV}} \cup \mathcal{T}^{\mathsf{SMV}} \cup \mathcal{T}^{\mathsf{OMV}}$, where $\mathcal{T}^{\mathsf{NMV}}$, $\mathcal{T}^{\mathsf{SMV}}$ and $\mathcal{T}^{\mathsf{OMV}}$ are three disjoint subsets of indices corresponding respectively to No Missing Values, Some Missing Values and Only Missing Values, such that $\#\mathcal{T}^{\mathsf{NMV}} + \#\mathcal{T}^{\mathsf{SMV}} + \#\mathcal{T}^{\mathsf{OMV}} = T$.

In many applications, it is not adequate to assume that the number of missing values in each column of the response matrix is the same. To illustrate this, recall the notation introduced in Section 2.2 and then consider the following example:

$$\mathbf{Y}_t = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{1,t} & \underset{\sim}{\mathbf{Y}}_{2,t} \end{bmatrix} = \begin{bmatrix} Y_{1,1,t} & \texttt{NA} \\ Y_{2,1,t} & \texttt{NA} \\ \texttt{NA} & Y_{3,2,t} \\ Y_{4,1,t} & Y_{4,2,t} \end{bmatrix} = \begin{bmatrix} Y_{1,1,t}^{\mathsf{obs}} & Y_{1,2,t}^{\mathsf{mis}} \\ Y_{2,1,t}^{\mathsf{obs}} & Y_{2,2,t}^{\mathsf{mis}} \\ Y_{3,1,t}^{\mathsf{mis}} & Y_{3,2,t}^{\mathsf{obs}} \\ Y_{4,1,t}^{\mathsf{obs}} & Y_{4,2,t}^{\mathsf{obs}} \end{bmatrix}.$$

In cases like this ($N = 4$, $q = 2$, $N_t^{\mathsf{obs}} = 5$ and $N_t^{\mathsf{mis}} = 3$), the response matrix cannot be partitioned into blocks. However, one may vectorize the response matrix by doing

$$\underset{\sim}{\mathbf{Y}}_t = \operatorname{vec} \mathbf{Y}_t = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{1,t} \\ \underset{\sim}{\mathbf{Y}}_{2,t} \end{bmatrix} = \begin{bmatrix} Y_{1,1,t}^{\mathsf{obs}} & Y_{2,1,t}^{\mathsf{obs}} & Y_{3,1,t}^{\mathsf{mis}} & Y_{4,1,t}^{\mathsf{obs}} & Y_{1,2,t}^{\mathsf{mis}} & Y_{2,2,t}^{\mathsf{mis}} & Y_{3,2,t}^{\mathsf{obs}} & Y_{4,2,t}^{\mathsf{obs}} \end{bmatrix}^{\mathsf{T}}$$

and, to split $\underset{\sim}{\mathbf{Y}}_t$ into two sub-vectors, say $\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}}$ and $\underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}}$, one may multiply $\underset{\sim}{\mathbf{Y}}_t$ from the left by a 8×8 permutation matrix $\mathbf{P}_t$ as follows:

$$\mathbf{P}_t \cdot \underset{\sim}{\mathbf{Y}}_t = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y^{\mathsf{obs}}_{1,1,t} \\ Y^{\mathsf{obs}}_{2,1,t} \\ Y^{\mathsf{mis}}_{3,1,t} \\ Y^{\mathsf{obs}}_{4,1,t} \\ Y^{\mathsf{mis}}_{1,2,t} \\ Y^{\mathsf{mis}}_{2,2,t} \\ Y^{\mathsf{obs}}_{3,2,t} \\ Y^{\mathsf{obs}}_{4,2,t} \end{bmatrix} = \begin{bmatrix} Y^{\mathsf{obs}}_{1,1,t} \\ Y^{\mathsf{obs}}_{2,1,t} \\ Y^{\mathsf{obs}}_{4,1,t} \\ Y^{\mathsf{obs}}_{3,2,t} \\ Y^{\mathsf{obs}}_{4,2,t} \\ Y^{\mathsf{mis}}_{3,1,t} \\ Y^{\mathsf{mis}}_{1,2,t} \\ Y^{\mathsf{mis}}_{2,2,t} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}^{\mathsf{obs}}_{1,t} \\ \underset{\sim}{\mathbf{Y}}^{\mathsf{obs}}_{2,t} \\ \underset{\sim}{\mathbf{Y}}^{\mathsf{mis}}_{1,t} \\ \underset{\sim}{\mathbf{Y}}^{\mathsf{mis}}_{2,t} \end{bmatrix} = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} \\ \underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} \end{bmatrix},$$

where

$$\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}^{\mathsf{obs}}_{1,t} \\ \underset{\sim}{\mathbf{Y}}^{\mathsf{obs}}_{2,t} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} = \begin{bmatrix} \underset{\sim}{\mathbf{Y}}^{\mathsf{mis}}_{1,t} \\ \underset{\sim}{\mathbf{Y}}^{\mathsf{mis}}_{2,t} \end{bmatrix}$$

have the following sub-vectors:

$$\underset{\sim}{\mathbf{Y}}^{\mathsf{obs}}_{1,t} = \begin{bmatrix} Y^{\mathsf{obs}}_{1,1,t} \\ Y^{\mathsf{obs}}_{2,1,t} \\ Y^{\mathsf{obs}}_{4,1,t} \end{bmatrix}, \quad \underset{\sim}{\mathbf{Y}}^{\mathsf{obs}}_{2,t} = \begin{bmatrix} Y^{\mathsf{obs}}_{3,2,t} \\ Y^{\mathsf{obs}}_{4,2,t} \end{bmatrix}, \quad \underset{\sim}{\mathbf{Y}}^{\mathsf{mis}}_{1,t} = \begin{bmatrix} Y^{\mathsf{mis}}_{3,1,t} \end{bmatrix} \quad \text{and} \quad \underset{\sim}{\mathbf{Y}}^{\mathsf{mis}}_{2,t} = \begin{bmatrix} Y^{\mathsf{mis}}_{1,2,t} \\ Y^{\mathsf{mis}}_{2,2,t} \end{bmatrix}.$$

Note that we are usually interested in working with the observed set $\underset{\sim}{\mathbf{Y}}_{\mathsf{obs}} = \{\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} : 0 < N^{\mathsf{obs}}_t \leq Nq\}$. Now, for $t \in \mathcal{T}^{\mathsf{SMV}}$, one may obtain the t^{th} vector of observations by multiplying the permuted vector from the left by a 5×8 block matrix $\mathbf{L}_{t,\mathsf{obs}} = [\, \mathbf{I}_{N^{\mathsf{obs}}_t} \; \mathbf{0}_{N^{\mathsf{obs}}_t \times N^{\mathsf{mis}}_t} \,] = \begin{bmatrix} \mathbf{I}_5 & \mathbf{0}_{5\times 3} \end{bmatrix}$ as follows:

$$\mathbf{L}_{t,\mathsf{obs}} \cdot \begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} \\ \underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} Y^{\mathsf{obs}}_{1,1,t} \\ Y^{\mathsf{obs}}_{2,1,t} \\ Y^{\mathsf{obs}}_{4,1,t} \\ Y^{\mathsf{obs}}_{3,2,t} \\ Y^{\mathsf{obs}}_{4,2,t} \\ Y^{\mathsf{mis}}_{3,1,t} \\ Y^{\mathsf{mis}}_{1,2,t} \\ Y^{\mathsf{mis}}_{2,2,t} \end{bmatrix} = \begin{bmatrix} Y^{\mathsf{obs}}_{1,1,t} \\ Y^{\mathsf{obs}}_{2,1,t} \\ Y^{\mathsf{obs}}_{4,1,t} \\ Y^{\mathsf{obs}}_{3,2,t} \\ Y^{\mathsf{obs}}_{4,2,t} \end{bmatrix} = \underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}}.$$

Due to the need to transform matrices into vectors to deal with missing values, we will use the results of the vector representation of the Equation (2.3). Based on Quintana (1987, Sec. 3.3.2),

this anisotropic model ($\mathcal{M}_{\mathrm{A}}$) is rewritten as follows:

$$\begin{aligned}
\underset{\sim}{\mathbf{Y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} &\sim \mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}]),\ t \in \{1, \ldots, T\}, \\
\mathbf{B} &= \left[\ B_{n,n'} \ \right]_{N \times N}, \\
B_{n,n'} &= \begin{cases} \exp\{-\phi \| d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'}) \|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases}, \\
\operatorname{vec} \boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{pq}([\mathbf{I}_q \otimes \mathbf{G}_t] \operatorname{vec} \boldsymbol{\beta}_{t-1}, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{W}]),\ t \in \{1, \ldots, T\}, \\
\operatorname{vec} \boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{pq}(\operatorname{vec} \mathbf{M}_0, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{C}_0]), \\
V &\sim \mathsf{IG}(a_V, b_V), \\
\boldsymbol{\Sigma} &\sim \mathsf{IW}_q(a_{\boldsymbol{\Sigma}}, \mathbf{b}_{\boldsymbol{\Sigma}}), \\
\phi &\sim \mathsf{G}(a_\phi, b_\phi), \\
\mathbf{D} &\sim \mathsf{N}_{2 \times N}(\mathbf{S}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d).
\end{aligned} \tag{3.1}$$

Note that $\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} = \underset{\sim}{\mathbf{Y}}_t$, $\mathbf{P}_t = \mathbf{I}_{Nq}$ and $\mathbf{L}_{t,\mathsf{obs}} = \mathbf{I}_{Nq}$ if $t \in \mathcal{T}^{\mathsf{NMV}}$. If $t \in \mathcal{T}^{\mathsf{SMV}}$, from Equation (3.1) we know that

$$\underset{\sim}{\mathbf{Y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}])$$

and, using some properties of the multivariate normal distribution (Press, 2005, Secs. 3.2 and 3.4), the random vector $\begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} \\ \underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} \end{bmatrix} = \mathbf{P}_t \cdot \underset{\sim}{\mathbf{Y}}_t$ is such that

$$(\mathbf{P}_t \cdot \underset{\sim}{\mathbf{Y}}_t) \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{Nq}(\underset{\sim}{\boldsymbol{\mu}}_t, V \cdot \boldsymbol{\Delta}_t)$$

or, equivalently,

$$\begin{bmatrix} \underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} \\ \underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} \end{bmatrix} \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N_t^{\mathsf{obs}} + N_t^{\mathsf{mis}}} \left(\begin{bmatrix} \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}} \\ \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}} \end{bmatrix}, V \cdot \begin{bmatrix} \boldsymbol{\Delta}_{t,\mathsf{obs}} & \boldsymbol{\Delta}_{t,\mathsf{o,m}} \\ \boldsymbol{\Delta}_{t,\mathsf{m,o}} & \boldsymbol{\Delta}_{t,\mathsf{mis}} \end{bmatrix} \right), \tag{3.2}$$

where, for $\mathbf{L}_{t,\mathsf{obs}} = [\, \mathbf{I}_{N_t^{\mathsf{obs}}} \ \mathbf{0}_{N_t^{\mathsf{obs}} \times N_t^{\mathsf{mis}}} \,]$ and $\mathbf{L}_{t,\mathsf{mis}} = [\, \mathbf{0}_{N_t^{\mathsf{mis}} \times N_t^{\mathsf{obs}}} \ \mathbf{I}_{N_t^{\mathsf{mis}}} \,]$, the quantities

$$\underset{\sim}{\boldsymbol{\mu}}_t = \mathbf{P}_t[\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t = \begin{bmatrix} \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}} \\ \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}} \end{bmatrix} \qquad \text{and} \qquad \boldsymbol{\Delta}_t = \mathbf{P}_t[\boldsymbol{\Sigma} \otimes \mathbf{B}]\mathbf{P}_t^\intercal = \begin{bmatrix} \boldsymbol{\Delta}_{t,\mathsf{obs}} & \boldsymbol{\Delta}_{t,\mathsf{o,m}} \\ \boldsymbol{\Delta}_{t,\mathsf{m,o}} & \boldsymbol{\Delta}_{t,\mathsf{mis}} \end{bmatrix}$$

have the following blocks:

$$\begin{aligned}
\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}} &= \mathbf{L}_{t,\mathsf{obs}} \underset{\sim}{\boldsymbol{\mu}}_t, & \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}} &= \mathbf{L}_{t,\mathsf{mis}} \underset{\sim}{\boldsymbol{\mu}}_t, \\
\boldsymbol{\Delta}_{t,\mathsf{obs}} &= \mathbf{L}_{t,\mathsf{obs}} \boldsymbol{\Delta}_t \mathbf{L}_{t,\mathsf{obs}}^\intercal, & \boldsymbol{\Delta}_{t,\mathsf{o,m}} &= \mathbf{L}_{t,\mathsf{obs}} \boldsymbol{\Delta}_t \mathbf{L}_{t,\mathsf{mis}}^\intercal, \\
\boldsymbol{\Delta}_{t,\mathsf{m,o}} &= \mathbf{L}_{t,\mathsf{mis}} \boldsymbol{\Delta}_t \mathbf{L}_{t,\mathsf{obs}}^\intercal, & \boldsymbol{\Delta}_{t,\mathsf{mis}} &= \mathbf{L}_{t,\mathsf{mis}} \boldsymbol{\Delta}_t \mathbf{L}_{t,\mathsf{mis}}^\intercal.
\end{aligned}$$

In addition to modelling the observed responses, we are interested in estimating the missing responses. Applying the multivariate normal theory (see West and Harrison, 1997, Sec. 17.2.2) in the joint distribution given in Equation (3.2), the conditional distribution of interest is

$$\underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} \mid \underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} = \underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}}, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N_t^{\mathsf{mis}}}(\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}} + \boldsymbol{\Delta}_{t,\mathsf{m,o}}\boldsymbol{\Delta}_{t,\mathsf{obs}}^{-1}[\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} - \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}}], V \cdot [\boldsymbol{\Delta}_{t,\mathsf{mis}} - \boldsymbol{\Delta}_{t,\mathsf{m,o}}\boldsymbol{\Delta}_{t,\mathsf{obs}}^{-1}\boldsymbol{\Delta}_{t,\mathsf{o,m}}])$$

and the marginal distribution of interest is

$$\underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N_t^{\mathsf{mis}}}\left(\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}}, V \cdot \boldsymbol{\Delta}_{t,\mathsf{mis}}\right).$$

Similarly to $\underset{\sim}{\mathbf{Y}}_{\mathsf{obs}}$, define the collection $\underset{\sim}{\mathbf{Y}}_{\mathsf{mis}} = \{\underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} : 0 < N_t^{\mathsf{mis}} \leq Nq\}$. Under the conditional independence assumption, the density of $\underset{\sim}{\mathbf{Y}}_{\mathsf{mis}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{Y}}_{\mathsf{obs}} = \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}$ is given by:

$$f(\underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) = \left[\prod_{t \in \mathcal{T}^{\mathsf{SMV}}} f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}}, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma})\right] \cdot \left[\prod_{t \in \mathcal{T}^{\mathsf{OMV}}} f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}} \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma})\right]. \tag{3.3}$$

One can apply Algorithm 10 to generate a draw from the density $f(\underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$ given in Equation (3.3) through Monte Carlo method. The implementation of this procedure is shown in Appendix B.2.3. Also note that the collection $\underset{\sim}{\mathbf{Y}}_{\mathsf{mis}}$ takes values in the following set:

$$\mathcal{Y}_{\mathsf{mis}} = \prod_{t \in \mathcal{T}^{\mathsf{SMV}} \cup \mathcal{T}^{\mathsf{OMV}}} \mathbb{R}^{N_t^{\mathsf{mis}}} = \left(\prod_{t \in \mathcal{T}^{\mathsf{SMV}}} \mathbb{R}^{N_t^{\mathsf{mis}}}\right) \times \left(\prod_{t \in \mathcal{T}^{\mathsf{OMV}}} \mathbb{R}^{Nq}\right).$$

3.2.2 Inference procedure

Inferences about the parameters of the statistical model given in Equation (3.1) can be made by following steps similar to those described in Section 2.2.3. To prevent the text from becoming repetitive, in Appendix A.6 we present an MCMC algorithm for sampling from the posterior density $f(\boldsymbol{\theta} \mid \underset{\sim}{\mathbf{y}}) \propto f(\boldsymbol{\theta}) l(\boldsymbol{\theta}; \underset{\sim}{\mathbf{y}})$ given in Equation (A.13).

In the context where there are missing values in the response matrices, note that we can write $f(\boldsymbol{\theta} \mid \underset{\sim}{\mathbf{y}}) = f(\boldsymbol{\theta} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}})$ due to exchangeability (see Migon et al., 2014, Sec. 2.4). In fact, for $t \in \mathcal{T}^{\mathsf{SMV}}$ the density of the distribution given in Equation (3.2) can be written as follows:

$$\begin{aligned}
f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}} \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) &= \frac{\exp\left\{-\frac{1}{2}(\mathbf{P}_t\underset{\sim}{\mathbf{y}}_t - \underset{\sim}{\boldsymbol{\mu}}_t)^{\intercal}(V \cdot \boldsymbol{\Delta}_t)^{-1}(\mathbf{P}_t\underset{\sim}{\mathbf{y}}_t - \underset{\sim}{\boldsymbol{\mu}}_t)\right\}}{(2\pi)^{\frac{Nq}{2}}[\det(V \cdot \boldsymbol{\Delta}_t)]^{\frac{1}{2}}} \\
&= \frac{\exp\left\{-\frac{1}{2}[\mathbf{P}_t(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t)]^{\intercal}(V \cdot \mathbf{P}_t[\boldsymbol{\Sigma} \otimes \mathbf{B}]\mathbf{P}_t^{\intercal})^{-1}[\mathbf{P}_t(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t)]\right\}}{(2\pi)^{\frac{Nq}{2}}\{\det\mathbf{P}_t[\det(V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}])]\det\mathbf{P}_t^{\intercal}\}^{\frac{1}{2}}} \\
&= \frac{\exp\left\{-\frac{1}{2}(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t)^{\intercal}(V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}])^{-1}(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t)\right\}}{(2\pi)^{\frac{Nq}{2}}[\det(V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}])]^{\frac{1}{2}}} \\
&= f(\underset{\sim}{\mathbf{y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}).
\end{aligned}$$

Algorithm 10 Monte Carlo method to generate a draw from $f(\underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \boldsymbol{\beta}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$ and imputation of missing values.

Start with $\boldsymbol{\theta}^{(j)} = \left\{V^{(j)}, \boldsymbol{\Sigma}^{(j)}, \boldsymbol{\beta}_0^{(j)}, \boldsymbol{\beta}_1^{(j)}, \ldots, \boldsymbol{\beta}_T^{(j)}, \phi^{(j)}, \mathbf{D}^{(j)}\right\}$, where $j \in \mathbb{N}$.

Compute the matrix $\mathbf{B}_{(j)} = \left[\begin{array}{c} B_{n,n'}^{(j)} \end{array}\right]_{N \times N}$, where $B_{n,n'}^{(j)} = \exp\{-\phi^{(j)} \| d^{(j)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j)}(\underset{\sim}{\mathbf{s}}_{n'}) \|\}$ for all $n, n' \in \{1, \ldots, N\}$.

for $t \leftarrow 1$ to T **do**

  if $N_t^{\mathsf{mis}} = Nq$ **then**

    Sample $\underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j)}$ from the distribution $\mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t^{(j)}, V^{(j)} \cdot [\boldsymbol{\Sigma}^{(j)} \otimes \mathbf{B}_{(j)}])$.

    Do $\underset{\sim}{\mathbf{y}}_t^{(j)} = \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j)}$.

  else

    if $0 < N_t^{\mathsf{mis}} < Nq$ **then**

      Compute the following vectors and matrices:

$$\begin{array}{rclcrcl}
\underset{\sim}{\boldsymbol{\mu}}_t^{(j)} & = & \mathbf{P}_t[\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t^{(j)}, & & \boldsymbol{\Delta}_t^{(j)} & = & \mathbf{P}_t[\boldsymbol{\Sigma}^{(j)} \otimes \mathbf{B}_{(j)}]\mathbf{P}_t^{\mathsf{T}}, \\
\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}}^{(j)} & = & \mathbf{L}_{t,\mathsf{obs}} \underset{\sim}{\boldsymbol{\mu}}_t^{(j)}, & & \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}}^{(j)} & = & \mathbf{L}_{t,\mathsf{mis}} \underset{\sim}{\boldsymbol{\mu}}_t^{(j)}, \\
\boldsymbol{\Delta}_{t,\mathsf{obs}}^{(j)} & = & \mathbf{L}_{t,\mathsf{obs}} \boldsymbol{\Delta}_t^{(j)} \mathbf{L}_{t,\mathsf{obs}}^{\mathsf{T}}, & & \boldsymbol{\Delta}_{t,\mathsf{o,m}}^{(j)} & = & \mathbf{L}_{t,\mathsf{obs}} \boldsymbol{\Delta}_t^{(j)} \mathbf{L}_{t,\mathsf{mis}}^{\mathsf{T}}, \\
\boldsymbol{\Delta}_{t,\mathsf{m,o}}^{(j)} & = & \mathbf{L}_{t,\mathsf{mis}} \boldsymbol{\Delta}_t^{(j)} \mathbf{L}_{t,\mathsf{obs}}^{\mathsf{T}}, & & \boldsymbol{\Delta}_{t,\mathsf{mis}}^{(j)} & = & \mathbf{L}_{t,\mathsf{mis}} \boldsymbol{\Delta}_t^{(j)} \mathbf{L}_{t,\mathsf{mis}}^{\mathsf{T}}.
\end{array}$$

      Sample $\underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j)}$ from the following distribution:

$$\mathsf{N}_{N_t^{\mathsf{mis}}}\left(\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}}^{(j)} + \boldsymbol{\Delta}_{t,\mathsf{m,o}}^{(j)} \left\{\boldsymbol{\Delta}_{t,\mathsf{obs}}^{(j)}\right\}^{-1} \left[\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} - \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}}^{(j)}\right], V^{(j)} \cdot \left[\boldsymbol{\Delta}_{t,\mathsf{mis}}^{(j)} - \boldsymbol{\Delta}_{t,\mathsf{m,o}}^{(j)} \left\{\boldsymbol{\Delta}_{t,\mathsf{obs}}^{(j)}\right\}^{-1} \boldsymbol{\Delta}_{t,\mathsf{o,m}}^{(j)}\right]\right).$$

      Do $\underset{\sim}{\mathbf{y}}_t^{(j)} = \mathbf{P}_t^{-1} \cdot \left[\begin{array}{c} \underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \\ \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j)} \end{array}\right]$.

    else

      Do $\underset{\sim}{\mathbf{y}}_t^{(j)} = \underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}}$.

    end if

  end if

end for

This procedure returns the following collection of vectors to the user:

$$\begin{array}{rcl}
\underset{\sim}{\mathbf{y}}^{(j)} & = & \left\{\underset{\sim}{\mathbf{y}}_t^{(j)} : t \in \{1, \ldots, T\}\right\} \\
& = & \left\{\left\{\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} : t \in \mathcal{T}^{\mathsf{NMV}}\right\}, \left\{\mathbf{P}_t^{-1} \cdot \left[\begin{array}{c} \underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \\ \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j)} \end{array}\right] : t \in \mathcal{T}^{\mathsf{SMV}}\right\}, \left\{\underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j)} : t \in \mathcal{T}^{\mathsf{OMV}}\right\}\right\}.
\end{array}$$

Here we are interested in sampling from $f(\underset{\sim}{\mathbf{y}}_{\mathsf{mis}}, \boldsymbol{\theta} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$, and this can be done via data augmentation method. The procedure is described in Algorithm 11 and its implementation is shown in Appendix B.2.6.

Algorithm 11 Data augmentation method to sample from $f(V, \boldsymbol{\Sigma}, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T, \phi, \mathbf{D}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$.

1: Set initial values for the parameters (i.e. $\boldsymbol{\theta}^{(0)} = \{V^{(0)}, \boldsymbol{\Sigma}^{(0)}, \boldsymbol{\beta}_0^{(0)}, \boldsymbol{\beta}_1^{(0)}, \ldots, \boldsymbol{\beta}_T^{(0)}, \phi^{(0)}, \mathbf{D}^{(0)}\} \in \boldsymbol{\Theta}$) and do $j \leftarrow 1$. ▷ Fix the locations of two sites in $\mathbf{D}^{(0)}$, e.g. pick $\mathbf{D}^{(0)} = \mathbf{S}$.
2: **repeat**
3: Run Algorithm 10 to sample $\underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j)}$ from the density $f(\cdot \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \boldsymbol{\theta}^{(j-1)})$ given in Equation (3.3) and also to obtain the collection $\underset{\sim}{\mathbf{y}}^{(j)} = \{\underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j)}\}$ of T response vectors of length Nq, where $\boldsymbol{\theta}^{(j-1)} = \{V^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}, \boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}\}$. ▷ No need to set the initial value $\underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(0)} \in \mathcal{Y}_{\mathsf{mis}}$.
4: Run Algorithm 19 to sample $\boldsymbol{\theta}^{(j)} = \{V^{(j)}, \boldsymbol{\Sigma}^{(j)}, \boldsymbol{\beta}_0^{(j)}, \boldsymbol{\beta}_1^{(j)}, \ldots, \boldsymbol{\beta}_T^{(j)}, \phi^{(j)}, \mathbf{D}^{(j)}\}$ from the density $f(\cdot \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j)}) = f(\cdot \mid \underset{\sim}{\mathbf{y}}^{(j)})$ given in Equation (A.13).
5: Set $j \leftarrow j + 1$.
6: **until** convergence is reached.

After achieving convergence (say after J iterations), Algorithm 11 is used to store K samples from the posterior distributions of the unknown quantities (say $\boldsymbol{\theta}^{(j_1)}, \ldots, \boldsymbol{\theta}^{(j_K)}$ and $\underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_1)}, \ldots, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_K)}$, where $j_1, \ldots, j_K \in \mathbb{N}$ are indices such that $J + 1 \leq j_1 < \cdots < j_K$). Thinning is used in the code presented in Appendix B.2.6, in order to avoid autocorrelation in the chains.

3.3 Forecasting and interpolation

In this section, we present an adaptation to the posterior predictive densities for forecasting and interpolation presented in Section 2.3, to address the presence of missing values in the response matrices.

3.3.1 Forecasting

Given the observed responses, we wish to predict the response vector at time $T + t$ for all N gauged sites, where T is the last observed time and t is a positive integer.

Using the Markovian structure of the model and supposing that $\mathbf{G}_{T+1}, \mathbf{G}_{T+2}, \ldots, \mathbf{G}_{T+t}$ are t fixed $p \times p$ evolution matrices, we showed in Section 2.3.1 that the conditional distributions

$\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}$ and $\mathbf{Y}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}$ are known. By Definition A.6, Expression (2.12) has the following equivalence:

$$\underset{\sim}{\mathbf{Y}}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_{T+t}] \operatorname{vec} \boldsymbol{\beta}_{T+t}, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}]), \quad t \geq 1.$$

Let $\underset{\sim}{\mathbf{Y}}_{\mathsf{pred}} = \{\underset{\sim}{\mathbf{Y}}_{T+1}, \ldots, \underset{\sim}{\mathbf{Y}}_{T+T^\star}\}$ be the collection of $T^\star$ response vectors to forecast after the last observed time. Now the collection of unknown parameters is augmented to $\boldsymbol{\theta}_{\mathsf{pred}} = \{\boldsymbol{\theta}, \underset{\sim}{\mathbf{Y}}_{\mathsf{mis}}, \boldsymbol{\beta}_{\mathsf{pred}}\}$, where $\boldsymbol{\beta}_{\mathsf{pred}} = \{\boldsymbol{\beta}_{T+1}, \ldots, \boldsymbol{\beta}_{T+T^\star}\}$, with parametric space $\boldsymbol{\Theta}_{\mathsf{pred}} = \boldsymbol{\Theta} \times \mathcal{Y}_{\mathsf{mis}} \times \left(\prod_{t=1}^{T^\star} \mathbb{R}^{p \times q} \right)$. One can forecast these T^* response vectors after the last observed time using the following predictive density:

$$\begin{aligned}
f(\underset{\sim}{\mathbf{y}}_{\mathsf{pred}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) &= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} f(\underset{\sim}{\mathbf{y}}_{\mathsf{pred}}, \boldsymbol{\theta}_{\mathsf{pred}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \partial \boldsymbol{\theta}_{\mathsf{pred}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} \frac{f(\underset{\sim}{\mathbf{y}}_{\mathsf{pred}}, \boldsymbol{\theta}_{\mathsf{pred}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\boldsymbol{\theta}_{\mathsf{pred}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \cdot \frac{f(\boldsymbol{\theta}_{\mathsf{pred}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \partial \boldsymbol{\theta}_{\mathsf{pred}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} f(\underset{\sim}{\mathbf{y}}_{\mathsf{pred}} \mid \boldsymbol{\beta}_{\mathsf{pred}}, \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}) \cdot \frac{f(\boldsymbol{\beta}_{\mathsf{pred}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}) f(\boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \partial \boldsymbol{\theta}_{\mathsf{pred}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{pred}}} \left[\prod_{t=1}^{T^\star} f(\underset{\sim}{\mathbf{y}}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}) \right] f(\boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \partial \boldsymbol{\theta}_{\mathsf{pred}}.
\end{aligned} \tag{3.4}$$

Since the integral that appears in Equation (3.4) is not analytically tractable, we may use Monte Carlo methods to compute it by adding extra steps to Algorithm 11. The procedure is described in Algorithm 12 and its implementation is shown in Appendix B.2.7.1.

Algorithm 12 Monte Carlo integration to approximate $f(\underset{\sim}{\mathbf{y}}_{\mathsf{pred}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$.

Start with K MCMC samples, where $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$ and $\underset{\sim}{\mathbf{y}}^{(j_k)} = \{\underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_k)}\}$ form the $j_k{}^{\text{th}}$ MCMC sample.

for $k \leftarrow 1$ to K **do**
  Compute $\mathbf{B}_{(j_k)} = \left[\ B_{n,n'}^{(j_k)}\ \right]_{N\times N}$, where $B_{n,n'}^{(j_k)} = \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{n'})\|\}$ for all $n, n' \in \{1, \ldots, N\}$.
  for $t \leftarrow 1$ to $T^\star$ **do**
    if $t \leftarrow 1$ **or** $T^\star \leftarrow 1$ **then**
      Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{p\times q}(\mathbf{G}_{T+1}\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \mathbf{W}, \boldsymbol{\Sigma}^{(j_k)})$.
    else
      Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the following distribution:

$$\mathsf{N}_{p\times q}\left(\left[\prod_{t'=t}^{1}\mathbf{G}_{T+t'}\right]\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \left\{\mathbf{W} + \sum_{t'=t}^{2}\left[\prod_{t''=t}^{t'}\mathbf{G}_{T+t''}\right]^{\mathsf{T}}\mathbf{W}\left[\prod_{t''=t}^{t'}\mathbf{G}_{T+t''}\right]\right\}, \boldsymbol{\Sigma}^{(j_k)}\right).$$

    end if
    Sample $\underset{\sim}{\mathbf{y}}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_{T+t}]\operatorname{vec}\boldsymbol{\beta}_{T+t}^{(j_k)}, V^{(j_k)} \cdot [\boldsymbol{\Sigma}^{(j_k)} \otimes \mathbf{B}_{(j_k)}])$.
  end for
end for

3.3.2 Interpolation

3.3.2.1 Interpolation for observed times

Given the observed responses, we wish to predict the response vector at N^* ungauged sites of the region of interest $\mathcal{S}$ for observed times.

The conditional distributions $\mathbf{D}^* \mid \mathbf{D}$ and $\mathbf{Y}_t^* \mid \mathbf{Y}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}$ are known, as discussed in Section 2.3.2.1. By Definition A.6, Expression (2.15) has the following equivalence:

$$\underset{\sim}{\mathbf{Y}}_t^* \mid \underset{\sim}{\mathbf{Y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{N^*q}(\underset{\sim}{\boldsymbol{\mu}}_{\underset{\sim}{\mathbf{Y}}_t^\star \mid \underset{\sim}{\mathbf{Y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}}, V \cdot \boldsymbol{\Delta}_{\underset{\sim}{\mathbf{Y}}_t^\star \mid \underset{\sim}{\mathbf{Y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}}),$$

for some $t \in \{1, \ldots, T\}$, where

$$\begin{aligned}
\underset{\sim}{\boldsymbol{\mu}}_{\underset{\sim}{\mathbf{Y}}_t^\star \mid \underset{\sim}{\mathbf{Y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}} &= \mathrm{E}[\underset{\sim}{\mathbf{Y}}_t^\star \mid \underset{\sim}{\mathbf{Y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}] \\
&= [\mathbf{I}_q \otimes \mathbf{X}_t^*]\operatorname{vec}\boldsymbol{\beta}_t + \mathbf{B}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}(\underset{\sim}{\mathbf{Y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t)
\end{aligned}$$

and

$$\boldsymbol{\Delta}_{\underset{\sim}{\mathbf{Y}}_t^\star|\underset{\sim}{\mathbf{Y}}_t,\boldsymbol{\beta}_t,V,\phi,\mathbf{D},\mathbf{D}^*,\boldsymbol{\Sigma}} = \text{Cov}[\underset{\sim}{\mathbf{Y}}_t^\star \mid \underset{\sim}{\mathbf{Y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}] = \boldsymbol{\Sigma} \otimes (\mathbf{B}^* - \mathbf{B}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}\mathbf{B}_{\mathsf{g},\mathsf{u}}).$$

Now the collection of unknown parameters is augmented to $\boldsymbol{\theta}_{\mathsf{int}} = \{\boldsymbol{\theta}, \underset{\sim}{\mathbf{Y}}_{\mathsf{mis}}, \mathbf{D}^*\}$, with parametric space $\boldsymbol{\Theta}_{\mathsf{int}} = \boldsymbol{\Theta} \times \mathcal{Y}_{\mathsf{mis}} \times \mathbb{R}^{2\times N^*}$. Define $\underset{\sim}{\mathbf{Y}}_{\mathsf{int}} = \{\underset{\sim}{\mathbf{Y}}_1^*, \ldots, \underset{\sim}{\mathbf{Y}}_T^*\}$. For observed times, one can interpolate the response vectors using the following predictive density:

$$\begin{aligned}
f(\underset{\sim}{\mathbf{y}}_{\mathsf{int}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) &= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} f(\underset{\sim}{\mathbf{y}}_{\mathsf{int}}, \boldsymbol{\theta}_{\mathsf{int}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \partial\boldsymbol{\theta}_{\mathsf{int}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} \frac{f(\underset{\sim}{\mathbf{y}}_{\mathsf{int}}, \boldsymbol{\theta}_{\mathsf{int}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\boldsymbol{\theta}_{\mathsf{int}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \cdot \frac{f(\boldsymbol{\theta}_{\mathsf{int}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \partial\boldsymbol{\theta}_{\mathsf{int}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} f(\underset{\sim}{\mathbf{y}}_{\mathsf{int}} \mid \mathbf{D}^*, \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \cdot \frac{f(\mathbf{D}^* \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) f(\boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \partial\boldsymbol{\theta}_{\mathsf{int}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{int}}} \left[\prod_{t=1}^{T} f(\underset{\sim}{\mathbf{y}}_t^* \mid \underset{\sim}{\mathbf{y}}_t, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}) \right] f(\mathbf{D}^* \mid \mathbf{D}) f(\boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \partial\boldsymbol{\theta}_{\mathsf{int}}. \qquad (3.5)
\end{aligned}$$

Since the integral that appears in Equation (3.5) is not analytically tractable, we may use Monte Carlo methods to compute it by adding extra steps to Algorithm 11. The procedure is described in Algorithm 13 and implemented in Appendix B.2.7.2.

Algorithm 13 Monte Carlo integration to approximate $f(\underset{\sim}{\mathbf{y}}_{\mathsf{int}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$.

Start with K MCMC samples and also with the hyperparameters $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$ and ψ, where $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$ and $\underset{\sim}{\mathbf{y}}^{(j_k)} = \{\underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_k)}\}$ form the k^{th} MCMC sample.

1: **for** $k \leftarrow 1$ to K **do**

2: Sample $\mathbf{D}^*_{(j_k)}$ from the distribution $\mathsf{N}_{2\times N^*}(\mathbf{S}^* + (\mathbf{D}^{(j_k)} - \mathbf{S})\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^* - \mathbf{R}_{\mathsf{u},\mathsf{g}}\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}})$.

3: Compute the matrices $\mathbf{B}_{(j_k)} = \left[\begin{array}{c} B_{n,n'}^{(j_k)} \end{array}\right]_{N\times N}$, $\mathbf{B}_{\mathsf{g},\mathsf{u}}^{(j_k)} = \left[\begin{array}{c} B_{n,n'}^{\mathsf{g},\mathsf{u}^{(j_k)}} \end{array}\right]_{N\times N^*}$ ($\mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}$ is its transpose) and $\mathbf{B}^*_{(j_k)} = \left[\begin{array}{c} B_{n,n'}^{*(j_k)} \end{array}\right]_{N^*\times N^*}$, where:

$$\begin{aligned}
B_{n,n'}^{(j_k)} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{n'})\|\}, && n, n' \in \{1,\ldots,N\},\\
B_{n,n'}^{\mathsf{g},\mathsf{u}^{(j_k)}} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}, && n \in \{1,\ldots,N\}, n' \in \{1,\ldots,N^*\},\\
B_{n,n'}^{*(j_k)} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n}) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}, && n, n' \in \{1,\ldots,N^*\}.
\end{aligned}$$

4: **for** $t \leftarrow 1$ to T **do**

5: Sample $\underset{\sim}{\mathbf{y}}_t^{*(j_k)}$ from the following distribution:

$$\mathsf{N}_{N^*q}([\mathbf{I}_q \otimes \mathbf{X}_t^*]\operatorname{vec}\boldsymbol{\beta}_t^{(j_k)} + \mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}\mathbf{B}_{(j_k)}^{-1}(\underset{\sim}{\mathbf{y}}_t^{(j_k)} - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t^{(j_k)}), V^{(j_k)} \cdot [\boldsymbol{\Sigma}^{(j_k)} \otimes (\mathbf{B}^*_{(j_k)} - \mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}\mathbf{B}_{(j_k)}^{-1}\mathbf{B}_{\mathsf{g},\mathsf{u}}^{(j_k)})]).$$

6: **end for**

7: **end for**

3.3.2.2 Interpolation after last observed time

Given the observed responses, we wish to predict the response vector at N^* ungauged sites of the region of interest $\mathcal{S}$ for future times.

As discussed in Sections 2.3.2.2 and 3.3.1, the following conditional distributions are known: $\mathbf{D}^* \mid \mathbf{D}$; $\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}$; $\underset{\sim}{\mathbf{Y}}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}$; and $\mathbf{Y}^*_{T+t} \mid \mathbf{Y}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}$. By Definition A.6, Expression (2.17) has the following equivalence:

$$\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma} \sim \mathsf{N}_{N^*q}(\underset{\sim}{\boldsymbol{\mu}}_{\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}}, V \cdot \boldsymbol{\Delta}_{\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}})$$

for all $t \geq 1$, where

$$\begin{aligned}
\underset{\sim}{\boldsymbol{\mu}}_{\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}} &= \mathrm{E}[\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}]\\
&= [\mathbf{I}_q \otimes \mathbf{X}^*_{T+t}]\operatorname{vec}\boldsymbol{\beta}_{T+t} + \mathbf{B}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}(\underset{\sim}{\mathbf{Y}}_{T+t} - [\mathbf{I}_q \otimes \mathbf{X}_{T+t}]\operatorname{vec}\boldsymbol{\beta}_{T+t})
\end{aligned}$$

and

$$\boldsymbol{\Delta}_{\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}} = \mathrm{Cov}[\underset{\sim}{\mathbf{Y}}^*_{T+t} \mid \underset{\sim}{\mathbf{Y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}] = \boldsymbol{\Sigma} \otimes (\mathbf{B}^* - \mathbf{B}_{\mathsf{u},\mathsf{g}}\mathbf{B}^{-1}\mathbf{B}_{\mathsf{g},\mathsf{u}}).$$

Let $\underset{\sim}{\mathbf{Y}}_{\mathsf{aug}} = \{\underset{\sim}{\mathbf{Y}}^*_{T+1}, \ldots, \underset{\sim}{\mathbf{Y}}^*_{T+T^\star}\}$ be the collection of $T^\star$ response vectors to forecast after the last observed time in N^* ungauged sites of interest. Now the collection of unknown parameters is augmented to $\boldsymbol{\theta}_{\mathsf{aug}} = \{\boldsymbol{\theta}, \underset{\sim}{\mathbf{Y}}_{\mathsf{mis}}, \mathbf{D}^*, \boldsymbol{\beta}_{\mathsf{pred}}, \underset{\sim}{\mathbf{Y}}_{\mathsf{pred}}\}$, where $\mathbf{D}^* = \begin{bmatrix} \mathbf{D}_{N+1} & \cdots & \mathbf{D}_{N+N^*} \end{bmatrix}$, $\boldsymbol{\beta}_{\mathsf{pred}} = \{\boldsymbol{\beta}_{T+1}, \ldots, \boldsymbol{\beta}_{T+T^\star}\}$ and $\underset{\sim}{\mathbf{Y}}_{\mathsf{pred}} = \{\underset{\sim}{\mathbf{Y}}_{T+1}, \ldots, \underset{\sim}{\mathbf{Y}}_{T+T^\star}\}$ is treated as a collection of unknown quantities, with parametric space $\boldsymbol{\Theta}_{\mathsf{aug}} = \boldsymbol{\Theta} \times \mathcal{Y}_{\mathsf{mis}} \times \mathbb{R}^{2\times N^*} \times \left(\prod_{t=1}^{T^\star} \mathbb{R}^{p\times q}\right) \times \left(\prod_{t=1}^{T^\star} \mathbb{R}^{Nq}\right)$. One can forecast and interpolate responses simultaneously using the following predictive density:

$$
\begin{aligned}
f(\underset{\sim}{\mathbf{y}}_{\mathsf{aug}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) &= \int_{\boldsymbol{\Theta}_{\mathsf{aug}}} f(\underset{\sim}{\mathbf{y}}_{\mathsf{aug}}, \boldsymbol{\theta}_{\mathsf{aug}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \partial\boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{aug}}} \frac{f(\underset{\sim}{\mathbf{y}}_{\mathsf{aug}}, \boldsymbol{\theta}_{\mathsf{aug}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\boldsymbol{\theta}_{\mathsf{aug}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \cdot \frac{f(\boldsymbol{\theta}_{\mathsf{aug}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})}{f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}})} \partial\boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{aug}}} f(\underset{\sim}{\mathbf{y}}_{\mathsf{aug}} \mid \boldsymbol{\theta}_{\mathsf{aug}}, \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) f(\boldsymbol{\theta}_{\mathsf{aug}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \partial\boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{aug}}} \Big\{ f(\underset{\sim}{\mathbf{y}}_{\mathsf{aug}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{pred}}, \boldsymbol{\beta}_{\mathsf{pred}}, \mathbf{D}^*, \boldsymbol{\theta}) f(\underset{\sim}{\mathbf{y}}_{\mathsf{pred}} \mid \boldsymbol{\beta}_{\mathsf{pred}}, \boldsymbol{\theta}) \\
&\qquad f(\boldsymbol{\beta}_{\mathsf{pred}} \mid \boldsymbol{\theta}) f(\mathbf{D}^* \mid \mathbf{D}) f(\boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \Big\} \partial\boldsymbol{\theta}_{\mathsf{aug}} \\
&= \int_{\boldsymbol{\Theta}_{\mathsf{aug}}} \Big\{ \Big[\prod_{t=1}^{T^\star} f(\underset{\sim}{\mathbf{y}}^*_{T+t} \mid \underset{\sim}{\mathbf{y}}_{T+t}, \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \mathbf{D}^*, \boldsymbol{\Sigma}) f(\underset{\sim}{\mathbf{y}}_{T+t} \mid \boldsymbol{\beta}_{T+t}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\
&\qquad f(\boldsymbol{\beta}_{T+t} \mid \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}) \Big] f(\mathbf{D}^* \mid \mathbf{D}) f(\boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}) \Big\} \partial\boldsymbol{\theta}_{\mathsf{aug}}. \qquad (3.6)
\end{aligned}
$$

Since the integral that appears in Equation (3.6) is not analytically tractable, we may use Monte Carlo methods to compute it by adding extra steps to Algorithm 11. The procedure is described in Algorithm 14 and implemented in Appendix B.2.7.2.

Algorithm 14 Monte Carlo integration to approximate $f(\underset{\sim}{\mathbf{y}}_{\mathsf{aug}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{obs}})$.

Start with K MCMC samples and also with the hyperparameters $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$ and ψ, where $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$ and $\underset{\sim}{\mathbf{y}}^{(j_k)} = \{\underset{\sim}{\mathbf{y}}_{\mathsf{obs}}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_k)}\}$ form the k^{th} MCMC sample.

1: **for** $k \leftarrow 1$ to K **do**

2: Sample $\mathbf{D}^*_{(j_k)}$ from the distribution $\mathsf{N}_{2\times N^*}(\mathbf{S}^* + (\mathbf{D}^{(j_k)} - \mathbf{S})\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^* - \mathbf{R}_{\mathsf{u},\mathsf{g}}\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}})$.

3: Compute the matrices $\mathbf{B}_{(j_k)} = \left[\begin{array}{c} B_{n,n'}^{(j_k)} \end{array}\right]_{N\times N}$, $\mathbf{B}_{\mathsf{g},\mathsf{u}}^{(j_k)} = \left[\begin{array}{c} B_{n,n'}^{\mathsf{g},\mathsf{u}^{(j_k)}} \end{array}\right]_{N\times N^*}$ ($\mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}$ is its transpose) and $\mathbf{B}^*_{(j_k)} = \left[\begin{array}{c} B^{*(j_k)}_{n,n'} \end{array}\right]_{N^*\times N^*}$, where:

$$
\begin{aligned}
B_{n,n'}^{(j_k)} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{n'})\|\}, && n, n' \in \{1,\ldots,N\},\\
B_{n,n'}^{\mathsf{g},\mathsf{u}^{(j_k)}} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}, && n \in \{1,\ldots,N\}, n' \in \{1,\ldots,N^*\},\\
B_{n,n'}^{*(j_k)} &= \exp\{-\phi^{(j_k)}\|d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n}) - d^{(j_k)}(\underset{\sim}{\mathbf{s}}_{N+n'})\|\}, && n, n' \in \{1,\ldots,N^*\}.
\end{aligned}
$$

4: **for** $t \leftarrow 1$ to $T^\star$ **do**

5: **if** $t \leftarrow 1$ **or** $T^\star \leftarrow 1$ **then**

6: Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{p\times q}(\mathbf{G}_{T+1}\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \mathbf{W}, \boldsymbol{\Sigma}^{(j_k)})$.

7: **else**

8: Sample $\boldsymbol{\beta}_{T+t}^{(j_k)}$ from the following distribution:

$$
\mathsf{N}_{p\times q}\left(\left[\prod_{t'=t}^{1}\mathbf{G}_{T+t'}\right]\boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)} \cdot \left\{\mathbf{W} + \sum_{t'=t}^{2}\left[\prod_{t''=t}^{t'}\mathbf{G}_{T+t''}\right]^{\mathsf{T}}\mathbf{W}\left[\prod_{t''=t}^{t'}\mathbf{G}_{T+t''}\right]\right\}, \boldsymbol{\Sigma}^{(j_k)}\right).
$$

9: **end if**

10: Sample $\underset{\sim}{\mathbf{y}}_{T+t}^{(j_k)}$ from the distribution $\mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_{T+t}]\operatorname{vec}\boldsymbol{\beta}_{T+t}^{(j_k)}, V^{(j_k)} \cdot [\boldsymbol{\Sigma}^{(j_k)} \otimes \mathbf{B}_{(j_k)}])$.

11: Sample $\underset{\sim}{\mathbf{y}}_{T+t}^{*(j_k)}$ from the following distribution:

$$
\mathsf{N}_{N^*q}([\mathbf{I}_q \otimes \mathbf{X}^*_{T+t}]\operatorname{vec}\boldsymbol{\beta}_{T+t}^{(j_k)} + \mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}\mathbf{B}_{(j_k)}^{-1}(\underset{\sim}{\mathbf{y}}_{T+t}^{(j_k)} - [\mathbf{I}_q \otimes \mathbf{X}_{T+t}]\operatorname{vec}\boldsymbol{\beta}_{T+t}^{(j_k)}), V^{(j_k)} \cdot [\boldsymbol{\Sigma}^{(j_k)} \otimes (\mathbf{B}^*_{(j_k)} - \mathbf{B}_{\mathsf{u},\mathsf{g}}^{(j_k)}\mathbf{B}_{(j_k)}^{-1}\mathbf{B}_{\mathsf{g},\mathsf{u}}^{(j_k)})]).
$$

12: **end for**

13: **end for**

3.4 Model comparison

In this work, we are interested in analyzing spatiotemporal data by comparing the predictive performance of the anisotropic model ($\mathcal{M}_{\text{A}}$) given in Equation (3.1) with that of its analogous

isotropic model ($\mathcal{M}_{\mathrm{I}}$), which is described as follows:

$$\begin{aligned}
\underset{\sim}{\mathbf{Y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} &\sim \mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}]),\ t \in \{1, \ldots, T\}, \\
\mathbf{B} &= \left[\ B_{n,n'}\ \right]_{N \times N}, \\
\operatorname{vec} \boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{pq}([\mathbf{I}_q \otimes \mathbf{G}_t] \operatorname{vec} \boldsymbol{\beta}_{t-1}, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{W}]),\ t \in \{1, \ldots, T\}, \\
\operatorname{vec} \boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{pq}(\operatorname{vec} \mathbf{M}_0, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{C}_0]), \\
V &\sim \mathsf{IG}(a_V, b_V), \\
\boldsymbol{\Sigma} &\sim \mathsf{IW}_q(a_{\boldsymbol{\Sigma}}, \mathbf{b}_{\boldsymbol{\Sigma}}), \\
\phi &\sim \mathsf{G}(a_\phi, b_\phi).
\end{aligned} \tag{3.7}$$

The entries of $\mathbf{B}$ are the form:

$$B_{n,n'} = \begin{cases} \exp\{-\phi \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases} \quad \text{or} \quad B_{n,n'} = \begin{cases} \exp\{-\frac{1}{\phi} \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases}.$$

Our first choice for the isotropic model is $B_{n,n'} = \exp\{-\phi \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_n\|\}$, as it has a parameterization that is closer to that of the anisotropic model $\mathcal{M}_{\mathrm{A}}$. In this thesis, we will use the specification $B_{n,n'} = \exp\{-\frac{1}{\phi} \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_n\|\}$ only when it is not possible to use $B_{n,n'} = \exp\{-\phi \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_n\|\}$ due to convergence problems in Algorithm 11. Texts such as Kitanidis (1997, Sec. 3.4.1.2), Diggle and Ribeiro-Jr. (2007, Sec. 3.4.1) and Zhu and Zhang (2013) also work with isotropic exponential covariance functions written in the form $V \cdot \exp\{-\frac{1}{\phi} \|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}$. According to Definition A.15, the quantities ϕ and $1/\phi$ are respectively known as *decay* and *range*.

Some metrics for model comparison were presented in Section 2.4, namely: the Deviance Information Criterion (DIC), the Predictive Mean Squared Error (PMSE), the Empirical Coverage Probability (ECP) and the Interval Score (IS). The DIC statistic is useful for comparing different models fitted to the same data set. The PMSE statistic evaluates how close or far the posterior means for the interpolated values are from the true values. The ECP statistic gives the fraction of true values that are in the intervals between the 2.5^{th} and 97.5^{th} percentiles of the predictive distributions of the interpolated values, while the IS statistic takes into account the lengths of the intervals and whether or not they cover the true values.

The goal is to use DIC, PMSE, ECP and IS statistics to compare the anisotropic model ($\mathcal{M}_{\mathrm{A}}$) with its analogous isotropic model ($\mathcal{M}_{\mathrm{I}}$). However, Expressions (2.20), (2.21), (2.22) and (2.23) cannot be applied in the context where there are missing values in the response matrices, requiring some slight modifications.

To adapt the DIC statistic, some distributional results are needed. Under the conditional

independence assumption, the density of $\underset{\sim}{\mathbf{Y}}_{\mathsf{obs}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{Y}}_{\mathsf{mis}} = \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}$ is given by the following expression:

$$f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}) = \left[\prod_{t \in \mathcal{T}^{\mathsf{NMV}}} f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \right] \cdot \left[\prod_{t \in \mathcal{T}^{\mathsf{SMV}}} f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \mid \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \right], \tag{3.8}$$

where, applying the multivariate normal theory (see West and Harrison, 1997, Sec. 17.2.2) in the joint distribution given in Equation (3.2), the conditional distribution of interest is

$$\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} \mid \underset{\sim}{\mathbf{Y}}_{t,\mathsf{mis}} = \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}, \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N_t^{\mathsf{obs}}}(\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}} + \boldsymbol{\Delta}_{t,\mathsf{o,m}} \boldsymbol{\Delta}_{t,\mathsf{mis}}^{-1} [\underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}} - \underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{mis}}], V \cdot [\boldsymbol{\Delta}_{t,\mathsf{obs}} - \boldsymbol{\Delta}_{t,\mathsf{o,m}} \boldsymbol{\Delta}_{t,\mathsf{mis}}^{-1} \boldsymbol{\Delta}_{t,\mathsf{m,o}}])$$

and the marginal distribution of interest is

$$\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{N}_{N_t^{\mathsf{obs}}} \left(\underset{\sim}{\boldsymbol{\mu}}_{t,\mathsf{obs}}, V \cdot \boldsymbol{\Delta}_{t,\mathsf{obs}} \right).$$

Let $\boldsymbol{\theta}^{(j_k)} = \left\{ \boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)} \right\}$ be the $j_k{}^{\text{th}}$ MCMC sample of $\boldsymbol{\theta}$, where $k \in \{1, \ldots, K\}$, K is size of the MCMC chains and $\{j_1, \ldots, j_K\}$ is a set of indices. Define $\bar{\boldsymbol{\theta}} = \{\bar{\boldsymbol{\beta}}_0, \bar{\boldsymbol{\beta}}_1, \ldots, \bar{\boldsymbol{\beta}}_T, \bar{V}, \bar{\phi}, \bar{\mathbf{D}}, \bar{\boldsymbol{\Sigma}}\}$, where $\bar{\boldsymbol{\beta}}_t = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\beta}_t^{(j_k)}$ for $t \in \{0, 1, \ldots, T\}$, $\bar{V} = \frac{1}{K} \sum_{k=1}^{K} V^{(j_k)}$, $\bar{\phi} = \frac{1}{K} \sum_{k=1}^{K} \phi^{(j_k)}$, $\bar{\mathbf{D}} = \frac{1}{K} \sum_{k=1}^{K} \mathbf{D}^{(j_k)}$ and $\bar{\boldsymbol{\Sigma}} = \frac{1}{K} \sum_{k=1}^{K} \boldsymbol{\Sigma}^{(j_k)}$. Similarly, write $\underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_k)} = \left\{ \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j_k)} : 0 < N_t^{\mathsf{mis}} \leq Nq \right\}$ and $\bar{\underset{\sim}{\mathbf{y}}}_{\mathsf{mis}} = \left\{ \bar{\underset{\sim}{\mathbf{y}}}_{t,\mathsf{mis}} = \frac{1}{K} \sum_{k=1}^{K} \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j_k)} : 0 < N_t^{\mathsf{mis}} \leq Nq \right\}$. The DIC statistic is given by:

$$\begin{aligned}
\text{DIC} &= 2\,\mathrm{E}_{\underset{\sim}{\mathbf{Y}}_{\mathsf{mis}}, \boldsymbol{\theta} \mid \underset{\sim}{\mathbf{Y}}_{\mathsf{obs}} = \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}} [-2 \ln f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}} \mid \boldsymbol{\theta}, \underset{\sim}{\mathbf{y}}_{\mathsf{mis}})] - \{-2 \ln f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}} \mid \mathrm{E}[\boldsymbol{\theta}, \underset{\sim}{\mathbf{Y}}_{\mathsf{mis}} \mid \underset{\sim}{\mathbf{Y}}_{\mathsf{obs}} = \underset{\sim}{\mathbf{y}}_{\mathsf{obs}}])\} \\
&\approx -4 \cdot \left\{ \frac{1}{K} \sum_{k=1}^{K} \ln f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}} \mid \underset{\sim}{\mathbf{y}}_{\mathsf{mis}}^{(j_k)}, \boldsymbol{\theta}^{(j_k)}) \right\} + 2 \ln f(\underset{\sim}{\mathbf{y}}_{\mathsf{obs}} \mid \bar{\boldsymbol{\theta}}, \bar{\underset{\sim}{\mathbf{y}}}_{\mathsf{mis}}) \\
&= -\frac{4}{K} \sum_{k=1}^{K} \left\{ \sum_{t \in \mathcal{T}^{\mathsf{NMV}}} \ln f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \mid \boldsymbol{\beta}_t^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}) + \right. \\
&\qquad \left. + \sum_{t \in \mathcal{T}^{\mathsf{SMV}}} \ln f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \mid \underset{\sim}{\mathbf{y}}_{t,\mathsf{mis}}^{(j_k)}, \boldsymbol{\beta}_t^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}) \right\} + \\
&+ 2 \left\{ \sum_{t \in \mathcal{T}^{\mathsf{NMV}}} \ln f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \mid \bar{\boldsymbol{\beta}}_t, \bar{V}, \bar{\phi}, \bar{\mathbf{D}}, \bar{\boldsymbol{\Sigma}}) + \sum_{t \in \mathcal{T}^{\mathsf{SMV}}} \ln f(\underset{\sim}{\mathbf{y}}_{t,\mathsf{obs}} \mid \bar{\underset{\sim}{\mathbf{y}}}_{t,\mathsf{mis}}, \bar{\boldsymbol{\beta}}_t, \bar{V}, \bar{\phi}, \bar{\mathbf{D}}, \bar{\boldsymbol{\Sigma}}) \right\}.
\end{aligned}$$

Based on Gelman et al. (2013, Sec. 18.1), we define the *inclusion indicator* as follows:

$$O_{n,i,t} = \begin{cases} 1, & \text{if } Y_{n,i,t} \text{ is an observed value } (Y_{n,i,t} = Y_{n,i,t}^{\mathsf{obs}}) \\ 0, & \text{if } Y_{n,i,t} \text{ is a missing value } (Y_{n,i,t} = Y_{n,i,t}^{\mathsf{mis}} = \mathtt{NA}) \end{cases}. \tag{3.9}$$

The expressions for the PMSE, ECP and average IS metrics are given by

$$\text{PMSE} = \frac{\sum_{n=1}^{N^*} \sum_{i=1}^{q} \sum_{t=1}^{T} O_{N+n,i,t} \cdot (\bar{Y}_{N+n,i,t} - Y_{N+n,i,t})^2}{\sum_{n=1}^{N^*} \sum_{i=1}^{q} \sum_{t=1}^{T} O_{N+n,i,t}},$$

$$\text{ECP}_{n,i}^{[\alpha]} = \frac{\sum_{t=1}^{T} O_{N+n,i,t} \cdot \mathbb{1}_{\left[\hat{Y}_{N+n,i,t}^{[\alpha/2]}, \hat{Y}_{N+n,i,t}^{[1-\alpha/2]}\right]}(Y_{N+n,i,t})}{\sum_{t=1}^{T} O_{N+n,i,t}}$$

and

$$\text{IS}_{N+n,i}^{[\alpha]} = \frac{\sum_{t=1}^{T} O_{N+n,i,t} \cdot \text{IS}_{N+n,i,t}^{[\alpha]}}{\sum_{t=1}^{T} O_{N+n,i,t}}.$$

3.5 Simulation studies

3.5.1 Simulation 1

In this simulation study, we want to evaluate whether Algorithm 11 is able to correctly retrieve the parameters considering different amounts of missing values in each response variable. For each response vector $\underset{\sim}{\mathbf{Y}}_{i,t}$, we will insert missing values following one of the three following schemes:

- **Case 1.** Choose a number uniformly and randomly from the set $\{1, \ldots, N\}$, say n_1, and let $Y_{n_1,i,t} = \texttt{NA}$;
- **Case 2.** Choose two numbers uniformly and randomly from the set $\{1, \ldots, N\}$, without replacement, say n_1 and n_2, and let $Y_{n_1,i,t} = Y_{n_2,i,t} = \texttt{NA}$;
- **Case 3.** Choose four numbers uniformly and randomly from the set $\{1, \ldots, N\}$, without replacement, say n_1, n_2, n_3, n_4, and let $Y_{n_1,i,t} = Y_{n_2,i,t} = Y_{n_3,i,t} = Y_{n_4,i,t} = \texttt{NA}$.

We consider $q = 2$ response variables, $T = 200$ times, and $p = 2$ regression coefficients per time and response variable. We simulate the process Y at $N = 16$ points in the unit square (i.e. $\mathcal{S} = [0, 1]^2$). Figure 3.1 shows the geographic region of interest and highlights which points are used as anchor points ($\underset{\sim}{\mathbf{s}}_1$ and $\underset{\sim}{\mathbf{s}}_2$) and the other non-anchor points ($\underset{\sim}{\mathbf{s}}_3, \ldots, \underset{\sim}{\mathbf{s}}_{16}$). These seventeen points are used to fit the model. We generate data from the following anisotropic scheme:

1. Generate $\mathbf{D} = \begin{bmatrix} \mathbf{D}_{1:2} & \mathbf{D}_{3:16} \end{bmatrix}_{2\times 16}$ proceeding as follows:
   (a) Do $\mathbf{D}_{1:2} = \mathbf{S}_{1:2}$ (i.e. $d(\underset{\sim}{\mathbf{s}}_1) = \underset{\sim}{\mathbf{s}}_1$ and $d(\underset{\sim}{\mathbf{s}}_2) = \underset{\sim}{\mathbf{s}}_2$) to fix the locations of two sites in $\mathcal{D}$-space, where $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{1:2} & \mathbf{S}_{3:16} \end{bmatrix}_{2\times 16}$.
   (b) Apply Equation (2.14) to obtain $\mathbf{D}_{3:16} \mid \mathbf{D}_{1:2} = \mathbf{S}_{1:2}$ and generate $\mathbf{D}_{3:16}$ from the distribution $\mathsf{N}_{2\times 14}(\mathbf{S}_{3:16}, \text{diag}\{0.200, 0.200\}, \mathbf{R}_{3:16} - \mathbf{R}_{3:16,1:2}\mathbf{R}_{1:2}^{-1}\mathbf{R}_{1:2,3:16})$, where $\mathbf{R}_{1:2}$ (2×2),

$\mathbf{R}_{1:2,3:16}$ (2×14), $\mathbf{R}_{3:16,1:2}$ (14×2) and $\mathbf{R}_{3:16}$ (14×14) are blocks of the following matrix:

$$\mathbf{R}_d = \left[\ \exp\{-\tilde{\psi}\cdot\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2\}\ \right]_{16\times 16} = \begin{bmatrix} \mathbf{R}_{1:2} & \mathbf{R}_{1:2,3:16} \\ \mathbf{R}_{3:16,1:2} & \mathbf{R}_{3:16} \end{bmatrix}.$$

Here $\tilde{\psi} = 1/\Big(2\cdot \max_{n,n'\in\{1,\ldots,16\}}\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2\Big) = 25/36 \simeq 0.694$.

2. Set $V = 0.1$, $\phi = 0.3$ and $\mathbf{\Sigma} = \left[\begin{smallmatrix} 1.00 & 0.75 \\ 0.75 & 1.00 \end{smallmatrix}\right]$ as true parameters, generate $\boldsymbol{\beta}_0$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{0}_{2\times 2}, V\cdot\mathbf{I}_2, \mathbf{\Sigma})$ and compute $\mathbf{B} = \left[\ B_{n,n'}\ \right]_{16\times 16}$, where, for all $n, n' \in \{1, 2, \ldots, 16\}$, we have:

$$B_{n,n'} = \begin{cases} \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'})\|\}, & n \neq n' \\ 1, & n = n' \end{cases}.$$

3. For $t = 1, \ldots, 200$ do the following:

   (a) Generate $\boldsymbol{\beta}_t$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{G}_t\boldsymbol{\beta}_{t-1}, V\cdot\mathbf{W}, \mathbf{\Sigma})$, where $\mathbf{G}_t = \mathbf{I}_2$ and $\mathbf{W} = \mathbf{I}_2$.

   (b) Generate $\mathbf{X}_t = \left[\begin{smallmatrix} 1 & U_{1,t} \\ 1 & U_{2,t} \\ \vdots & \vdots \\ 1 & U_{16,t} \end{smallmatrix}\right]^{\intercal}$ (16×2), where $U_{1,t}, \ldots, U_{16,t}$ are sixteen random numbers between zero and one.

   (c) Generate $\mathbf{Y}_t$ from the distribution $\mathsf{N}_{16\times 2}(\mathbf{X}_t\boldsymbol{\beta}_t, V\cdot\mathbf{B}, \mathbf{\Sigma})$.

   (d) Insert missing values in each column of $\mathbf{Y}_t$, following Case 1, 2 or 3.

   (e) Write the vector $\underset{\sim}{\mathbf{Y}}_t = \text{vec}(\mathbf{Y}_t)$ and obtain the sub-vector $\underset{\sim}{\mathbf{Y}}_{t,\text{obs}} = \mathbf{L}_{t,\text{obs}}\mathbf{P}_t\underset{\sim}{\mathbf{Y}}_t$.

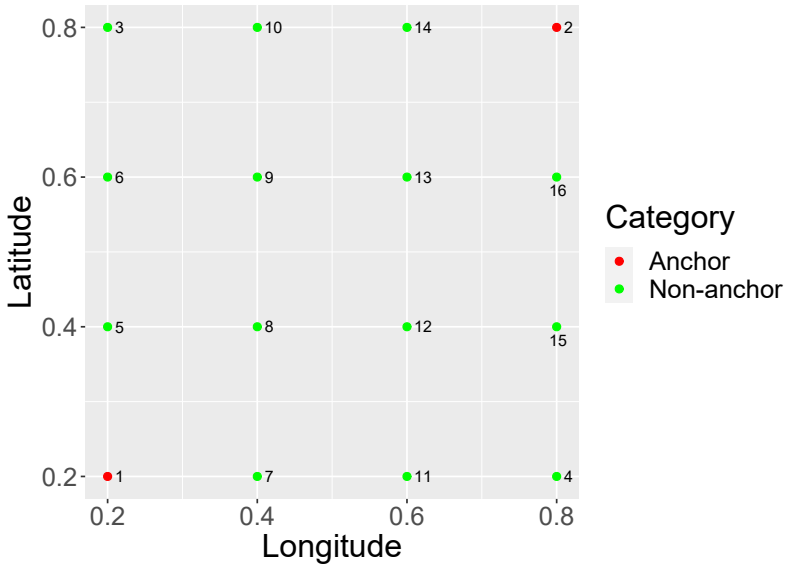


Figure 3.1: Geographic region of interest ($\mathcal{S}$) of the first simulation study in Chapter 3.

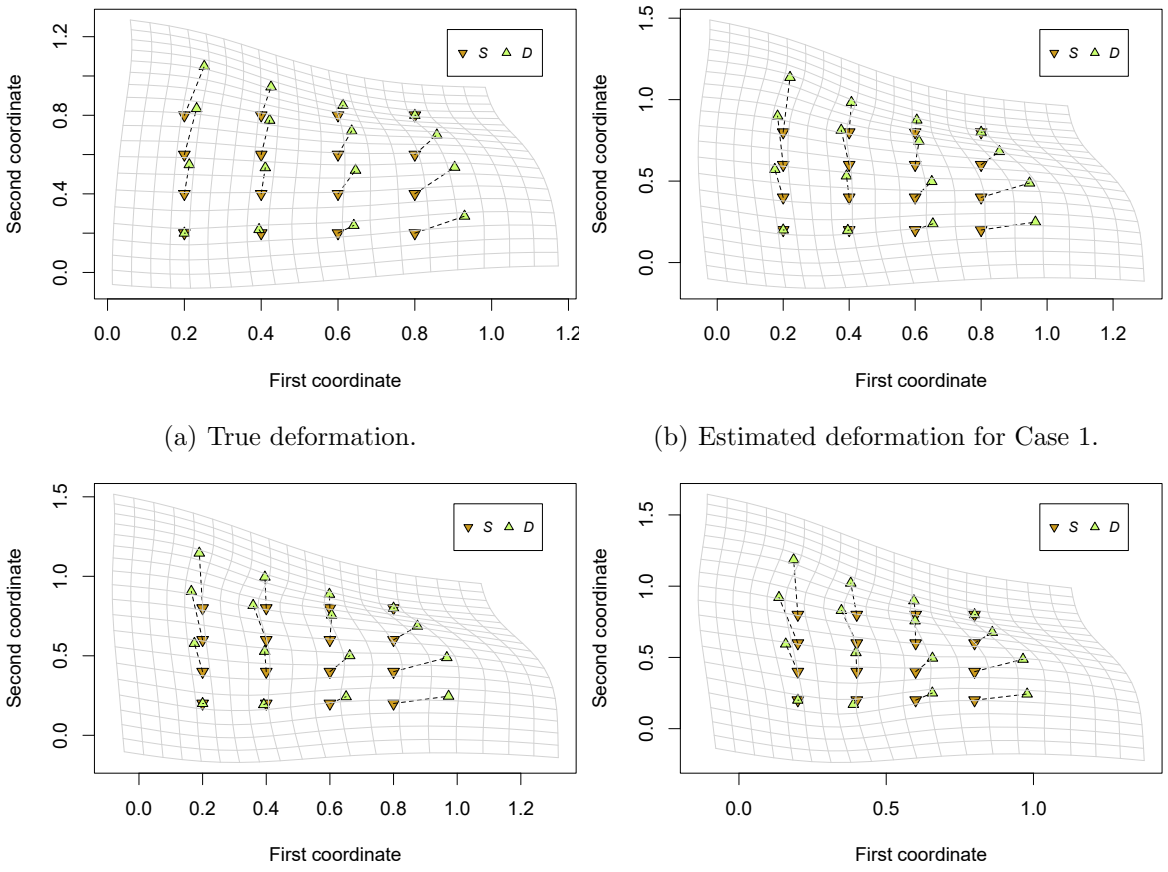
To run Algorithm 11, we use the quantities generated on 3(a)-(e) considering Case 1, 2 or 3: the matrices $\mathbf{W}$, $\mathbf{G}_1, \ldots, \mathbf{G}_{200}$, and $\mathbf{X}_1, \ldots, \mathbf{X}_{200}$, and the sub-vectors $\underset{\sim}{\mathbf{Y}}_{1,\mathsf{obs}}, \ldots, \underset{\sim}{\mathbf{Y}}_{200,\mathsf{obs}}$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\boldsymbol{\Sigma}} = 0.001$, $\mathbf{b}_{\boldsymbol{\Sigma}} = 0.001 \cdot \mathbf{I}_2$, $a_\phi = 0.001$, $b_\phi = 0.001$, $\mathbf{M}_0 = \mathbf{0}_{2\times 2}$, $\mathbf{C}_0 = \mathbf{I}_2$, and $\boldsymbol{\sigma}_d^2$ as the empirical covariance matrix of the gauged sites. We choose $\psi = 9.85$ after a few tries. In Appendix B.2.4, we present the code written in Python for the data generation scheme related to the first simulation study and specifying these hyperparameters.

Algorithm 11 was run 12000 times for all three scenarios (Cases 1, 2 and 3), having presented convergence after $J \approx 2000$ iterations. To avoid autocorrelation in the chains, we form a sample of size $K = 1000$ of the posterior distribution of the parameters by systematically sampling every ten iterations (i.e. $j_1 = 2001, j_2 = 2011, \ldots, j_{1000} = 11991$). The acceptance rates of the parameter ϕ are equal to 41.91%, 42.17% and 42.70%, respectively for Cases 1, 2 and 3. The trace plots for the posterior distributions of $\phi \cdot V \cdot \Sigma_{i,i'}$ and $\mathbf{D}$ from this simulation study are available in Appendix C.2.1.

As $N = 16$, Cases 1, 2 and 3 generate the respective percentages of missing values per response variable: 6.25%, 12.5% and 25%. The true deformation and the estimated deformations for the three sample sizes are shown in Figure 3.2, where we visually notice that the estimation of the true deformation worsens as the fraction of missing values grows. Only in Case 1 all components of $\mathbf{D}$ were well estimated. Figure C.16(e) shows that $D_{1,9}$ was not well estimated in Case 2, while Figures C.15(l), C.16(f), C.17(c) and C.18(c) respectively show that components $D_{1,6}$, $D_{1,9}$, $D_{1,13}$ and $D_{2,3}$ were not well estimated in Case 3.

Figure 3.3 shows the posterior distribution of $\phi \cdot V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$. We study the behaviour of $\phi \cdot V \cdot \Sigma_{i,i'}$ due to the lack of identifiability problem discussed in Section 2.5.1. It is noted that the parameters were well estimated in Cases 1, 2 and 3, noting that the distance between the posterior means and the true values increases as the fraction of missing values increases. Figures 3.4, 3.5, 3.6 and 3.7 show that the posterior distributions of $\beta_{0,1,t}, \beta_{0,2,t}, \beta_{1,1,t}, \beta_{1,2,t}$ $(t = 0, 1, \ldots, 200)$ estimate well their true values for all the three scenarios.

Thus, through this simulation study we conclude that Algorithm 11 is suitable to correctly retrieve all the parameters of the model given in Equation (3.1) when the percentage of missing values in each response variable is equal to 6.25%.



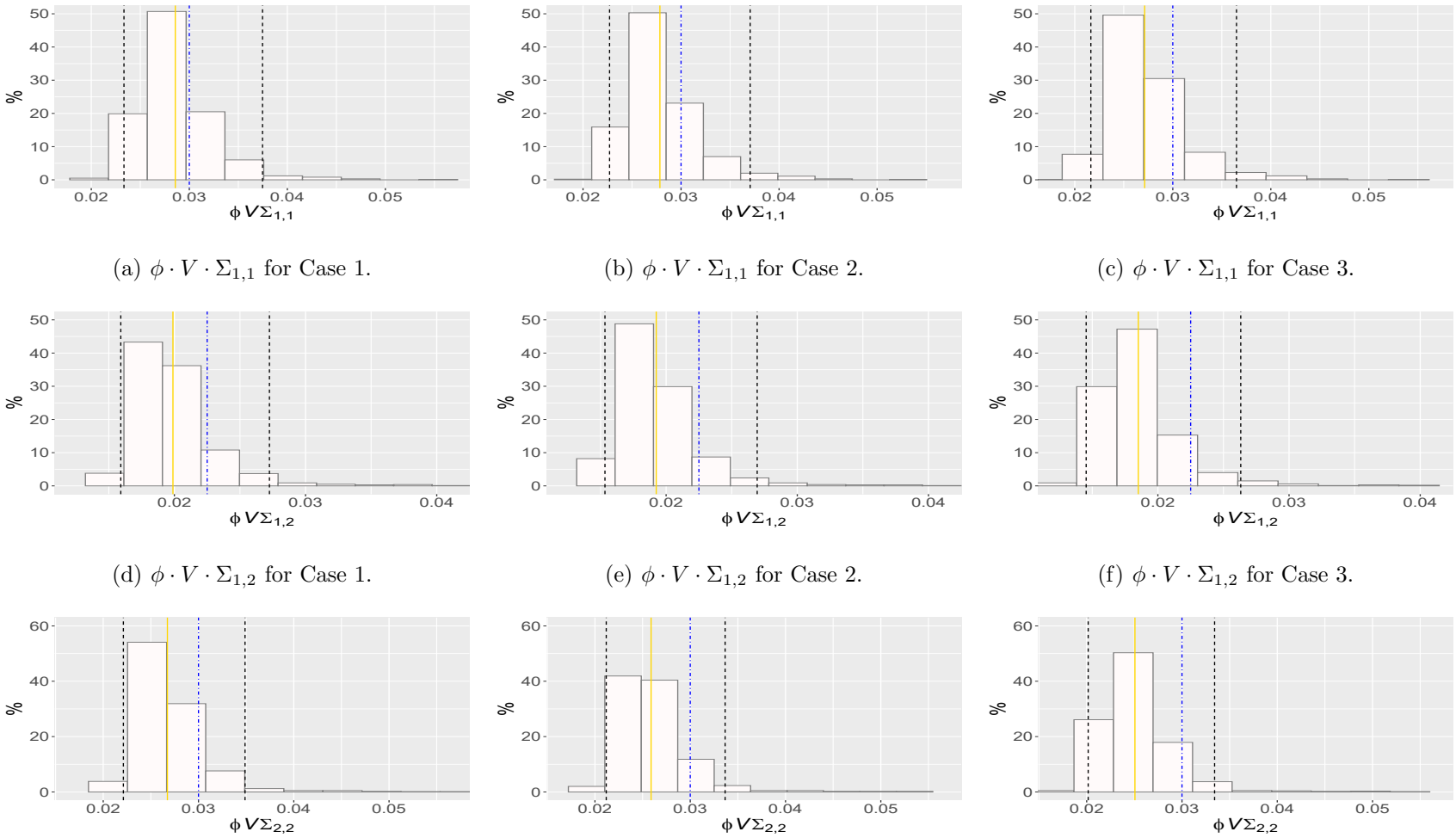
(a) True deformation.

(b) Estimated deformation for Case 1.

(c) Estimated deformation for Case 2.

(d) Estimated deformation for Case 3.

Figure 3.2: True deformation and posterior means of **D** for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3.



(a) $\phi \cdot V \cdot \Sigma_{1,1}$ for Case 1. (b) $\phi \cdot V \cdot \Sigma_{1,1}$ for Case 2. (c) $\phi \cdot V \cdot \Sigma_{1,1}$ for Case 3.

(d) $\phi \cdot V \cdot \Sigma_{1,2}$ for Case 1. (e) $\phi \cdot V \cdot \Sigma_{1,2}$ for Case 2. (f) $\phi \cdot V \cdot \Sigma_{1,2}$ for Case 3.

(g) $\phi \cdot V \cdot \Sigma_{2,2}$ for Case 1. (h) $\phi \cdot V \cdot \Sigma_{2,2}$ for Case 2. (i) $\phi \cdot V \cdot \Sigma_{2,2}$ for Case 3.

Figure 3.3: Histograms of the posterior distribution of $\phi \cdot V \cdot \Sigma_{1,1}$ ($i \in \{1, 2\}$) for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

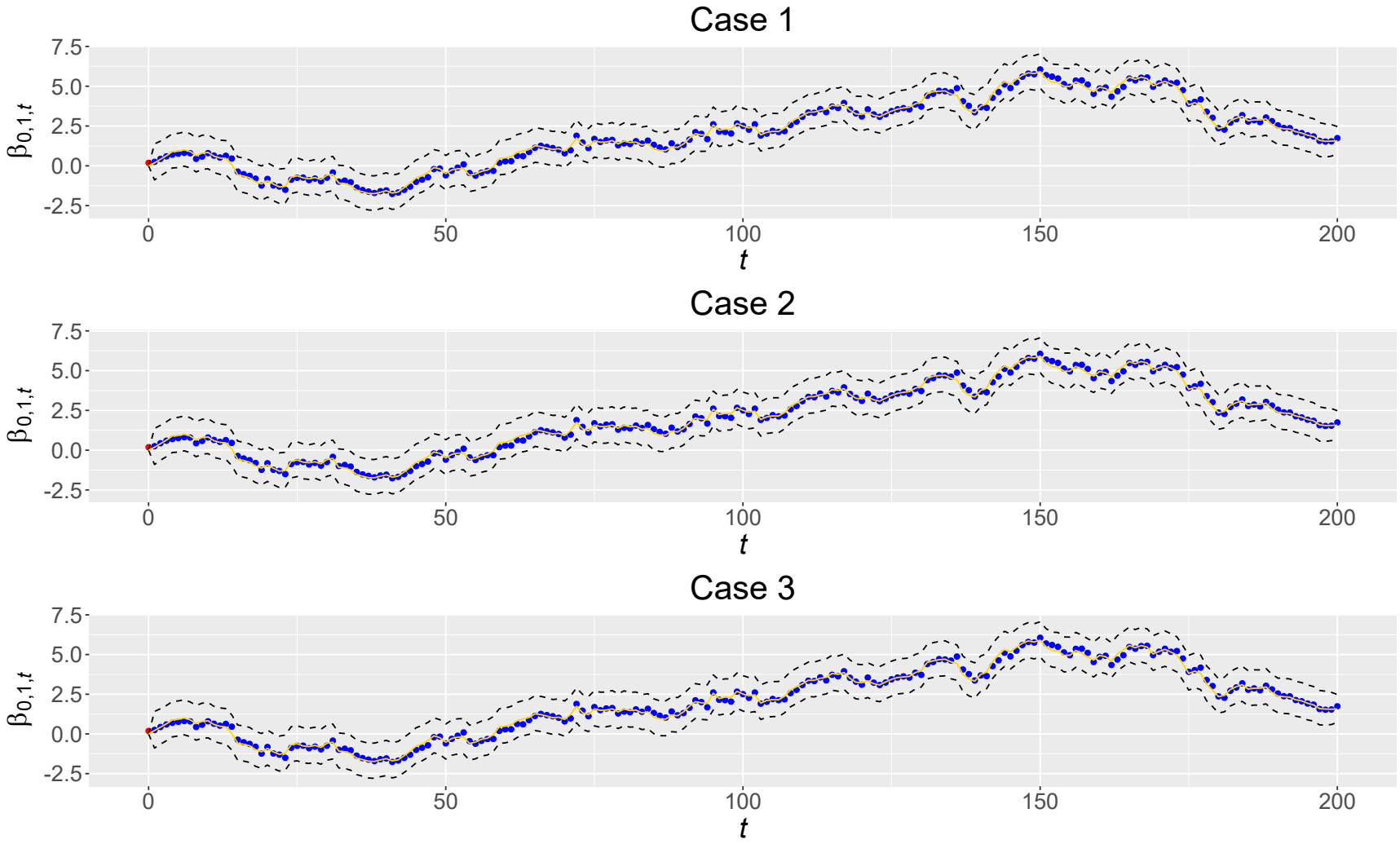


Figure 3.4: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, 200\}$ and Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

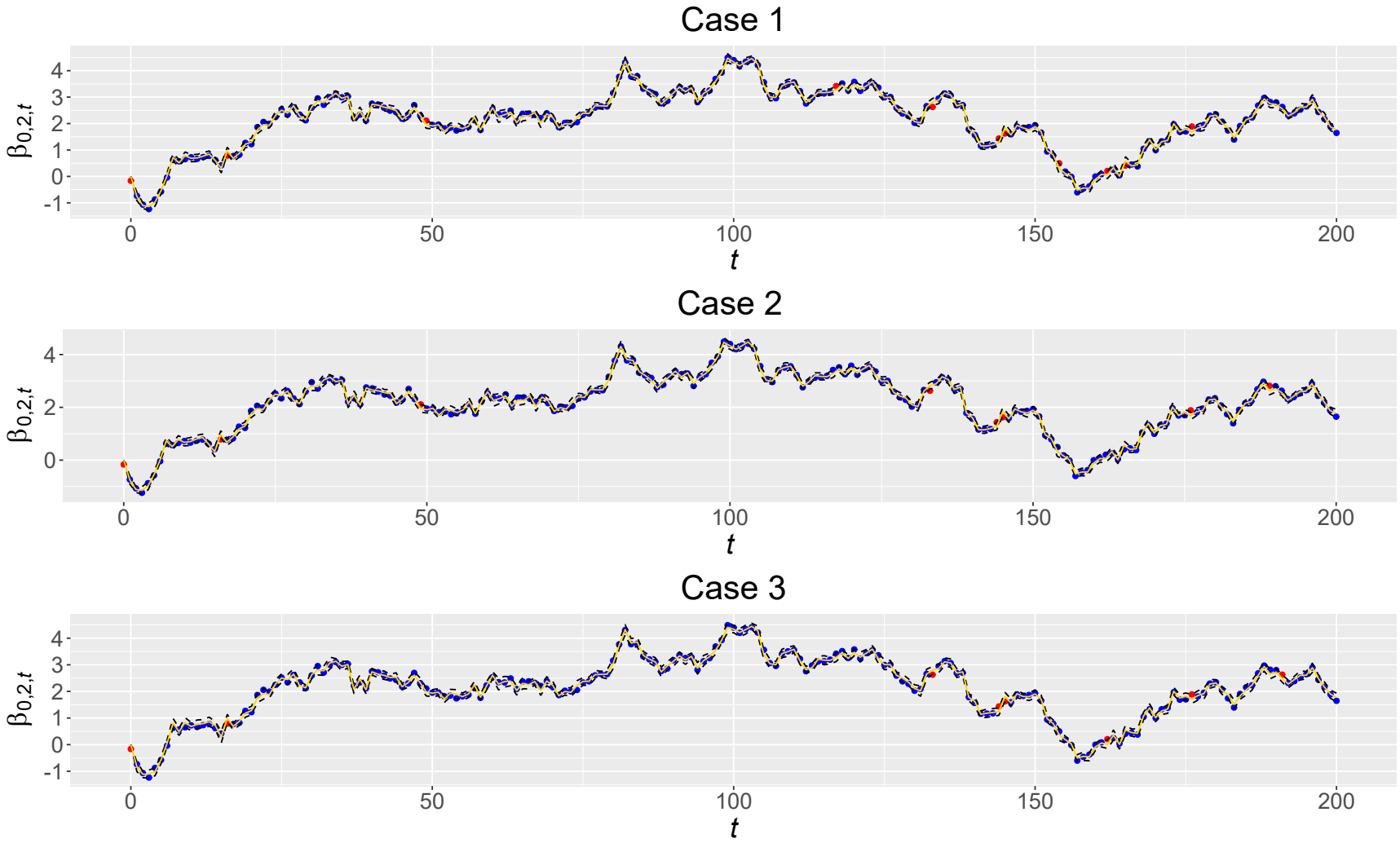


Figure 3.5: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, 200\}$ and Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

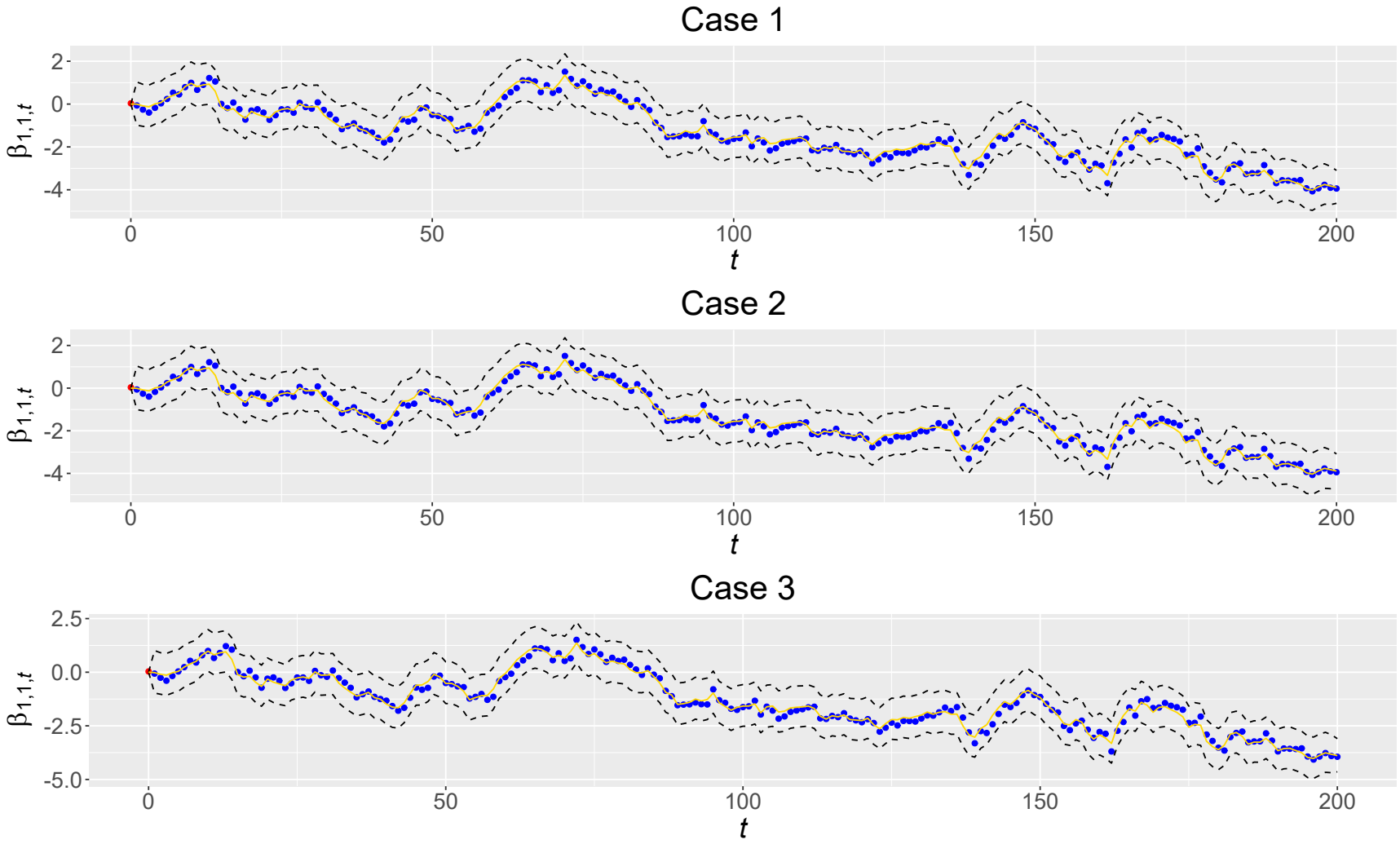


Figure 3.6: Line chart of the posterior distribution of $\beta_{1,1,t}$ for $t \in \{0, 1, \ldots, 200\}$ and Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

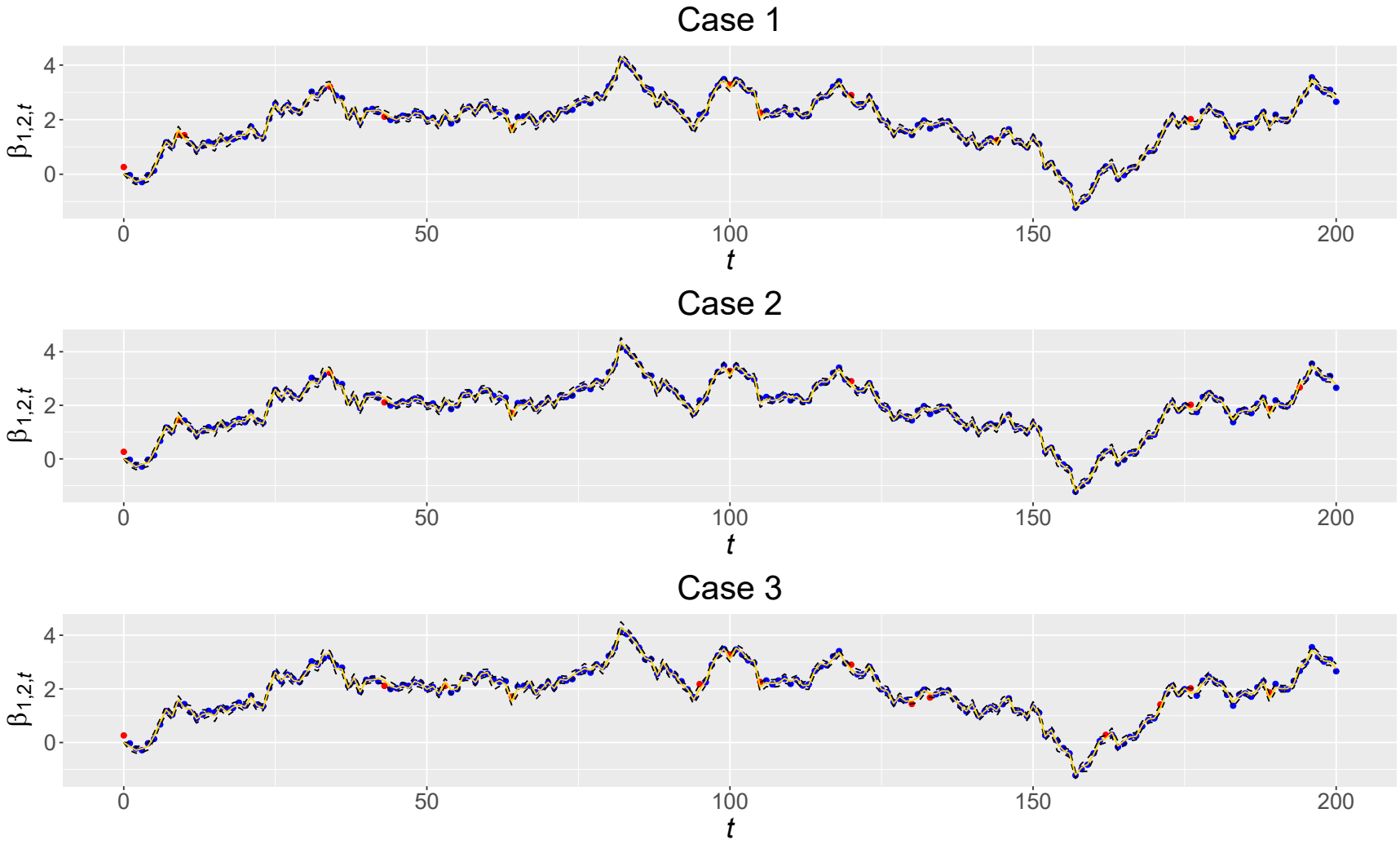


Figure 3.7: Line chart of the posterior distribution of $\beta_{1,2,t}$ for $t \in \{0, 1, \ldots, 200\}$ and Cases 1, 2, and 3, resulting from the first simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

3.5.2 Simulation 2

In this second simulation study, we will generate anisotropic data with some missing values to study the predictive performance (forecasting and interpolation) of the anisotropic model ($\mathcal{M}_{\text{A}}$) given in Equation (3.1) and compare its results with those obtained through an analogous isotropic model ($\mathcal{M}_{\text{I}}$) given in Equation (3.7) with $B_{n,n'} = \exp\{-\phi\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}$.

The generation of anisotropic data will be done in a very similar way to what was presented in Section 2.5.2. We consider $q = 2$ response variables, $p = 2$ regression coefficients per time and response variable, and $T + T^\star = 110$ periods of time, where the first $T = 100$ times are used to fit the model and the last $T^\star = 10$ times are generated to study the forecasting performance. We simulate the process Y at $N = 16$ equally spaced points in the unit square and, jointly, also simulate the process Y^* at $N^* = 3$ not equally spaced points in the unit square. We are interested in interpolating $\mathbf{Y}_t^*$ using the anisotropic and isotropic models given by Equations (3.1) and (3.7) after observing $\mathbf{Y}_t$, and compare to their real generated values. Figure 3.8 shows the geographic region and highlights the anchor points ($\underset{\sim}{\mathbf{s}}_1$ and $\underset{\sim}{\mathbf{s}}_2$), the non-anchor points ($\underset{\sim}{\mathbf{s}}_3, \ldots, \underset{\sim}{\mathbf{s}}_{16}$) and the sites considered as ungauged to study the interpolation performance ($\underset{\sim}{\mathbf{s}}_{17}$, $\underset{\sim}{\mathbf{s}}_{18}$ and $\underset{\sim}{\mathbf{s}}_{19}$). We generate data from the following anisotropic scheme:

1. Generate $\mathbf{D} = \begin{bmatrix} \mathbf{D}_{1:2} & \mathbf{D}_{3:16} \end{bmatrix}_{2\times 16}$ and $\mathbf{D}^* = \begin{bmatrix} \mathbf{D}_{17} & \mathbf{D}_{18} & \mathbf{D}_{19} \end{bmatrix}_{2\times 3}$ proceeding as follows:
   (a) Do $\mathbf{D}_{1:2} = \mathbf{S}_{1:2}$ (i.e. $d(\underset{\sim}{\mathbf{s}}_1) = \underset{\sim}{\mathbf{s}}_1$ and $d(\underset{\sim}{\mathbf{s}}_2) = \underset{\sim}{\mathbf{s}}_2$) to fix the locations of two sites in $\mathcal{D}$-space, where $\mathbf{S} = \begin{bmatrix} \mathbf{S}_{1:2} & \mathbf{S}_{3:16} \end{bmatrix}_{2\times 16}$.
   (b) Do $\underset{\sim}{\mathbf{d}}_n = d(\underset{\sim}{\mathbf{s}}_n) = \boldsymbol{\Lambda} \cdot \underset{\sim}{\mathbf{s}}_n$ for $n \in \{3, \ldots, 16, 17, 18, 19\}$ to generate deterministic deformations that induce geometric anisotropy, where $\boldsymbol{\Lambda}$ is such that $\boldsymbol{\Lambda}^\intercal\boldsymbol{\Lambda} = \mathbf{A}$.

   Following Maity and Sherman (2012, Sec. 4), we use $\mathbf{A} = 9 \cdot \mathbf{I}_2 - 4 \cdot \mathbf{1}_{2\times 2}$. The Cholesky decomposition was applied to obtain $\boldsymbol{\Lambda}$, having:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \sqrt{5} & -4\sqrt{5}/5 \\ 0 & 3\sqrt{5}/5 \end{bmatrix}.$$

2. Set $V = 0.6$, $\phi = 0.4$ and $\boldsymbol{\Sigma} = \left[\begin{smallmatrix} 1.00 & 0.85 \\ 0.85 & 1.00 \end{smallmatrix}\right]$ as true parameters, generate $\boldsymbol{\beta}_0$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{0}_{2\times 2}, V \cdot \mathbf{I}_2, \boldsymbol{\Sigma})$ and compute the block matrix $\mathbf{B}^{\mathsf{aug}} = \left[B_{n,n'}^{\mathsf{aug}}\right]_{19\times 19} = \begin{bmatrix} \mathbf{B} & \mathbf{B}_{\mathsf{g,u}} \\ \mathbf{B}_{\mathsf{u,g}} & \mathbf{B}^* \end{bmatrix}$, where $\mathbf{B}$ (16×16), $\mathbf{B}_{\mathsf{g,u}}$ (16×3), $\mathbf{B}_{\mathsf{u,g}}$ (3×16) and $\mathbf{B}^*$ (3×3) are four sub-matrices and, for all $n, n' \in \{1, \ldots, 16, 17, 18, 19\}$, $B_{n,n'}^{\mathsf{aug}}$ is a generic entry given by:

$$B_{n,n'}^{\mathsf{aug}} = \begin{cases} \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'})\|\}, & n \neq n' \\ 1, & n = n' \end{cases}.$$

3. For $t = 1, \ldots, 100, 101, \ldots, 110$ do the following:

   (a) Generate $\boldsymbol{\beta}_t$ from the distribution $\mathsf{N}_{2\times 2}(\mathbf{G}_t\boldsymbol{\beta}_{t-1}, V \cdot \mathbf{W}, \boldsymbol{\Sigma})$, where $\mathbf{G}_t = \mathbf{I}_2$ and $\mathbf{W} = \mathbf{I}_2$.

   (b) Generate $\mathbf{X}_t = [\,{}^1_{U_{1,t}}\ {}^1_{U_{2,t}}\ \cdots\ {}^1_{U_{16,t}}\,]^\intercal$ (16×2) and $\mathbf{X}^*_t = [\,{}^1_{U_{17,t}}\ {}^1_{U_{18,t}}\ {}^1_{U_{19,t}}\,]^\intercal$ (3×2), where $U_{1,t}, \ldots, U_{19,t}$ are nineteen random numbers between zero and one.

   (c) Generate $\mathbf{Y}_t$ and $\mathbf{Y}^*_t$ jointly from the following distribution:

$$\mathsf{N}_{19\times 2}\left(\begin{bmatrix}\mathbf{X}_t\\ \mathbf{X}^*_t\end{bmatrix} \cdot \boldsymbol{\beta}_t, V \cdot \mathbf{B}^{\mathsf{aug}}, \boldsymbol{\Sigma}\right).$$

   (d) For each column of $\mathbf{Y}_t$ (i.e. $\underset{\sim}{\mathbf{Y}}_{i,t}$ for $i \in \{1, \ldots, q\}$), choose two numbers uniformly and randomly from the set $\{1, \ldots, N\}$, without replacement, say n_1 and n_2, and let $Y_{n_1,i,t} = Y_{n_2,i,t} = \texttt{NA}$.

   (e) Write the vector $\underset{\sim}{\mathbf{Y}}_t = \mathrm{vec}(\mathbf{Y}_t)$ and obtain the sub-vector $\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}} = \mathbf{L}_{t,\mathsf{obs}}\mathbf{P}_t\underset{\sim}{\mathbf{Y}}_t$.

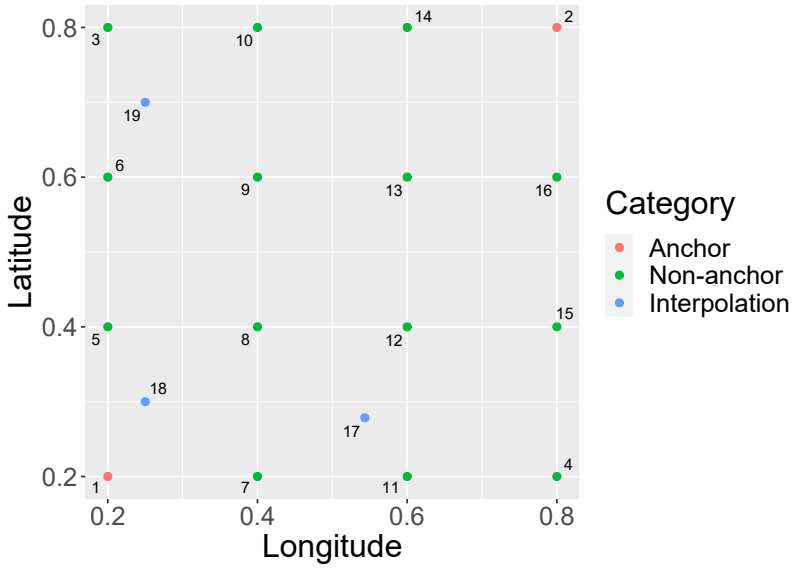


Figure 3.8: Geographic region ($\mathcal{S}$) of the second simulation study in Chapter 3.

To run Algorithm 11, we use the quantities generated on 3(a)-(e): the matrices $\mathbf{W}, \mathbf{G}_1, \ldots, \mathbf{G}_{100}$, and $\mathbf{X}_1, \ldots, \mathbf{X}_{100}$, and the sub-vectors $\underset{\sim}{\mathbf{Y}}_{1,\mathsf{obs}}, \ldots, \underset{\sim}{\mathbf{Y}}_{100,\mathsf{obs}}$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\boldsymbol{\Sigma}} = 0.001$, $\mathbf{b}_{\boldsymbol{\Sigma}} = 0.001 \cdot \mathbf{I}_2$, $a_\phi = 0.001$, $b_\phi = 0.001$, $\mathbf{M}_0 = \mathbf{0}_{2\times 2}$, $\mathbf{C}_0 = \mathbf{I}_2$, and $\boldsymbol{\sigma}^2_d$ as the empirical covariance matrix of the gauged sites. We choose $\psi = 9.85$ after a few tries. In Appendix B.2.5, we present a code implemented in Python for the data generation scheme related to the second simulation study.

Algorithm 11 was run for 20000 iterations, having converged after $J \approx 5000$ iterations. The same configuration was adopted for sampling from the analogous isotropic model. To avoid autocorrelation in the chains, we form a sample of size $K = 1000$ of the posterior distribution of the parameters by systematically sampling every 15 iterations (i.e. $j_1 = 5001, j_2 = 5016, \ldots, j_{1000} = 19986$). The acceptance rates of the parameter ϕ are equal to 45.16% and 41.23%, respectively for the models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$. The trace plots for the posterior distributions of the parameters $\phi \cdot V \cdot \Sigma_{i,i'}$ and $\mathbf{D}$ from this simulation study are available in Appendix C.2.2.

True and estimated deformations are shown in Figure 3.9, where visually it can be seen that the true deformation is well estimated. Through the samples $\mathbf{D}^{(j_1)}, \ldots, \mathbf{D}^{(j_{1000})}$, the entries of matrix $\mathbf{\Lambda}$ can be recovered by applying the solution described in Appendix A.4 – it can be verified in Figure 3.10. We study the behaviour of the product $\phi \cdot V \cdot \Sigma_{i,i'}$ due to the lack of identifiability problem discussed in Section 2.5.1. Figures 3.11(a)-3.11(f) show that the posterior distributions of the parameters $\phi \cdot V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ estimate well their true values only on model $\mathcal{M}_{\mathrm{A}}$. Figures 3.12, 3.13, 3.14 and 3.15 show that $\beta_{0,1,t}, \beta_{0,2,t}, \beta_{1,1,t}, \beta_{1,2,t}$ $(t = 0, 1, \ldots, 110)$ estimate well their true values for both models ($\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$).

To facilitate visual comparison of the missing data estimation, forecast and interpolation under each model, we divide the time series into three periods: $t \in \{1, \ldots, 36\}$, $t \in \{37, \ldots, 73\}$ and $t \in \{74, \ldots, 110\}$.

Figures 3.16-3.21 show the estimated missing values and predictions at the gauged site $\underset{\sim}{\mathbf{s}}_{14}$, for both response variables. Since the missing values were chosen randomly, note that these absences do not occur at the same times for each response variable. Using the gauged site $\underset{\sim}{\mathbf{s}}_{14}$ as a case in point once more, it is worth noting the absence of a value at time $t = 14$ in Response 1 (Figure 3.16), whereas conversely, it is observed as a value in Response 2 (Figure 3.19). For Response 1, the histograms of the estimated values at the gauged site $n = 14$ and time $t = 14$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$ are given respectively in Figures 3.22(a) and 3.22(b). For Response 2, the histograms of the estimated values at the gauged site $n = 14$ and time $t = 2$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$ are given respectively in Figures 3.22(c) and 3.22(d). The results show that there is more advantage for the anisotropic model when it comes to estimating missing values, with narrower credibility intervals and smaller differences between the means and the true values.

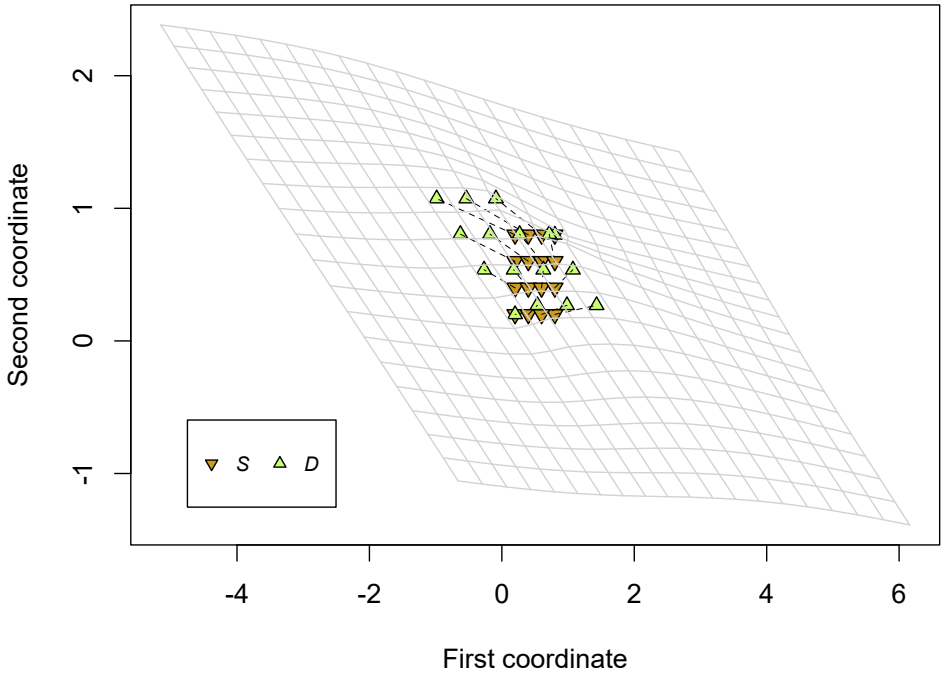
Table 3.1 shows some metrics for comparing models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$. Based on the DIC and PMSE statistics, $\mathcal{M}_{\mathrm{A}}$ fits the data better than $\mathcal{M}_{\mathrm{I}}$. Figures 3.23-3.28 show that the interpolation performance of model $\mathcal{M}_{\mathrm{A}}$ is superior to the one of model $\mathcal{M}_{\mathrm{I}}$ at the ungauged site $\underset{\sim}{\mathbf{s}}_{19}$, for both

response variables, noting that their credibility intervals are narrower, which is in agreement with the results of the ECP and IS statistics. The histograms of interpolated values at the third ungauged site and for time $t = 25$ are in Figure 3.29, for both response variables and both models, showing that the results are more accurate for the anisotropic model.

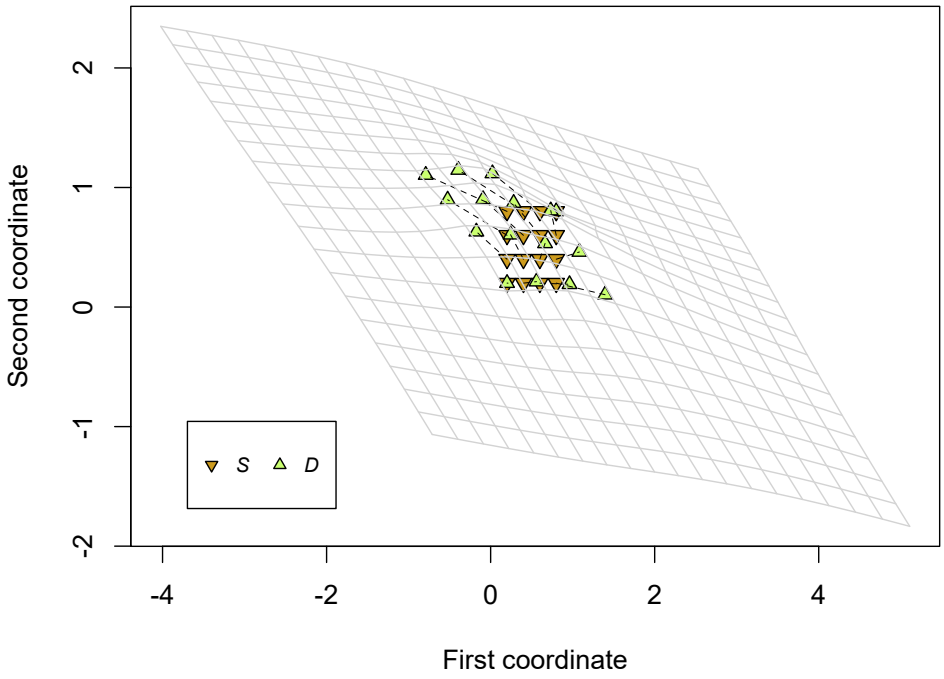
Thus, through this simulation study we concluded that, to analyze data with geometric anisotropy in the way we generate the data, model $\mathcal{M}_{\mathrm{A}}$ (with spatial deformation) makes more assertive interpolations than model $\mathcal{M}_{\mathrm{I}}$ (without spatial deformation).

Table 3.1: Metrics for model comparison (DIC, PMSE, ECP and IS by response variable and ungauged site), from the second simulation study in Chapter 3.

Metric	$\mathcal{M}_{\mathrm{A}}$		$\mathcal{M}_{\mathrm{I}}$	
DIC	388.8		1000.3	
PMSE	0.05579		0.06613	
ECP (%)	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{17}$	97.0	95.0	97.0	97.0
$\underset{\sim}{\mathbf{s}}_{18}$	98.0	96.0	98.0	96.0
$\underset{\sim}{\mathbf{s}}_{19}$	95.0	93.0	95.0	96.0
IS	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{17}$	0.02714	0.02866	0.03031	0.03206
$\underset{\sim}{\mathbf{s}}_{18}$	0.02784	0.02863	0.02929	0.03041
$\underset{\sim}{\mathbf{s}}_{19}$	0.02600	0.02870	0.02748	0.02947

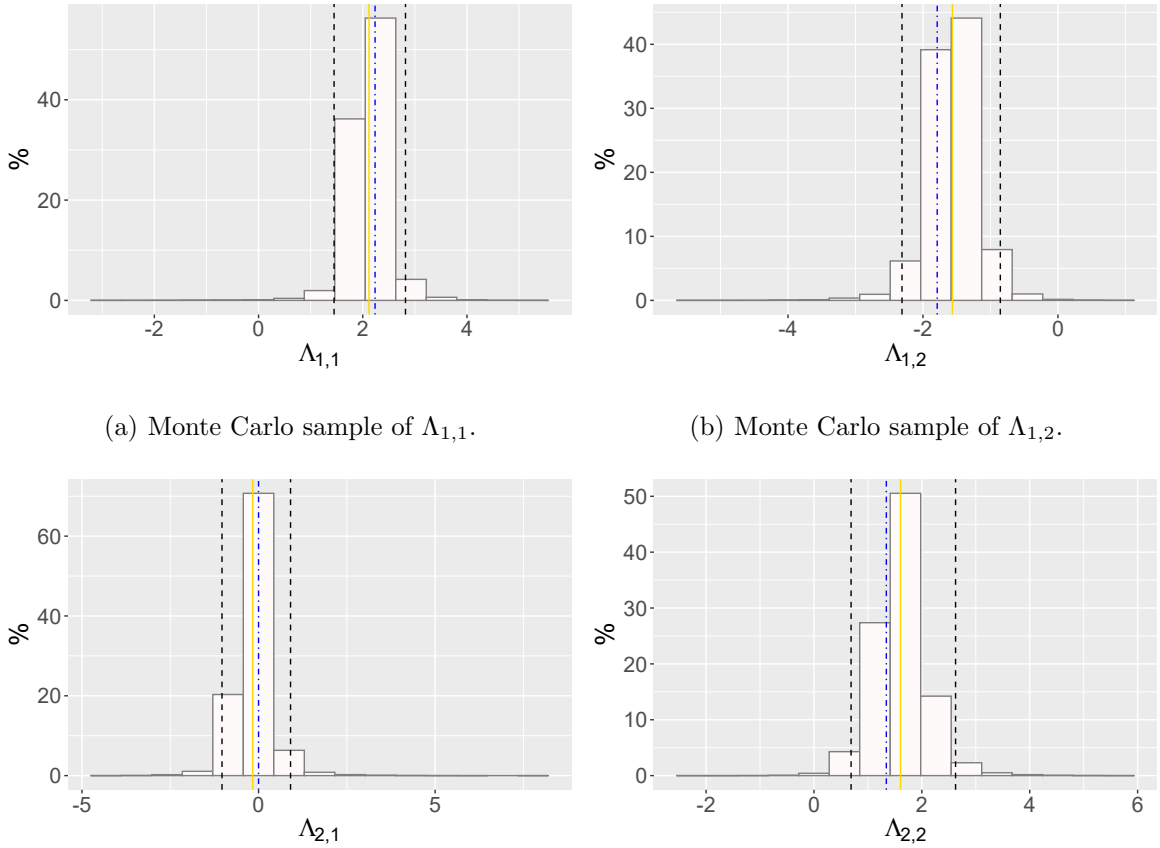


(a) True deformation.



(b) Estimated deformation.

Figure 3.9: True deformation and posterior means of $\mathbf{D}$, resulting from the second simulation study in Chapter 3.



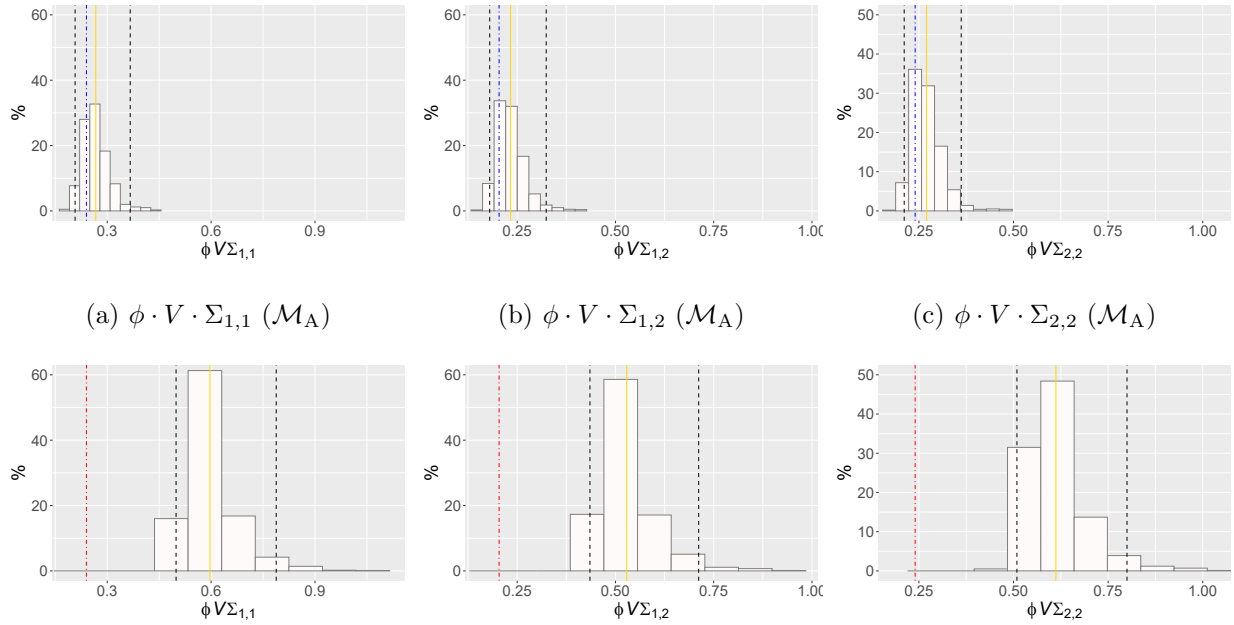
(a) Monte Carlo sample of $\Lambda_{1,1}$.

(b) Monte Carlo sample of $\Lambda_{1,2}$.

(c) Monte Carlo sample of $\Lambda_{2,1}$.

(d) Monte Carlo sample of $\Lambda_{2,2}$.

Figure 3.10: Histograms of the Monte Carlo samples of the parameters $\Lambda_{1,1}$, $\Lambda_{1,2}$, $\Lambda_{2,1}$ and $\Lambda_{2,2}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).



(a) $\phi \cdot V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $\phi \cdot V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $\phi \cdot V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $\phi \cdot V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $\phi \cdot V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $\phi \cdot V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$)

Figure 3.11: Histograms of the posterior distributions of $\phi \cdot V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

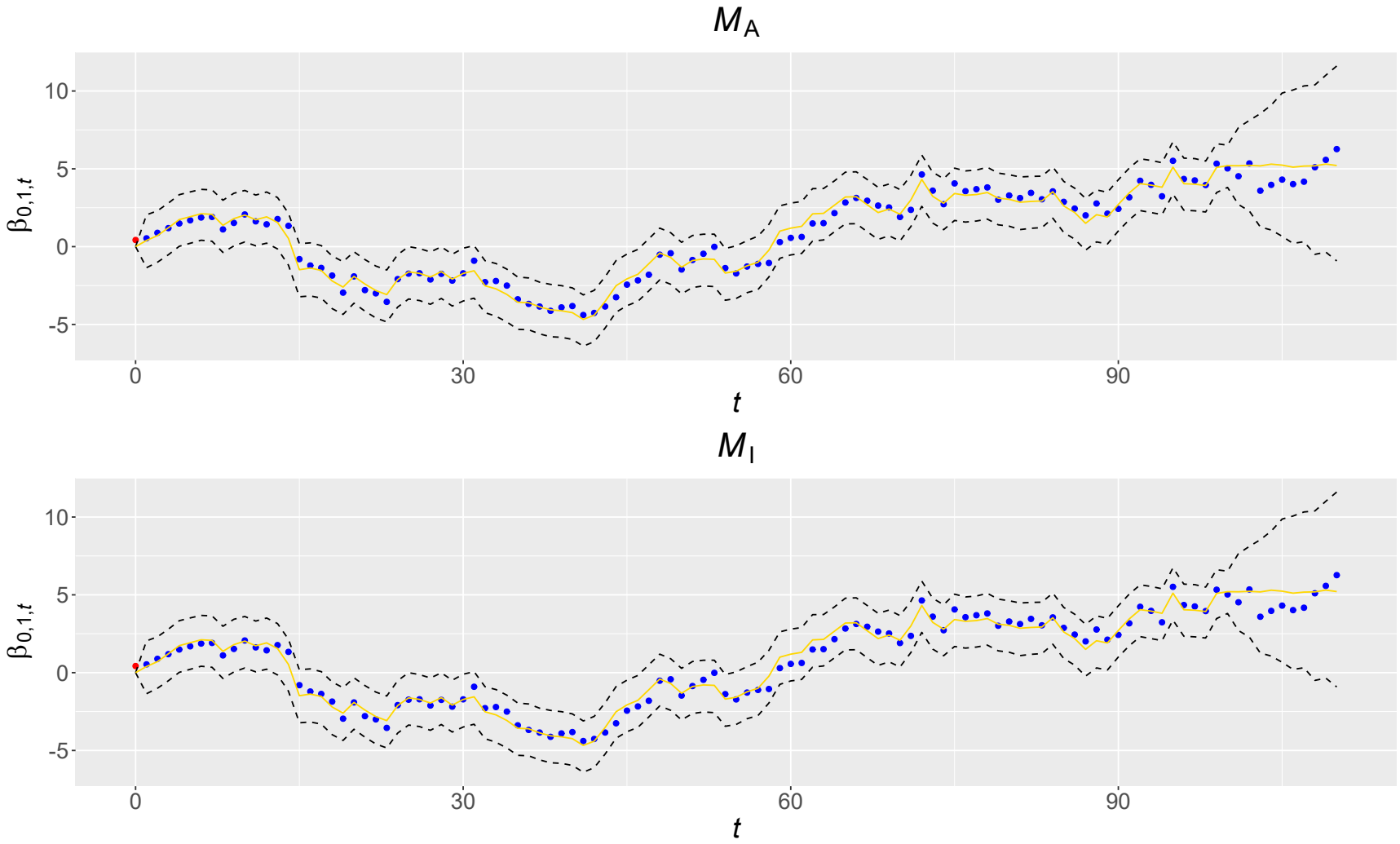


Figure 3.12: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

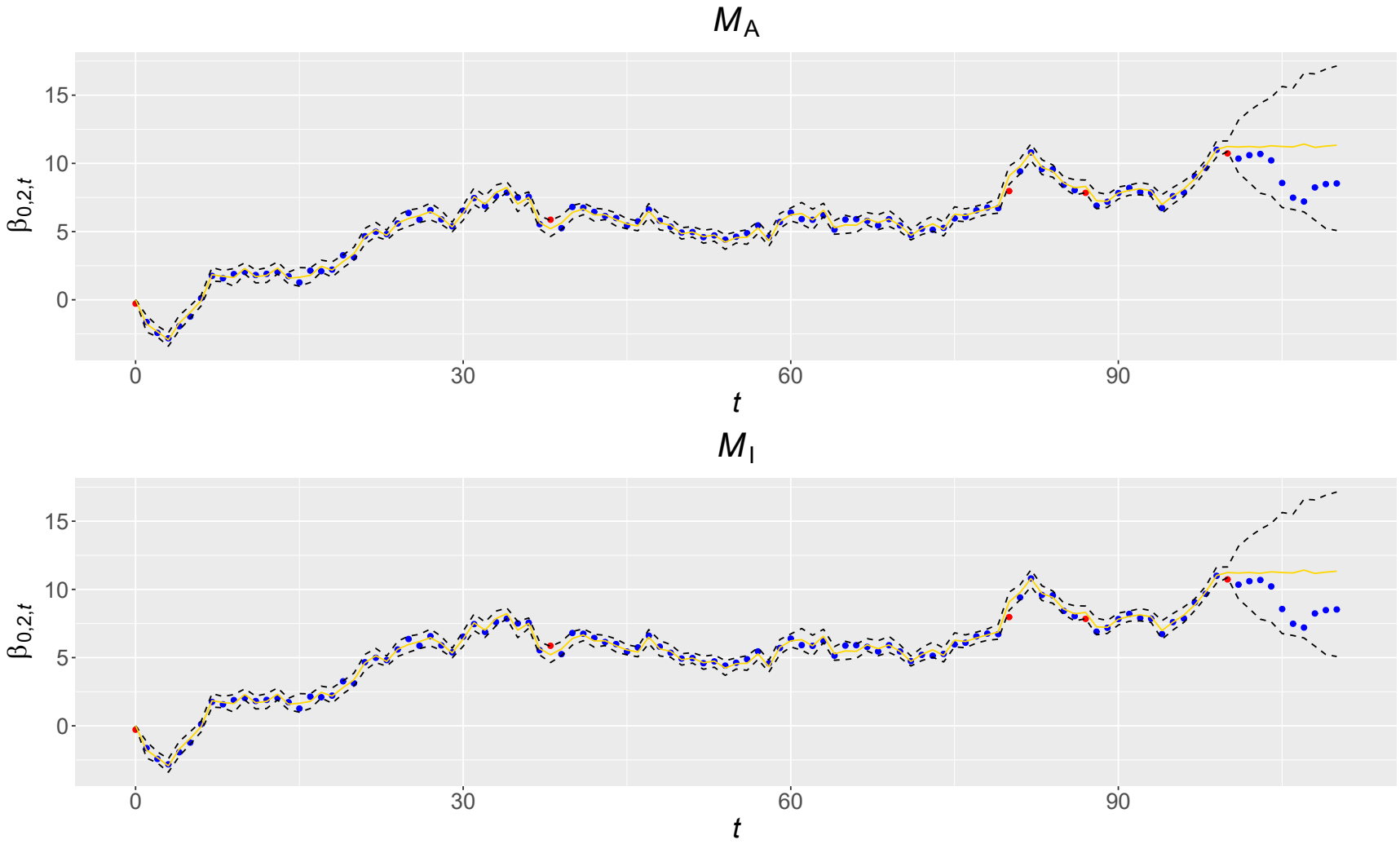


Figure 3.13: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

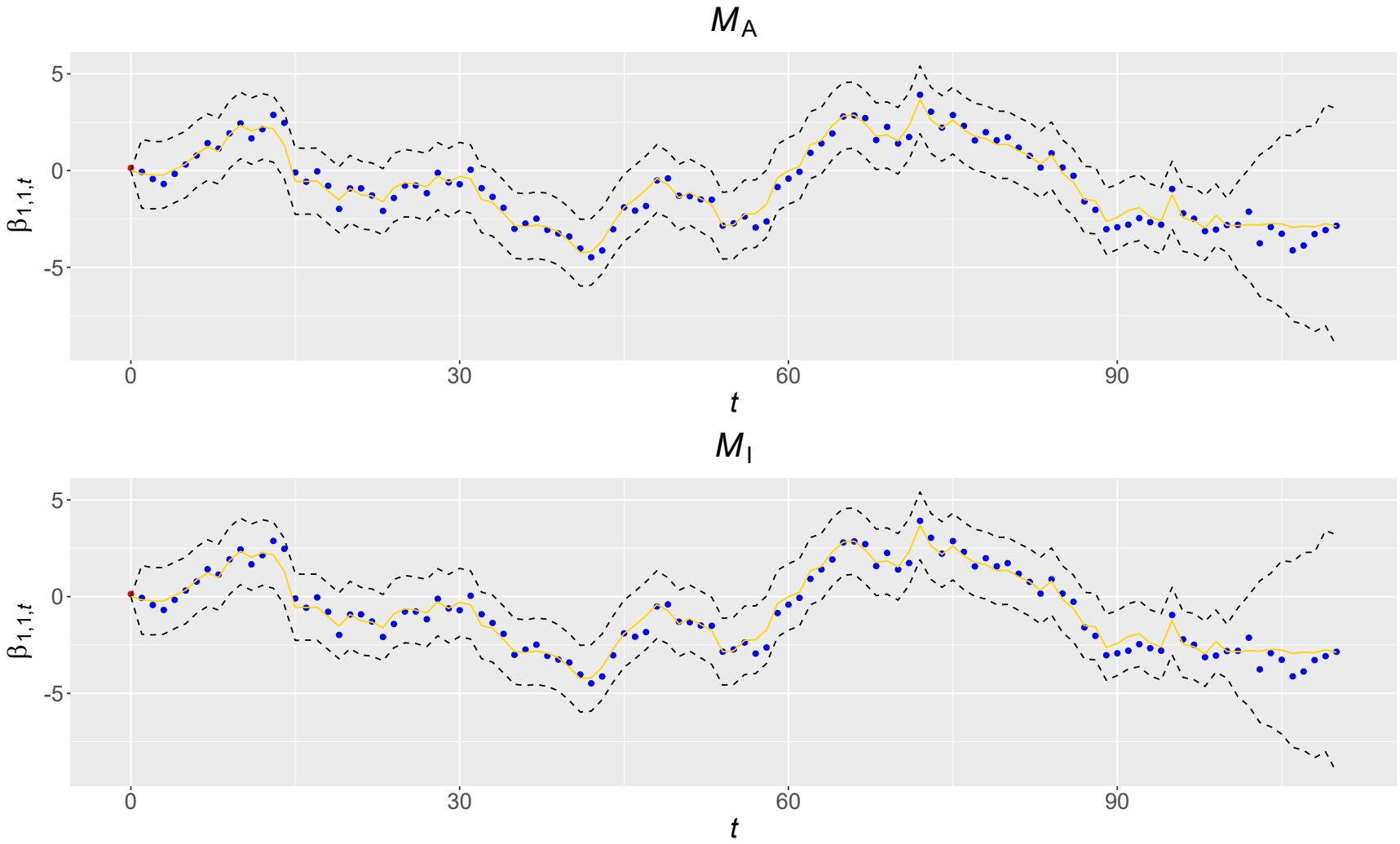


Figure 3.14: Line chart of the posterior distribution of $\beta_{1,1,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_A$ and $\mathcal{M}_I$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

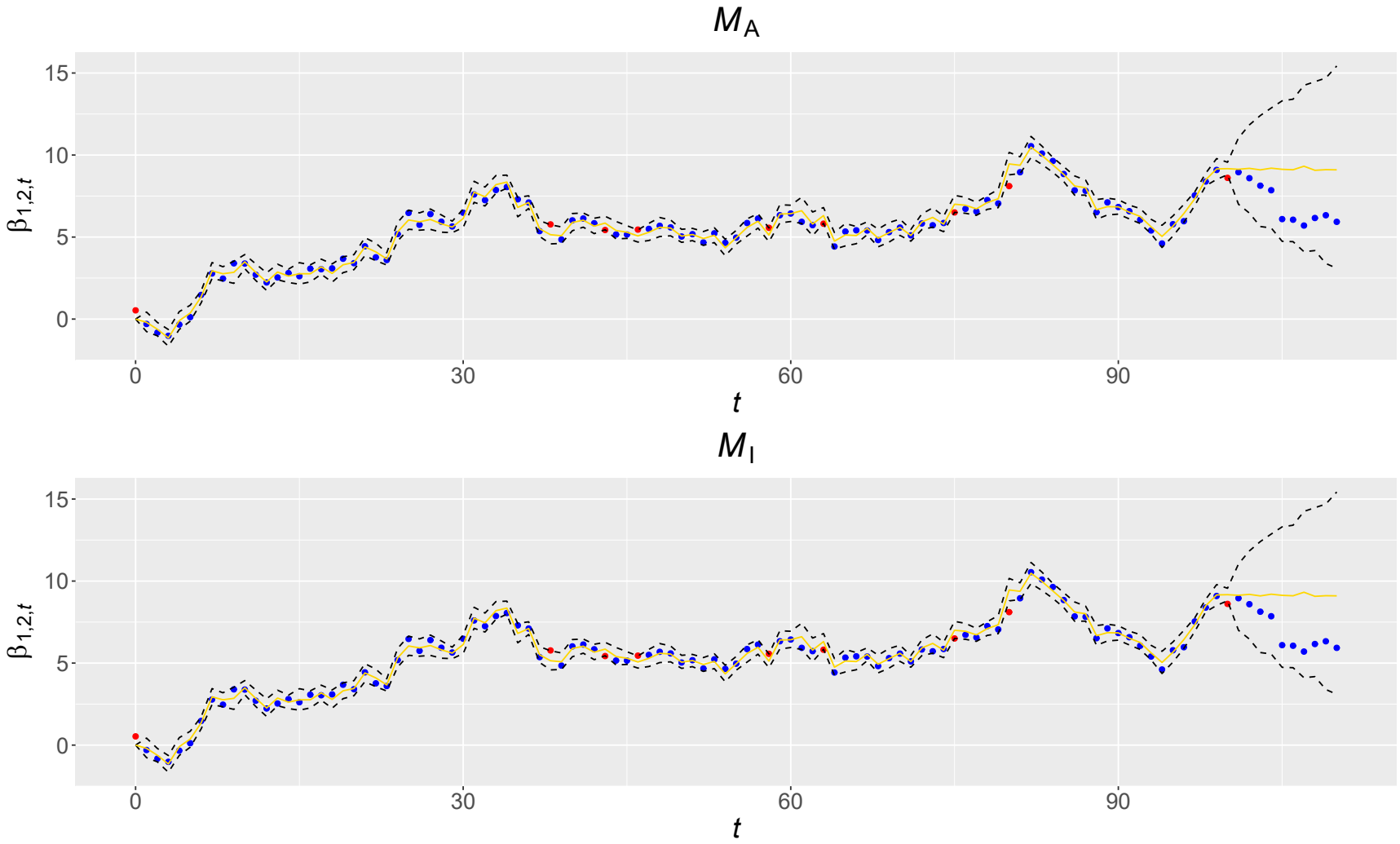


Figure 3.15: Line chart of the posterior distribution of $\beta_{1,2,t}$ for $t \in \{0, 1, \ldots, 110\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

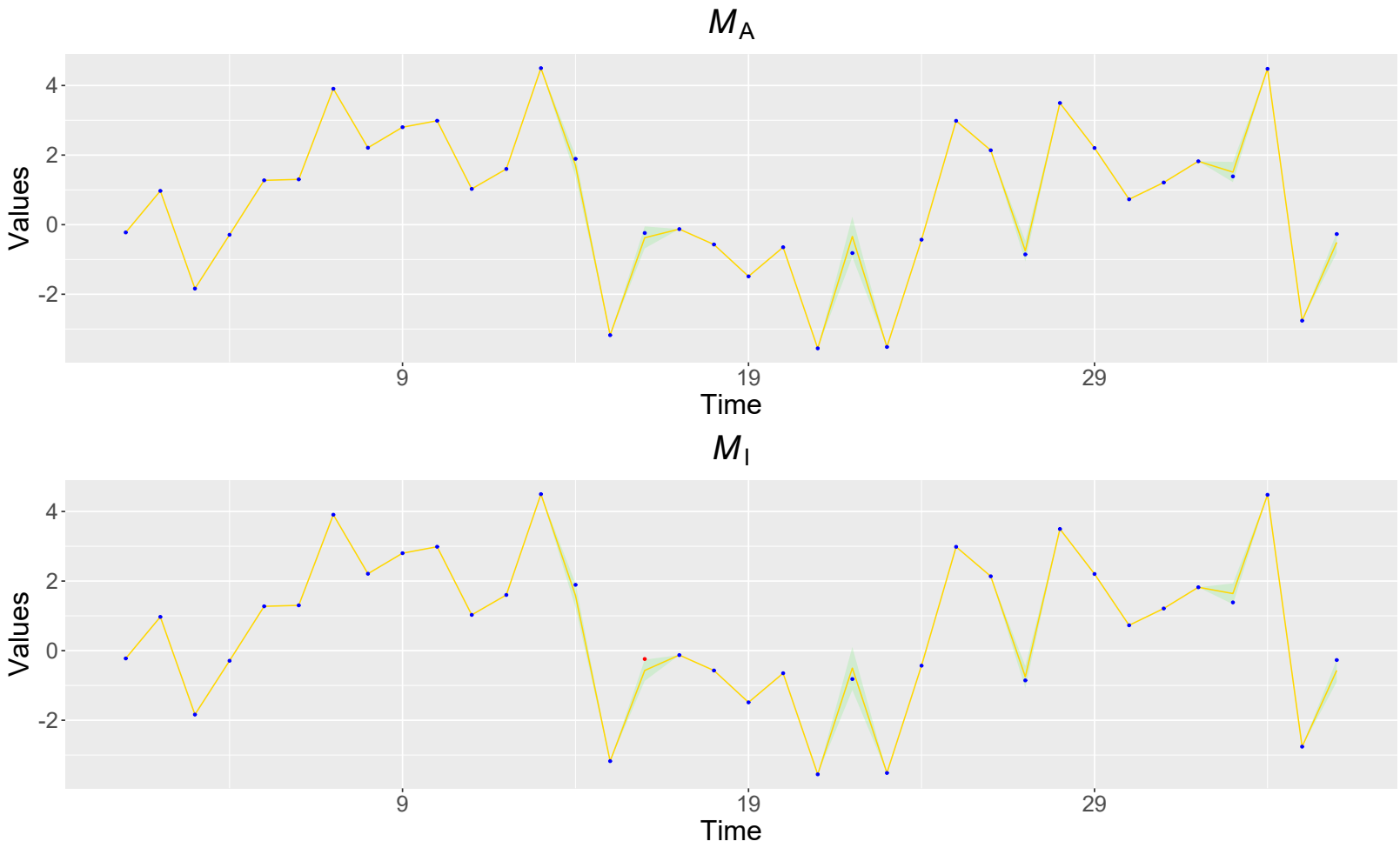


Figure 3.16: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 14$, $i = 1$ and $t \in \{1, \ldots, 36\}$ for models $\mathcal{M}_A$ and $\mathcal{M}_I$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

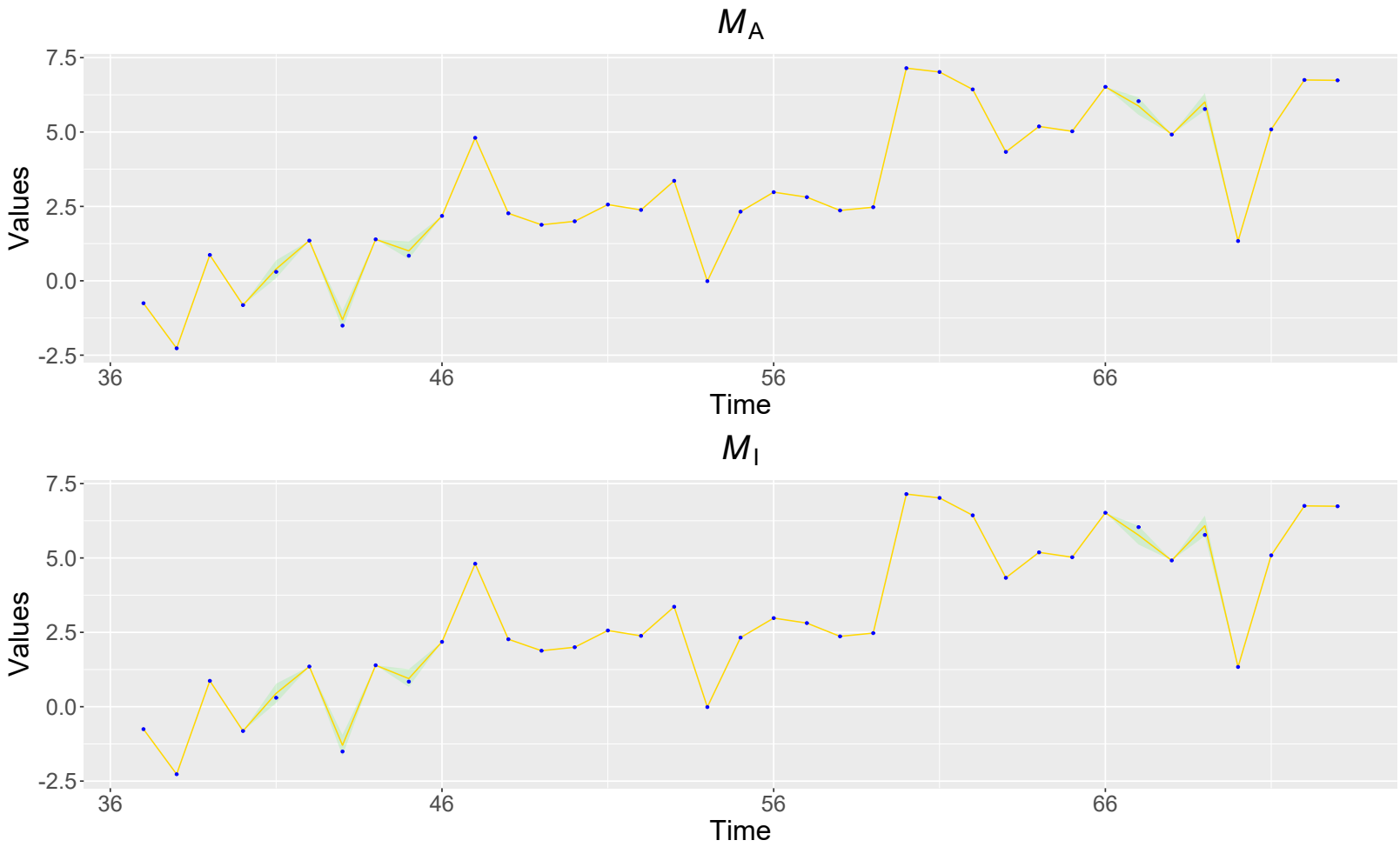


Figure 3.17: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 14$, $i = 1$ and $t \in \{37, \ldots, 73\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

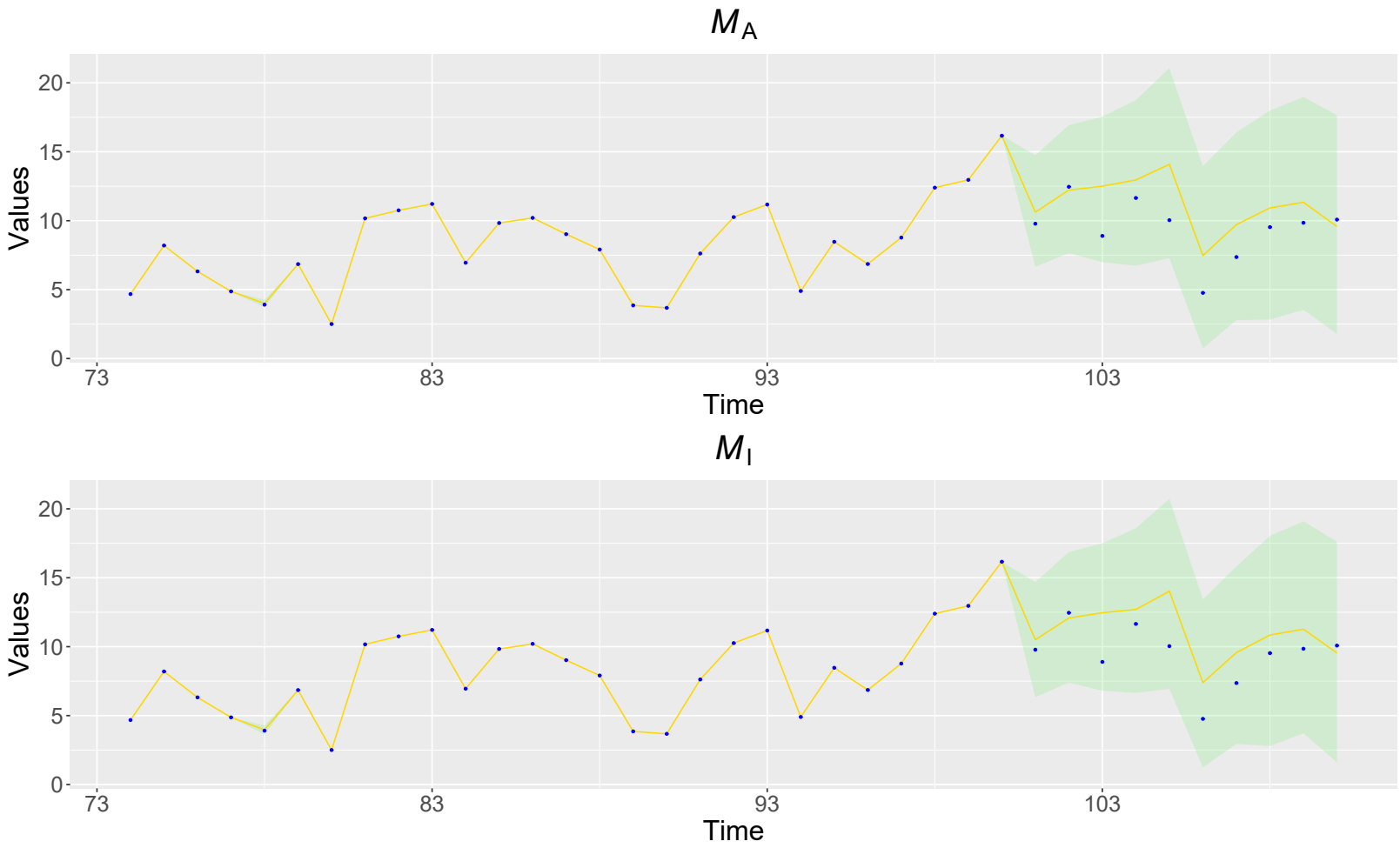


Figure 3.18: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 14$, $i = 1$ and $t \in \{74, \ldots, 110\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

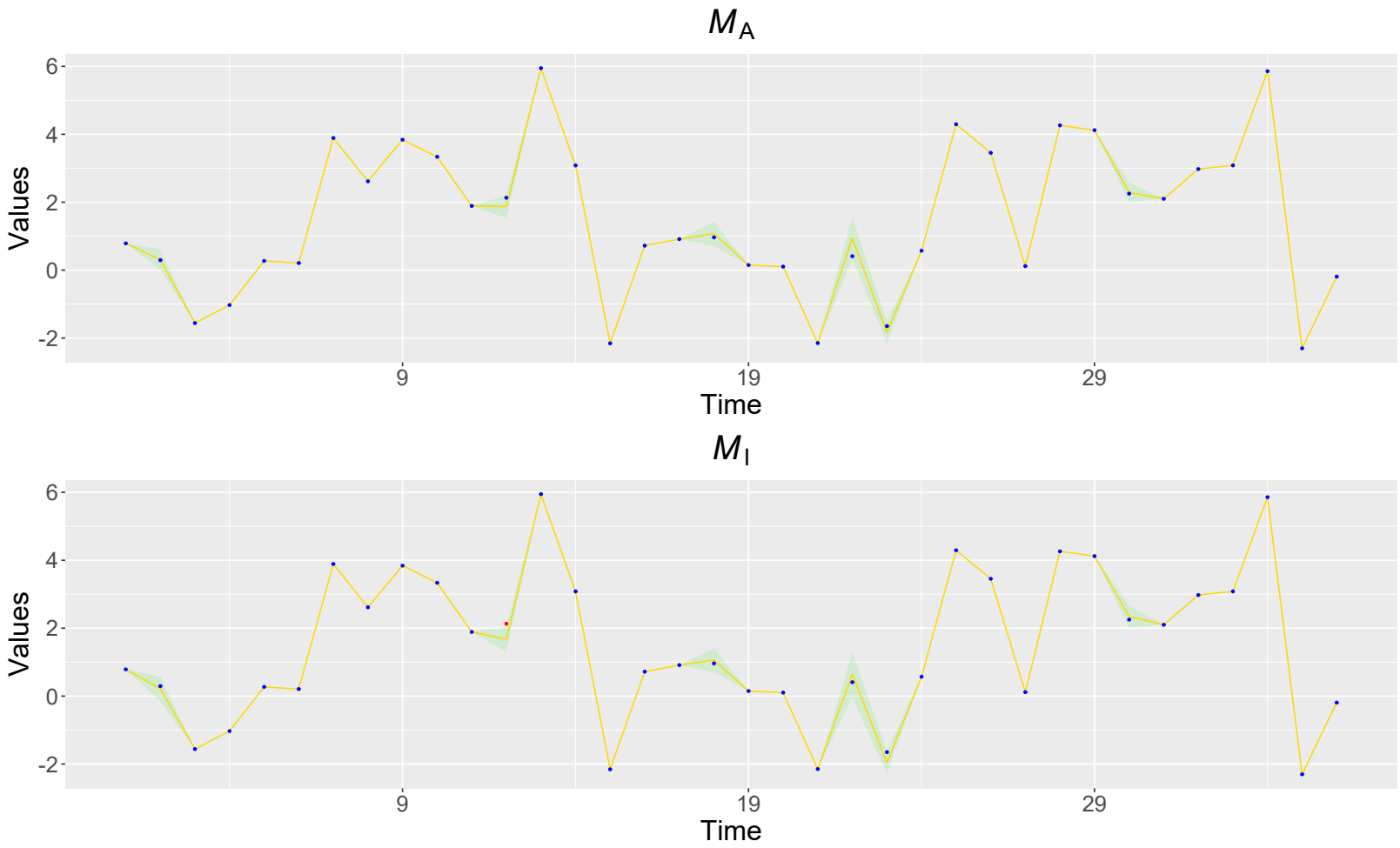


Figure 3.19: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 14$, $i = 2$ and $t \in \{1, \ldots, 36\}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

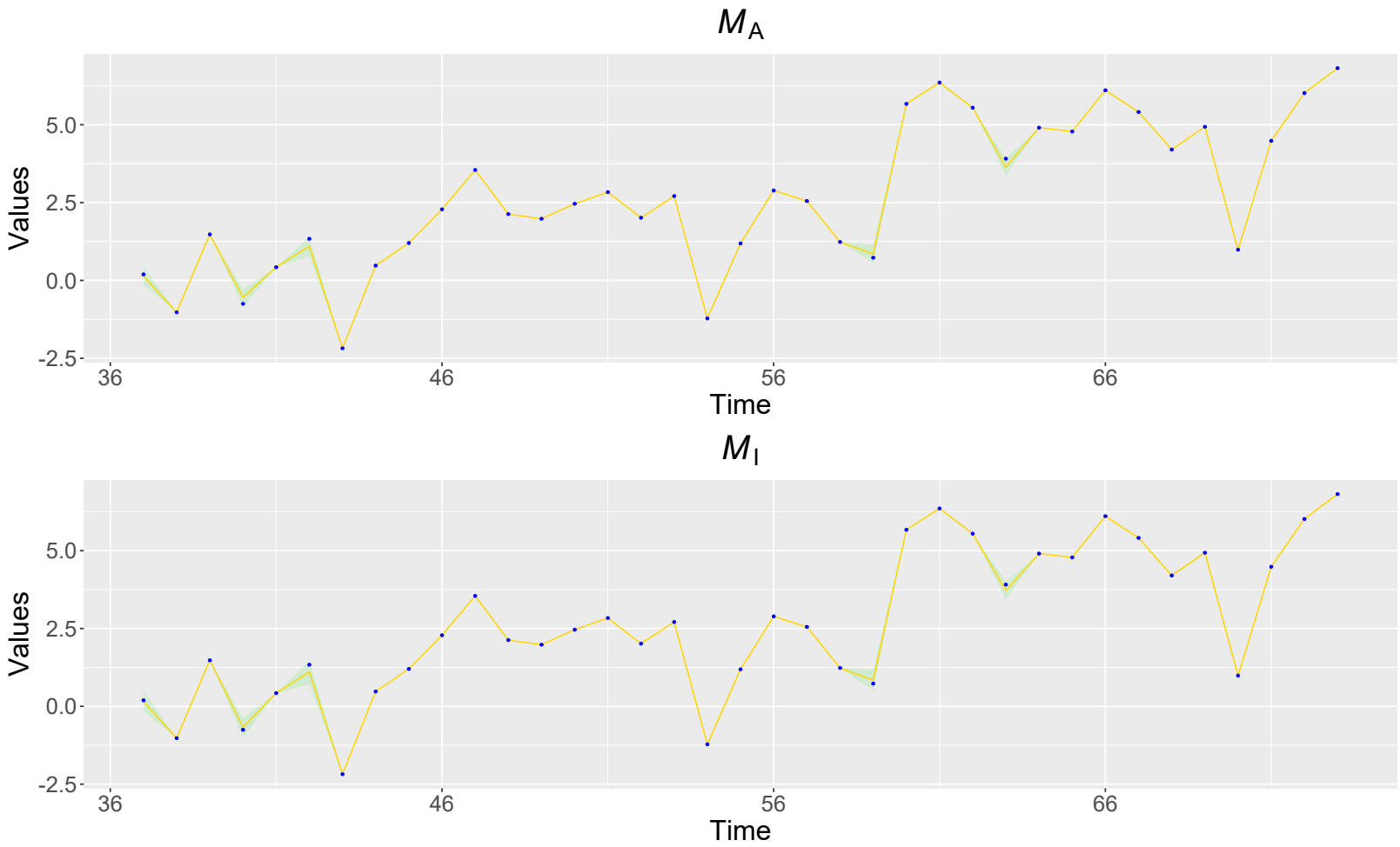


Figure 3.20: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 14$, $i = 2$ and $t \in \{37, \ldots, 73\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

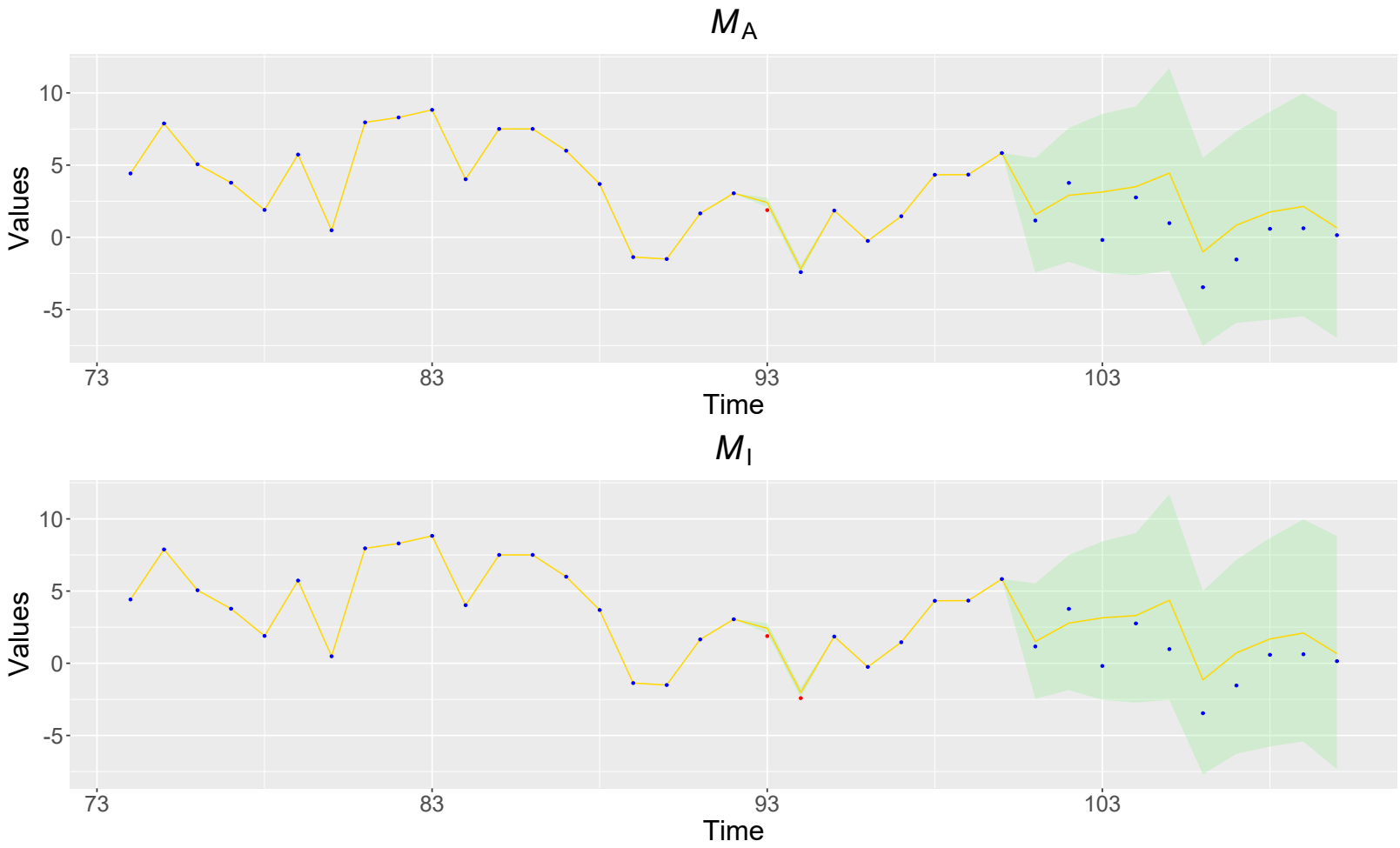
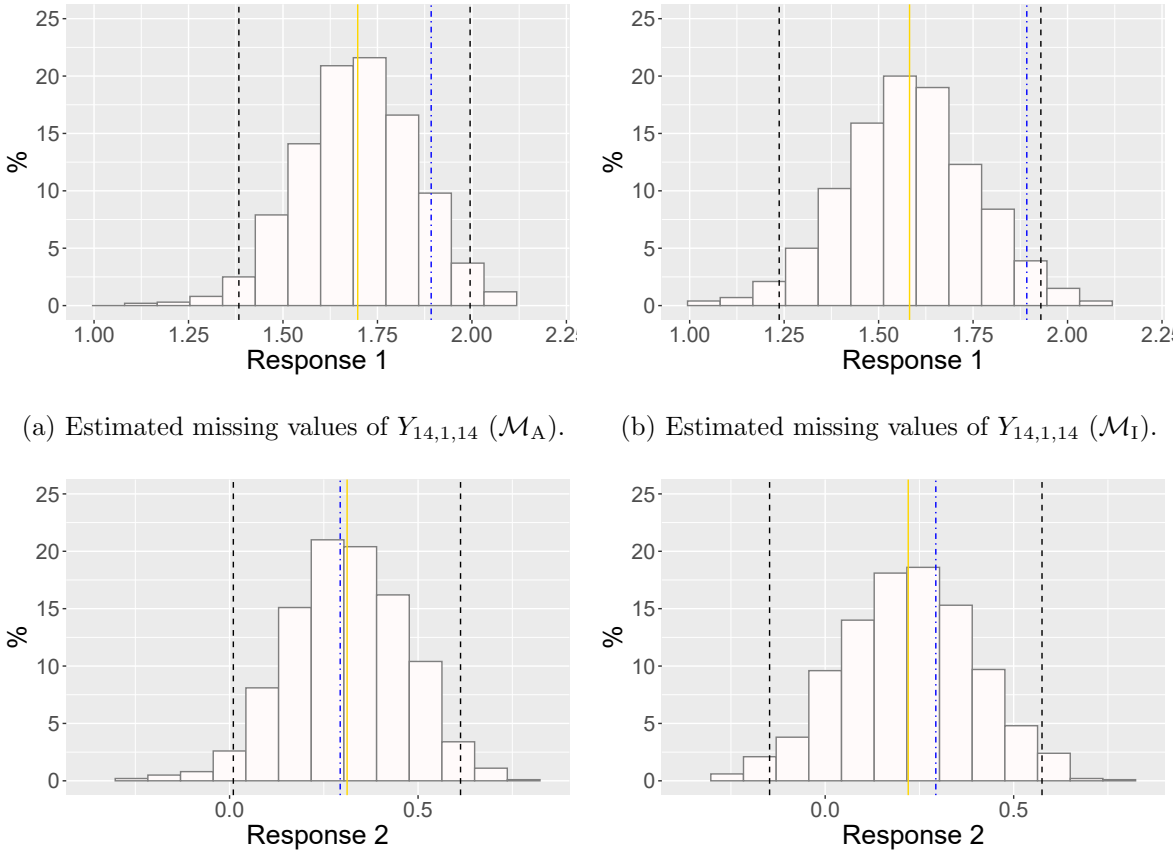


Figure 3.21: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 14$, $i = 2$ and $t \in \{74, \ldots, 110\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).



(a) Estimated missing values of $Y_{14,1,14}$ ($\mathcal{M}_{\mathrm{A}}$).

(b) Estimated missing values of $Y_{14,1,14}$ ($\mathcal{M}_{\mathrm{I}}$).

(c) Estimated missing values of $Y_{14,2,2}$ ($\mathcal{M}_{\mathrm{A}}$).

(d) Estimated missing values of $Y_{14,2,2}$ ($\mathcal{M}_{\mathrm{I}}$).

Figure 3.22: Histograms of the estimated missing values of $Y_{14,1,22}$ and $Y_{14,2,2}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

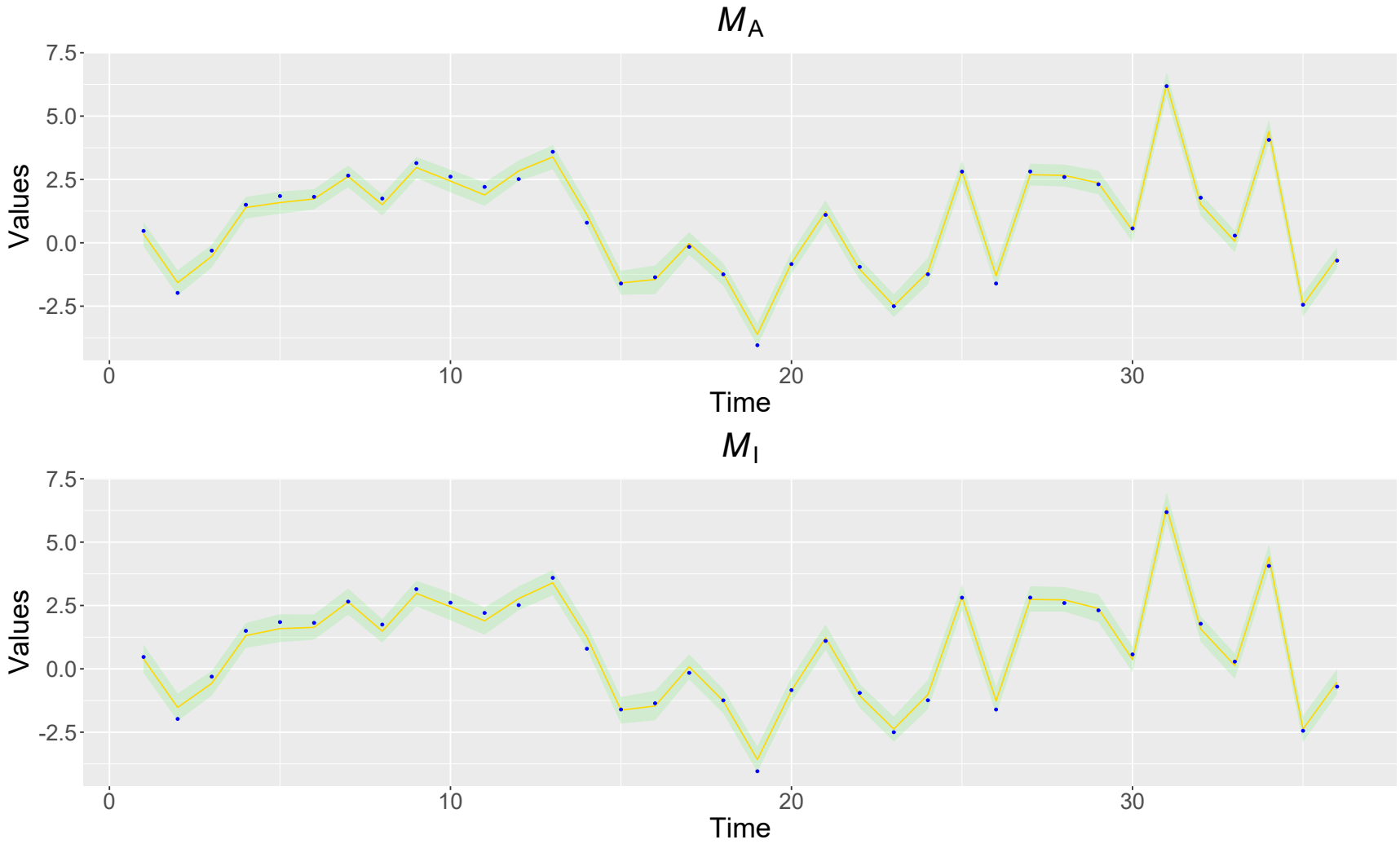


Figure 3.23: Line chart of the interpolated values $Y_{16+n,i,t}$ for $n = 3$, $i = 1$ and $t \in \{1, \ldots, 36\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

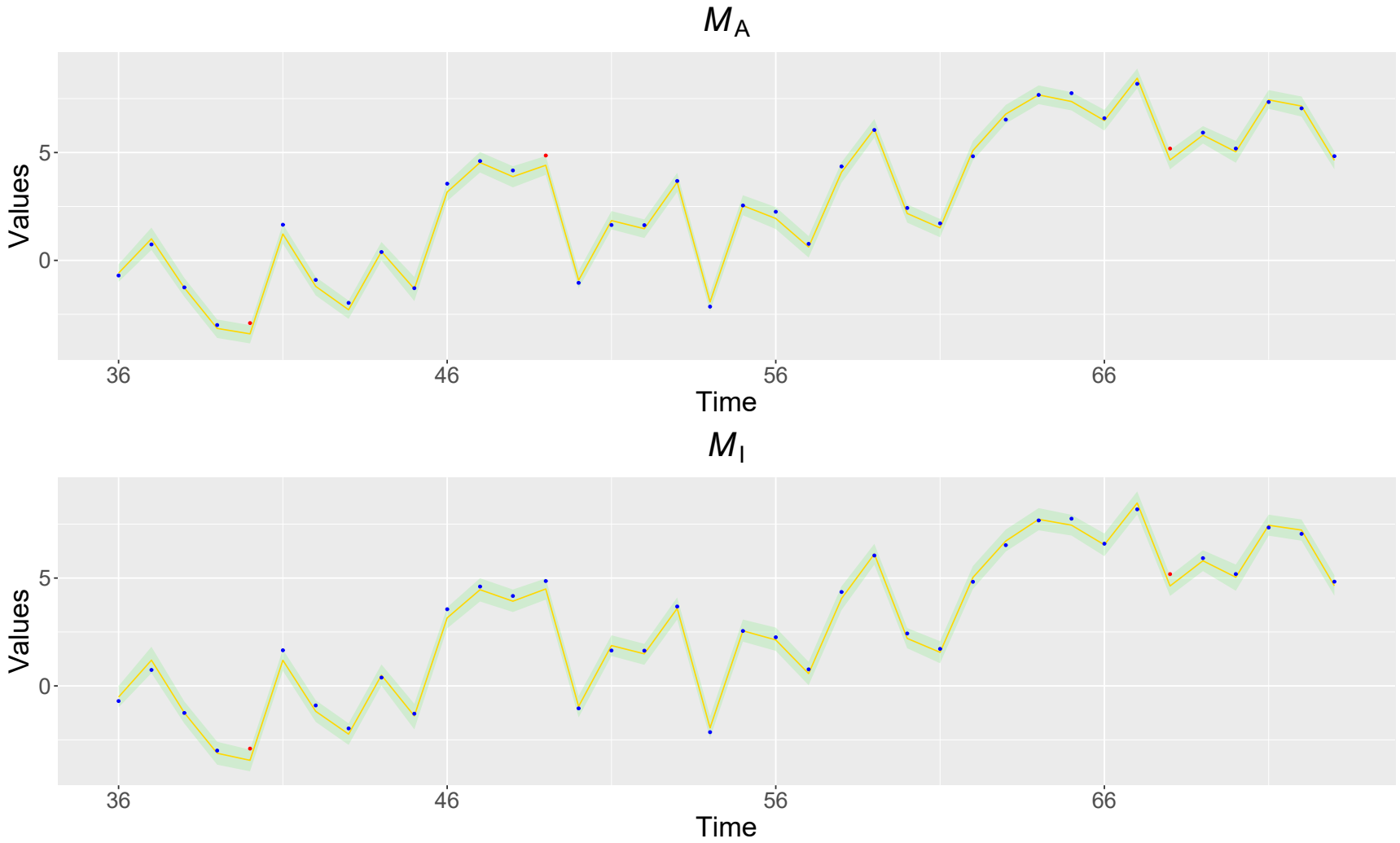


Figure 3.24: Line chart of the interpolated values $Y_{16+n,i,t}$ for $n = 3$, $i = 1$ and $t \in \{37, \ldots, 73\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

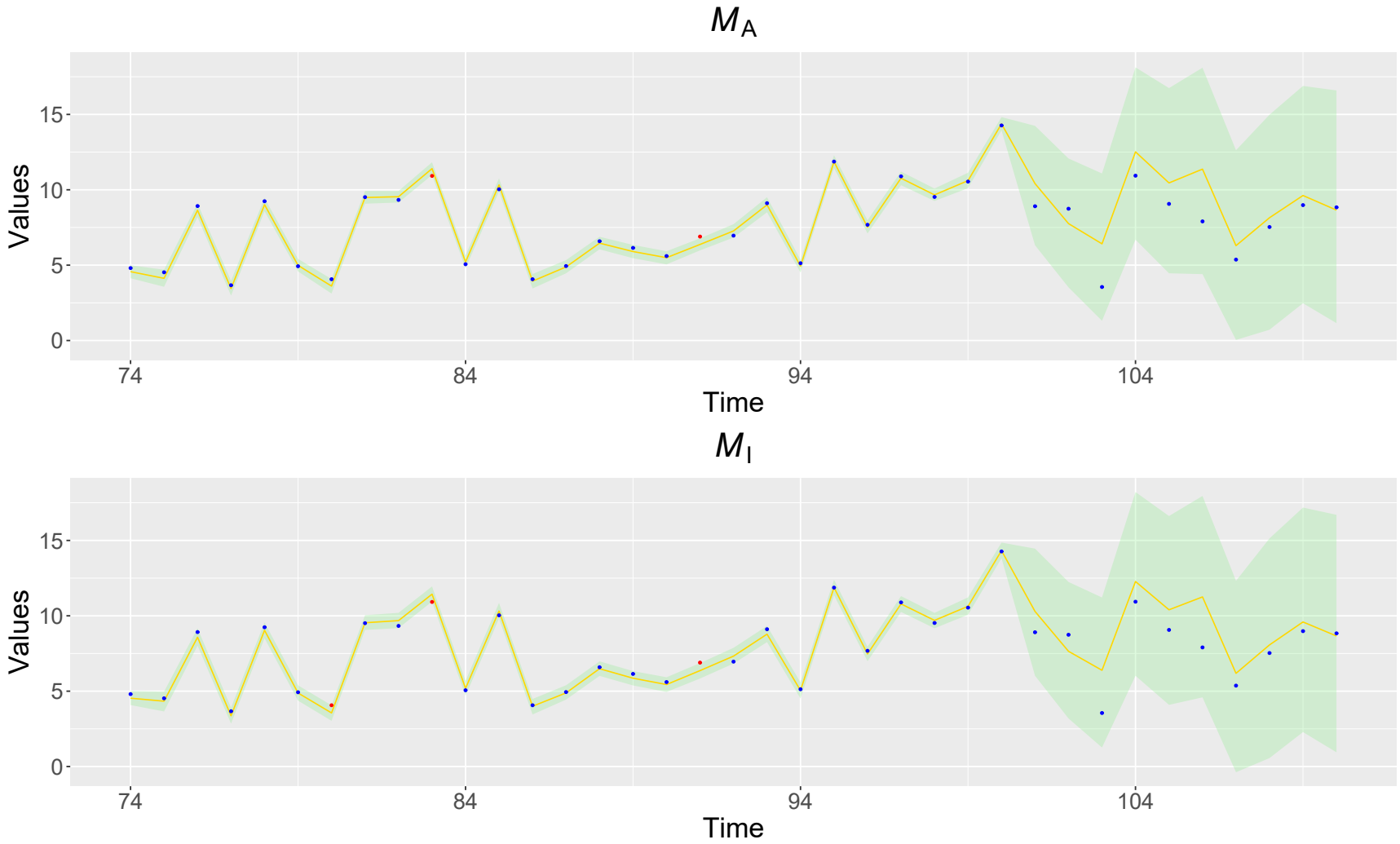


Figure 3.25: Line chart of the interpolated values $Y_{16+n,i,t}$ for $n = 3$, $i = 1$ and $t \in \{74, \ldots, 110\}$ for models $\mathcal{M}_A$ and $\mathcal{M}_I$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

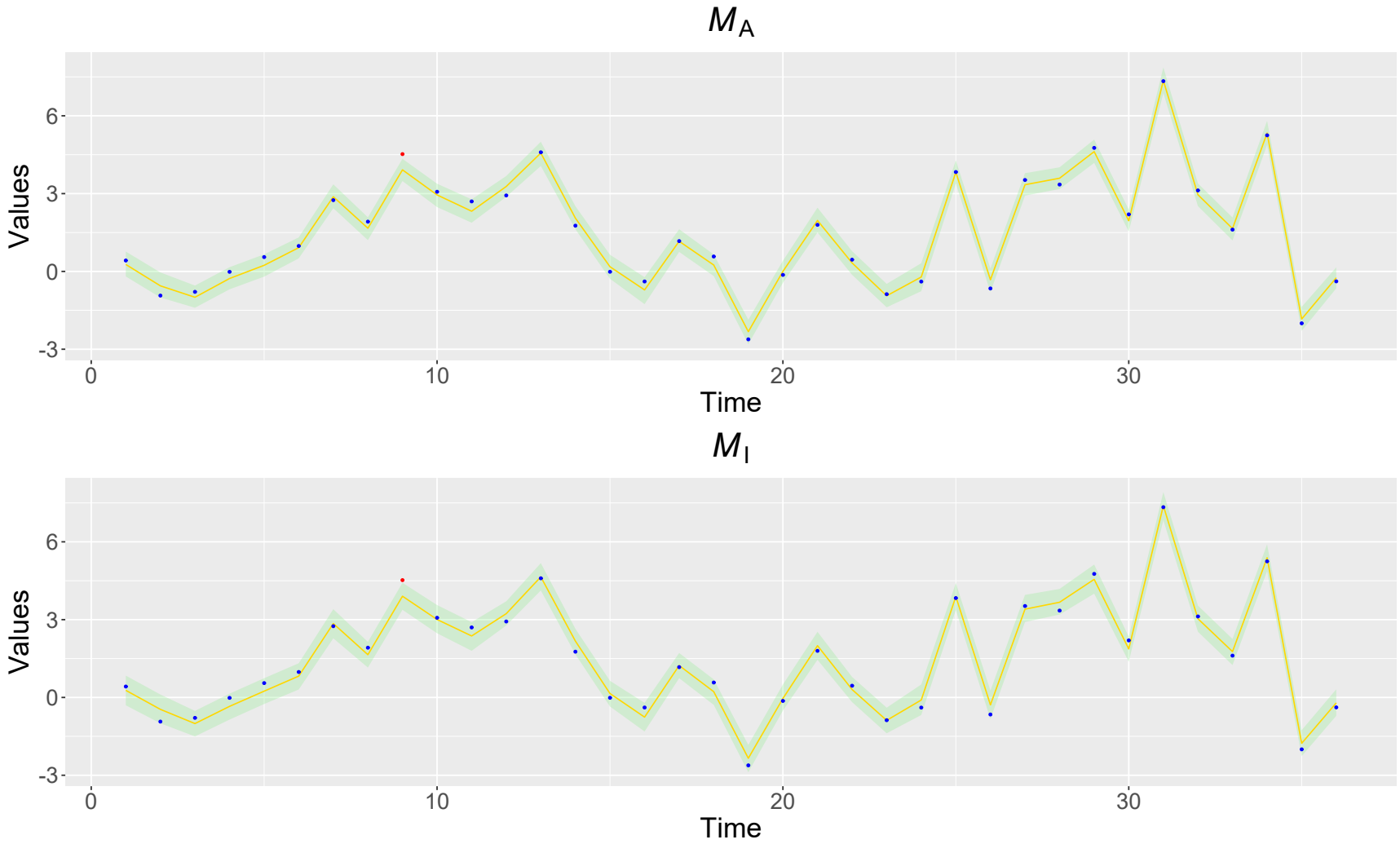


Figure 3.26: Line chart of the interpolated values $Y_{16+n,i,t}$ for $n = 3$, $i = 2$ and $t \in \{1, \ldots, 36\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

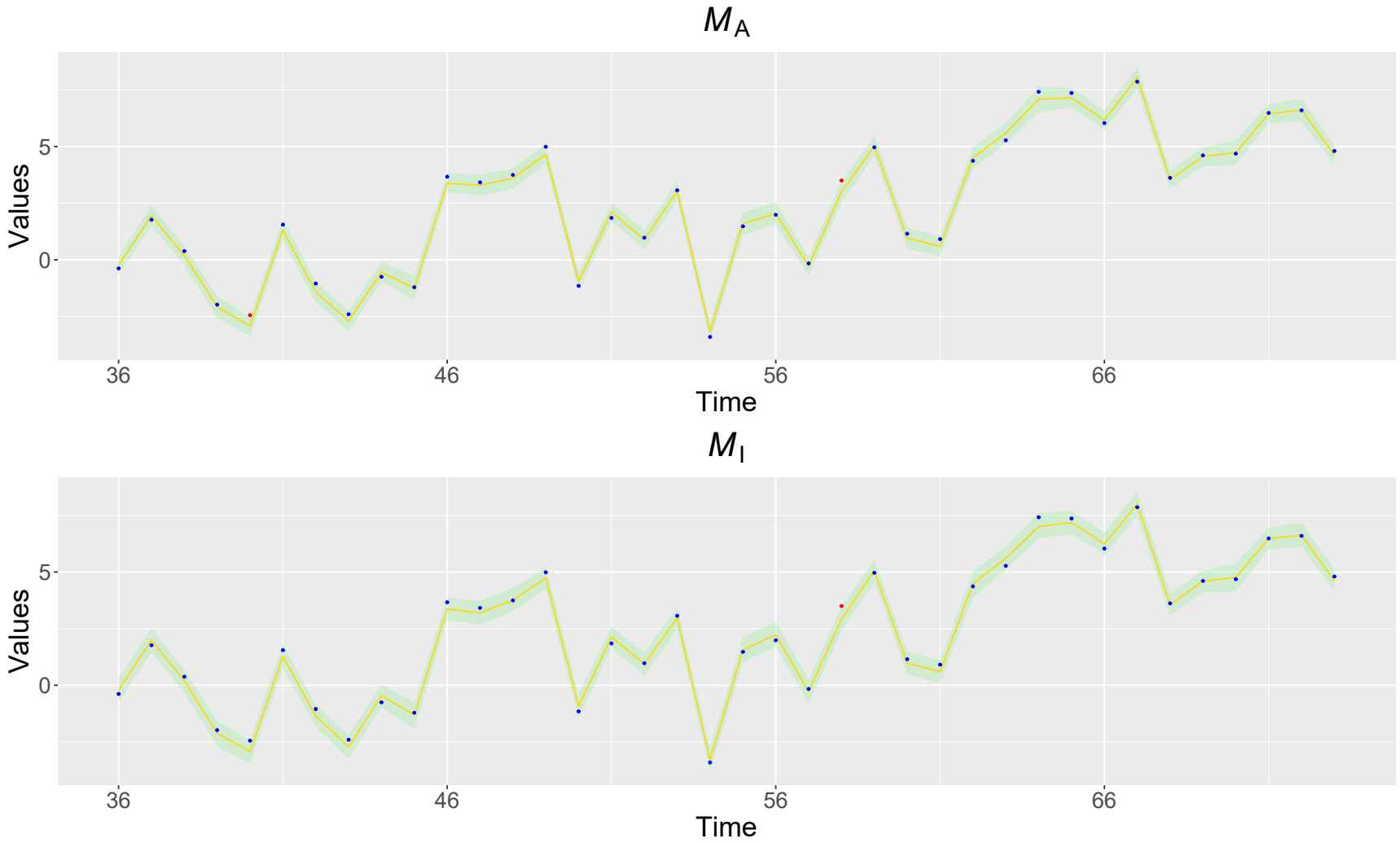


Figure 3.27: Line chart of the interpolated values $Y_{16+n,i,t}$ for $n = 3$, $i = 2$ and $t \in \{37, \ldots, 73\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

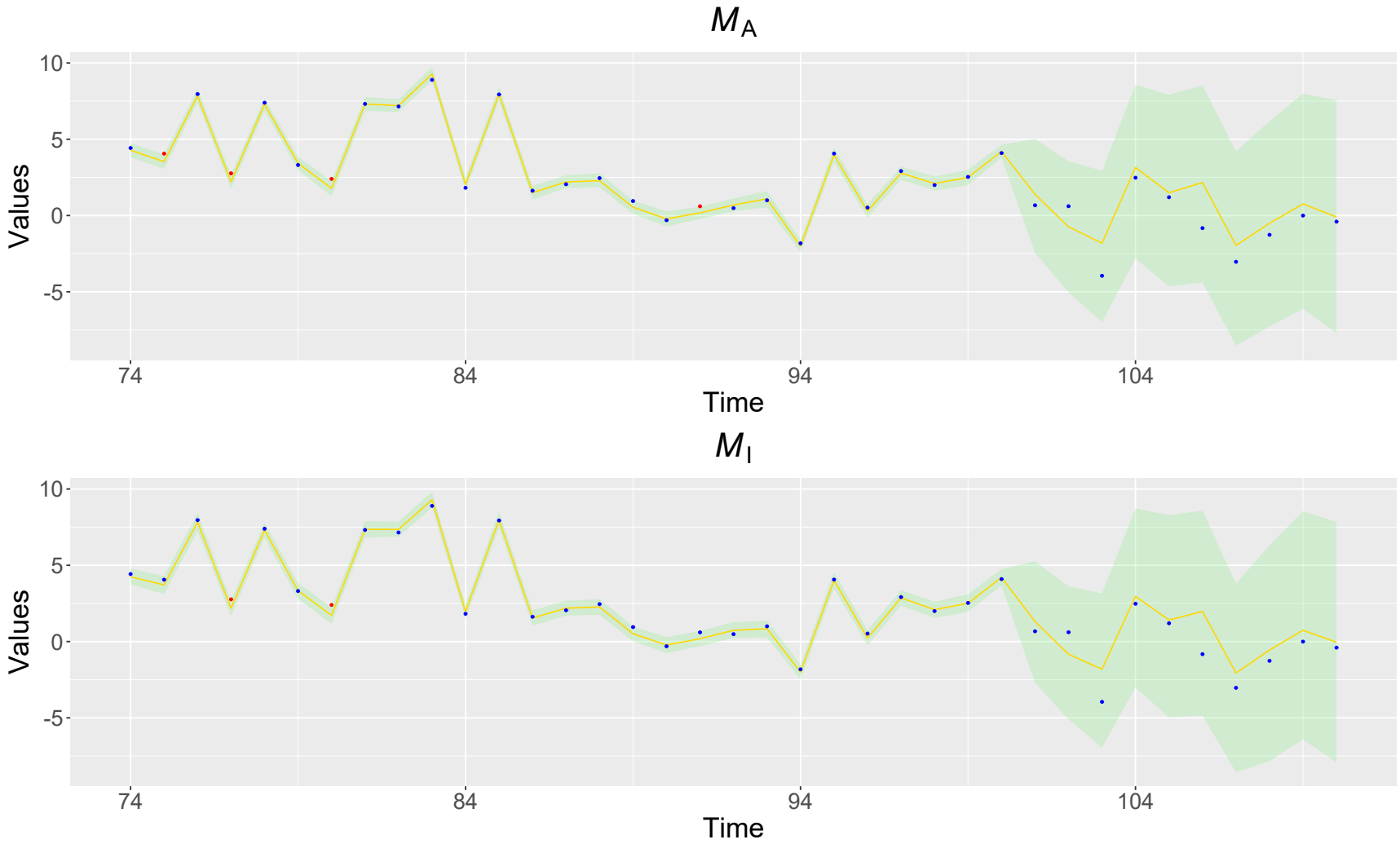
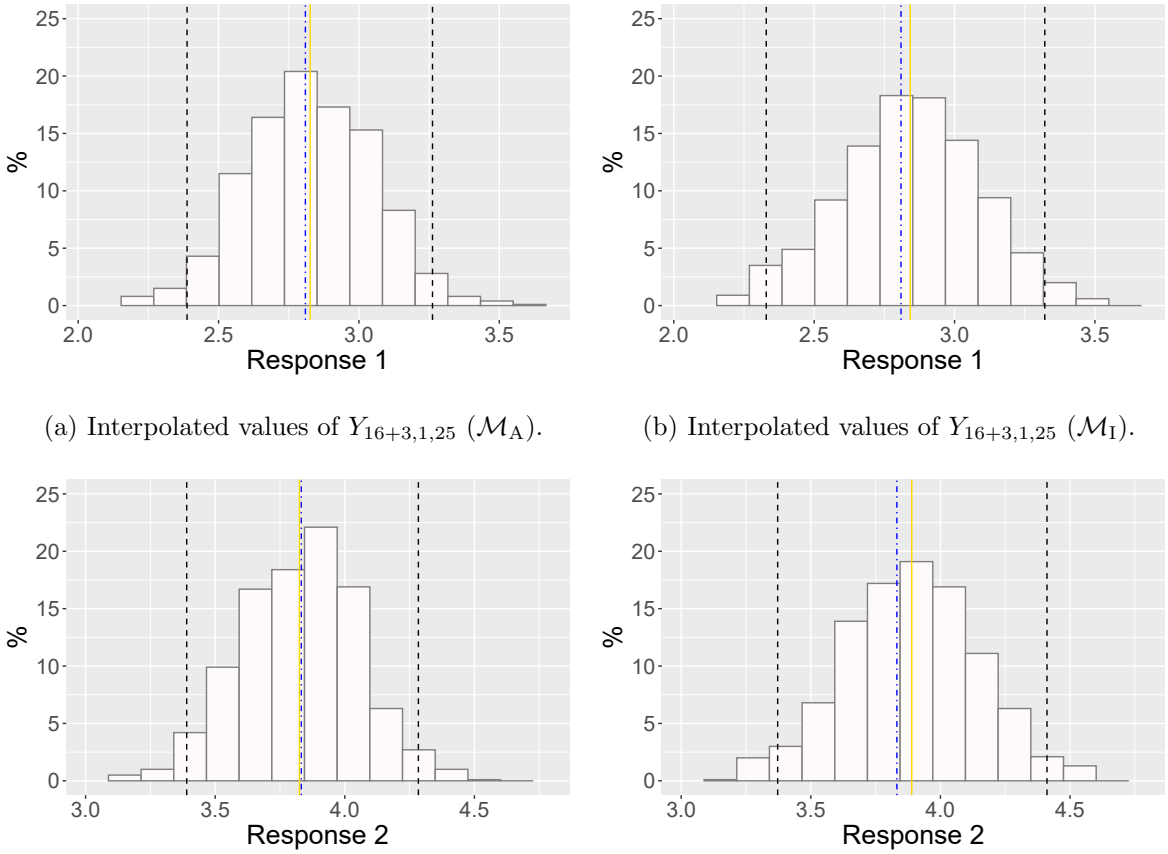


Figure 3.28: Line chart of the interpolated values $Y_{16+n,i,t}$ for $n = 3$, $i = 2$ and $t \in \{74, \ldots, 110\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).



(a) Interpolated values of $Y_{16+3,1,25}$ ($\mathcal{M}_{\mathrm{A}}$).

(b) Interpolated values of $Y_{16+3,1,25}$ ($\mathcal{M}_{\mathrm{I}}$).

(c) Interpolated values of $Y_{16+3,2,25}$ ($\mathcal{M}_{\mathrm{A}}$).

(d) Interpolated values of $Y_{16+3,2,25}$ ($\mathcal{M}_{\mathrm{I}}$).

Figure 3.29: Histograms of the interpolated values of $Y_{16+3,i,25}$ for $i \in \{1, 2\}$ and for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the second simulation study in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line. The true value is represented by the dot-dashed line (in blue, if within the range; or in red, otherwise).

3.6 Illustrative example

We will again analyze the real data set used in Section 2.6. The main change is that it is now no longer necessary to impute missing values in order to apply the statistical model.

We used $N = 10$ sites and the first $T = 360$ days to fit the model, storing $N^* = 2$ sites to investigate the interpolation performance (Figure 3.30(c) highlights these sites on the map) and the last $T^\star = 5$ days to study the forecasting performance. We also worked with no explanatory

variables ($p = 1$), considering $\mathbf{W} = \mathbf{I}_1$ and, for $t \in \{1, \ldots, 360\}$, $\mathbf{G}_t = \mathbf{I}_1$, $\mathbf{X}_t = \mathbf{1}_{10}$ and $\mathbf{X}_t^* = \mathbf{1}_2$.

To run Algorithm 11, we use the fixed quantities $\mathbf{W}$, $\mathbf{G}_t$, $\mathbf{X}_t$ and $\underset{\sim}{\mathbf{Y}}_{t,\mathsf{obs}}$ for $t = 1, \ldots, 360$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\boldsymbol{\Sigma}} = 0.001$, $\mathbf{b}_{\boldsymbol{\Sigma}} = 0.001 \cdot \mathbf{I}_2$, $a_\phi = 0.001$, $b_\phi = 0.001$, $\mathbf{M}_0 = \mathbf{0}_{1\times 2}$, $\mathbf{C}_0 = \mathbf{I}_1$, and $\boldsymbol{\sigma}_d^2$ as the empirical covariance matrix of the gauged sites. We choose $\psi = 0.50$ after a few tries.

Algorithm 6 was run 60000 times, having converged after $J \approx 10000$ iterations. The same configuration was adopted for sampling from the analogous isotropic model $\mathcal{M}_{\mathrm{I}}$, with $B_{n,n'} = \exp\{-\frac{1}{\phi}\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}$. To avoid autocorrelation in the chains, we form a sample of size $K = 1000$ of the posterior distribution of the parameters by systematically sampling every 50 iterations (i.e. $j_1 = 10001, j_2 = 10051, \ldots, j_{1000} = 59951$). The acceptance rates of the parameter ϕ are equal to 43.47% and 42.24%, respectively for the models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$. The trace plots for the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$, ϕ and $\mathbf{D}$ from this illustrative example are available in Appendix C.2.3.

The geographic region map and its corresponding estimated deformation[1] are shown in Figure 3.30. The deformed map has some folds and points with more shrinking distances. For models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, Figure 3.31 shows the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ and ϕ, while the posterior distributions of $\beta_{0,1,t}$ and $\beta_{0,2,t}$ $(t = 0, 1, \ldots, 365)$ are shown respectively in Figures 3.32 and 3.33. It can be noted that the posterior means of ϕ change a lot depending on the model, as well as that model $\mathcal{M}_{\mathrm{A}}$ provided wider credibility intervals than model $\mathcal{M}_{\mathrm{I}}$ for parameters $\beta_{0,1,t}$ and $\beta_{0,2,t}$.

Table 3.2 shows some metrics for comparing models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$. Based on the DIC and PMSE statistics, $\mathcal{M}_{\mathrm{A}}$ fits the data better than $\mathcal{M}_{\mathrm{I}}$. To facilitate visual comparison of missing data imputation, forecast and interpolation by model, we divided the time series into two intervals: 1-182 days and 183-365 days.

Figures 3.34-3.36 show the estimated missing values and predictions at the gauged site $\underset{\sim}{\mathbf{s}}_6$, for both response variables. The intervals obtained under model $\mathcal{M}_{\mathrm{A}}$ are narrower for imputing missing data, while they are wider for predicting future values.

The results were similar for both models, with a slight advantage for model $\mathcal{M}_{\mathrm{A}}$ in relation to slightly narrower intervals in imputation. Although Tables 2.3 and 3.2 do not present many differences in results, due to the low fraction of missing values in both variables, imputing missing data through the statistical model has the advantage of knowing the uncertainty through the

[1]The way we obtain the borders of the estimated deformed map is explained in Appendix A.5.

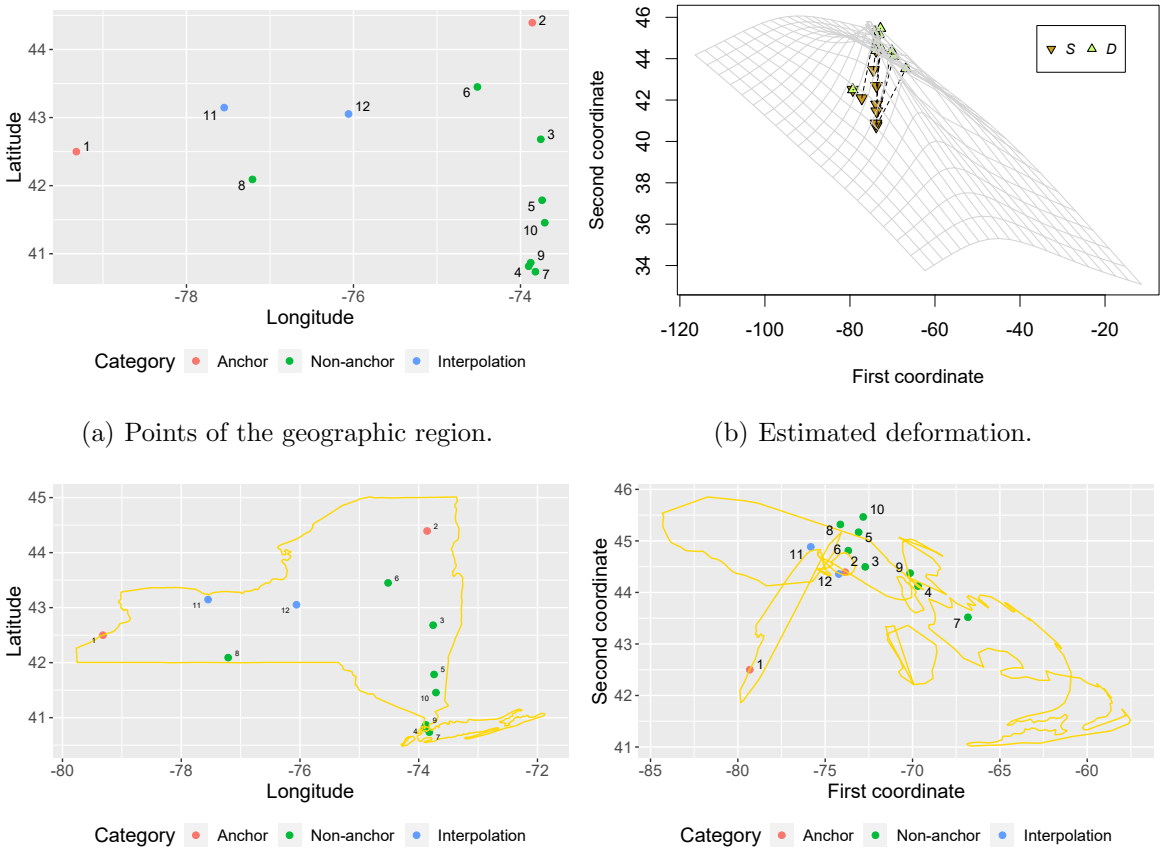
posterior distribution.

Figures 3.38-3.41 show that the interpolation performance of model $\mathcal{M}_{\mathrm{A}}$ is superior to the one of model $\mathcal{M}_{\mathrm{I}}$ at the ungauged site $\underset{\sim}{\mathbf{s}}_{12}$ for the second response variable, noting that their credibility intervals are narrower, which is in agreement with the results of the ECP and IS statistics. The imputation results obtained with both models were quite similar for the first response variable.

Thus, through this illustrative example we concluded that, to analyze real data with some anisotropy, model $\mathcal{M}_{\mathrm{A}}$ (with spatial deformation) makes more assertive interpolation than model $\mathcal{M}_{\mathrm{I}}$ (without spatial deformation) and less assertive forecasts than model $\mathcal{M}_{\mathrm{I}}$.

Table 3.2: Metrics for model comparison (DIC, PMSE, ECP and IS) by response variable and ungauged site, from the illustrative example in Chapter 3.

Metric	$\mathcal{M}_{\mathrm{A}}$		$\mathcal{M}_{\mathrm{I}}$	
DIC	-19081.0		-17422.2	
PMSE	0.04829		0.05643	
ECP (%)	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{11}$	99.7	99.4	97.2	99.4
$\underset{\sim}{\mathbf{s}}_{12}$	99.2	100.0	98.6	99.7
IS	**Response 1**	**Response 2**	**Response 1**	**Response 2**
$\underset{\sim}{\mathbf{s}}_{11}$	0.00068	0.07056	0.00071	0.08400
$\underset{\sim}{\mathbf{s}}_{12}$	0.00066	0.06278	0.00064	0.08307



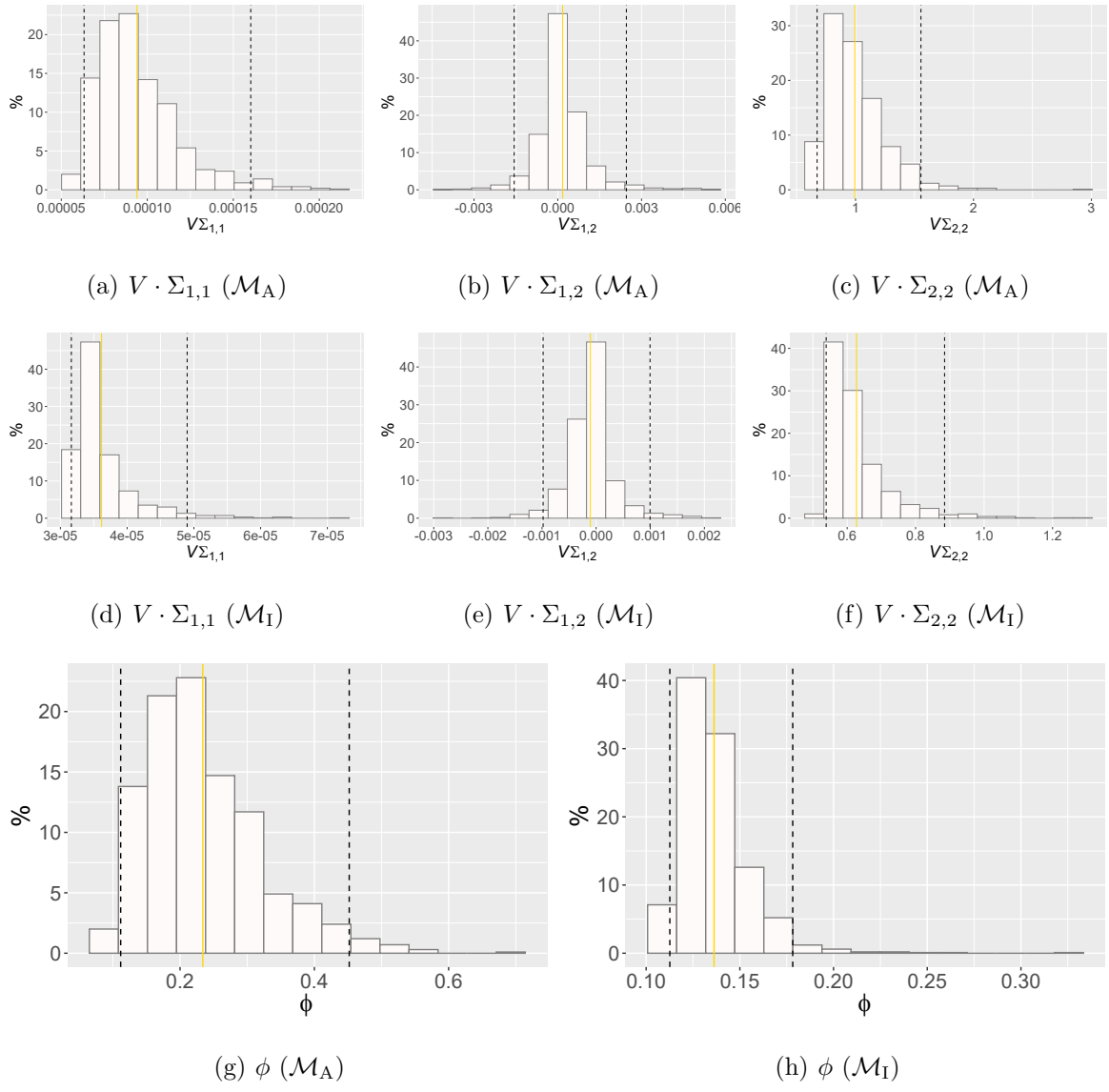
(a) Points of the geographic region.

(b) Estimated deformation.

(c) Geographic region map.

(d) Estimated deformed map.

Figure 3.30: Geographic region map (State of New York, US), estimated deformed map and their sites, resulting from the illustrative example in Chapter 3.



(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$)

(g) ϕ ($\mathcal{M}_{\mathrm{A}}$) (h) ϕ ($\mathcal{M}_{\mathrm{I}}$)

Figure 3.31: Histograms of the posterior distributions of ϕ and $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2\})$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

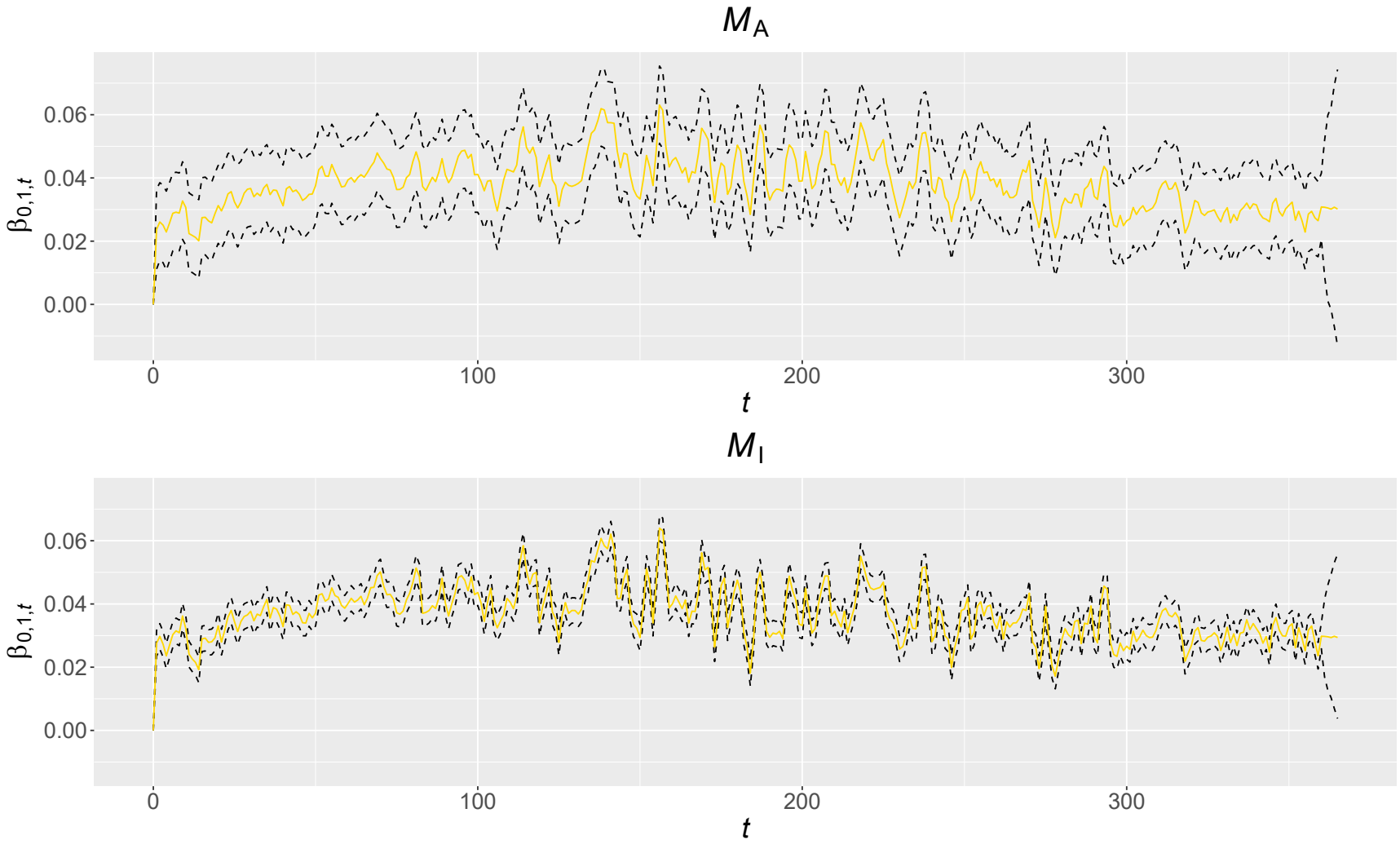


Figure 3.32: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

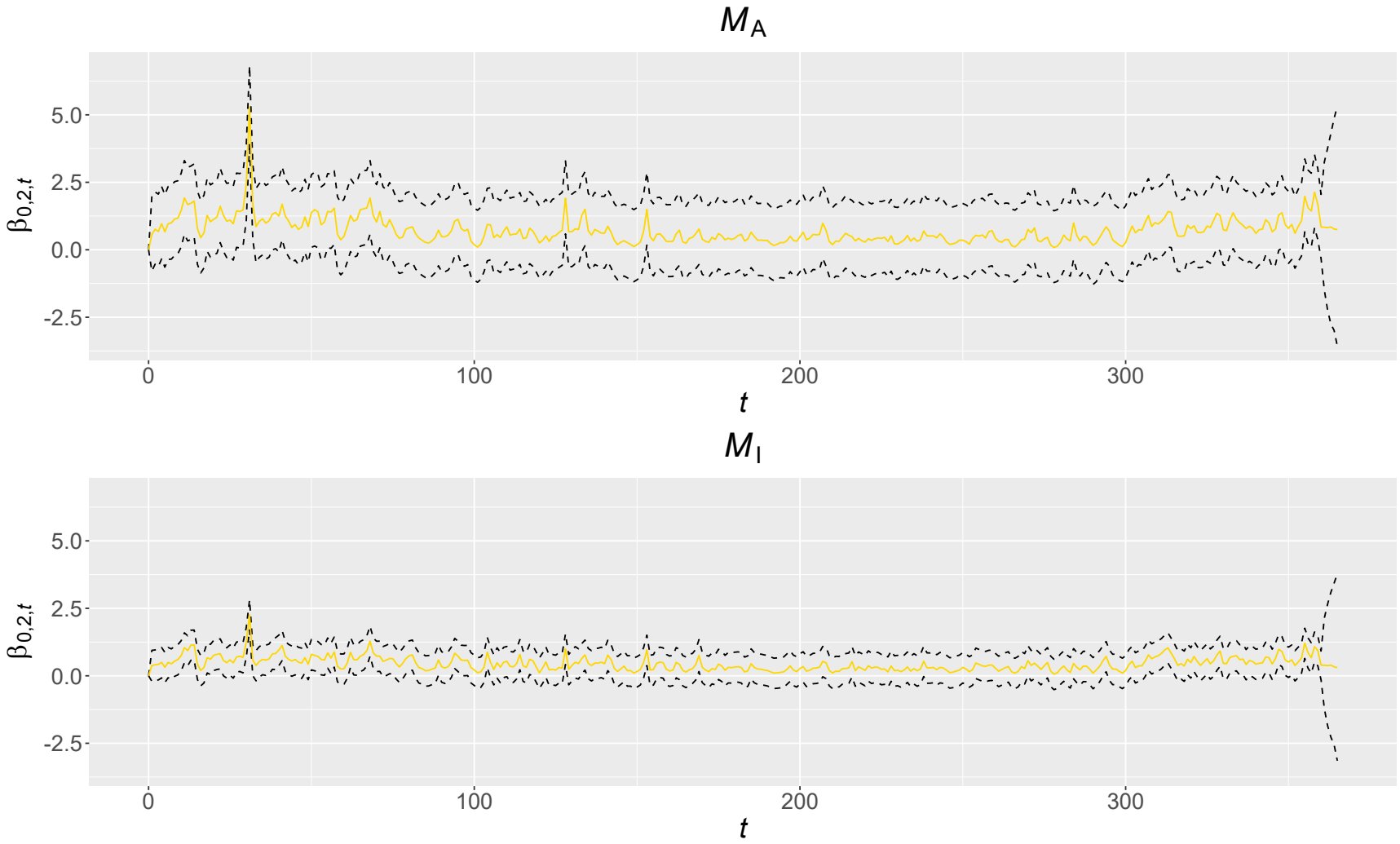


Figure 3.33: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

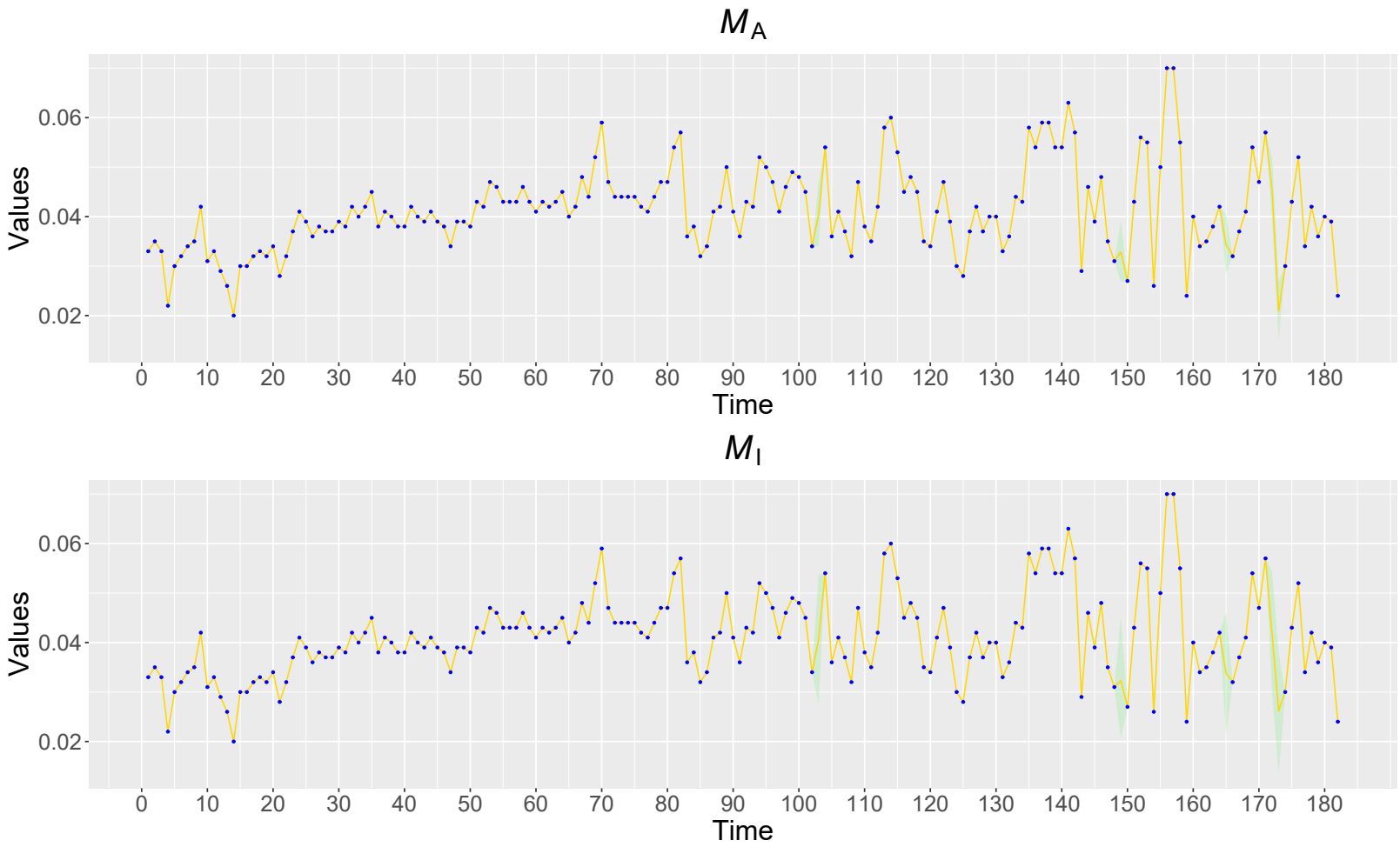


Figure 3.34: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 6$, $i = 1$ and $t \in \{1, \ldots, 182\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

Figure 3.35: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 6$, $i = 1$ and $t \in \{183, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

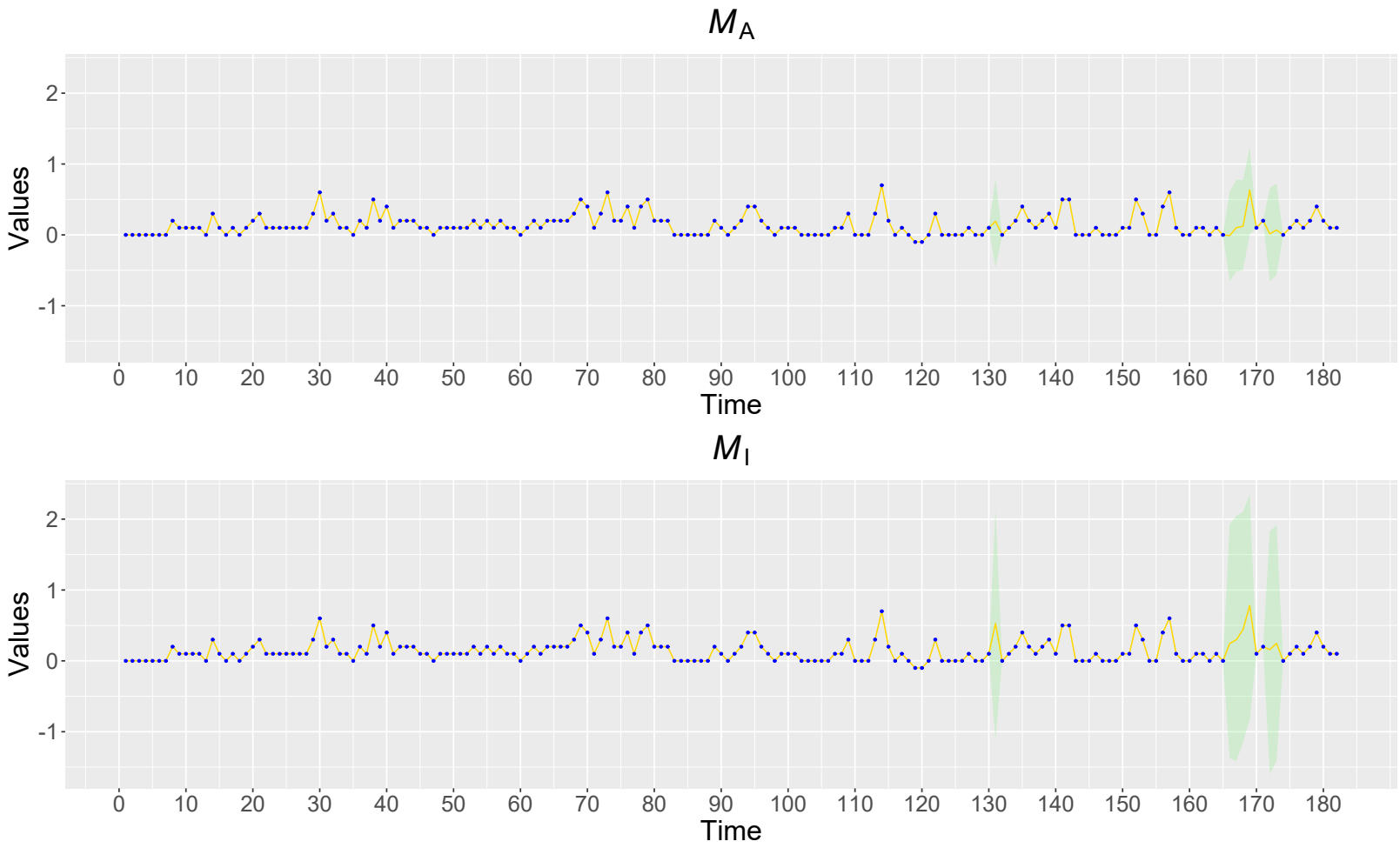


Figure 3.36: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 6$, $i = 2$ and $t \in \{1, \ldots, 182\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

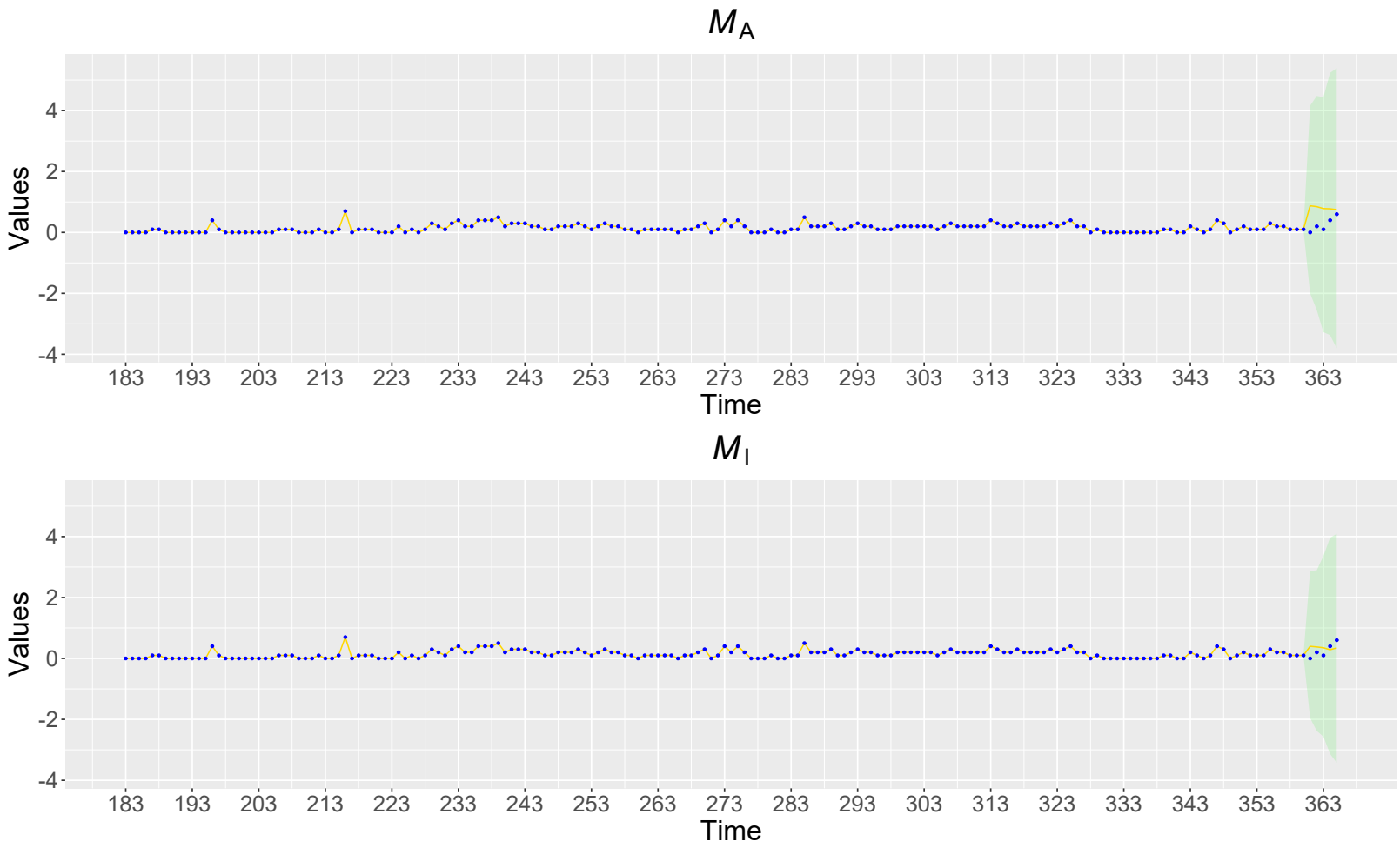


Figure 3.37: Line chart of the posterior distribution of $Y_{n,i,t}$ for $n = 6$, $i = 2$ and $t \in \{183, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area. Points represent the true values (in blue, if within range; in red, otherwise). The solid gold line connects the posterior means (if there is a range) and the true values (if not).

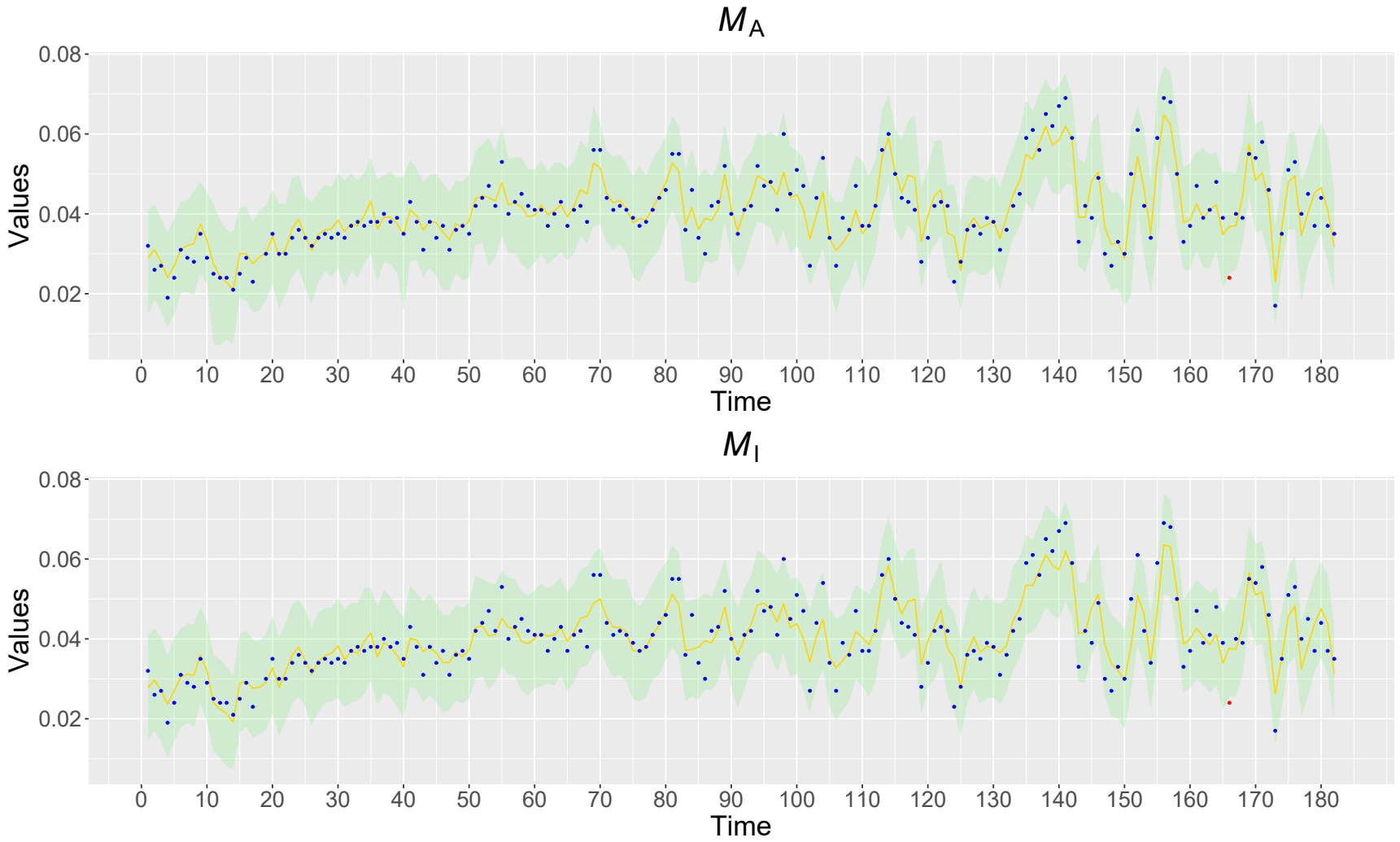


Figure 3.38: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 1$ and $t \in \{1, \ldots, 182\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

Figure 3.39: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 1$ and $t \in \{183, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

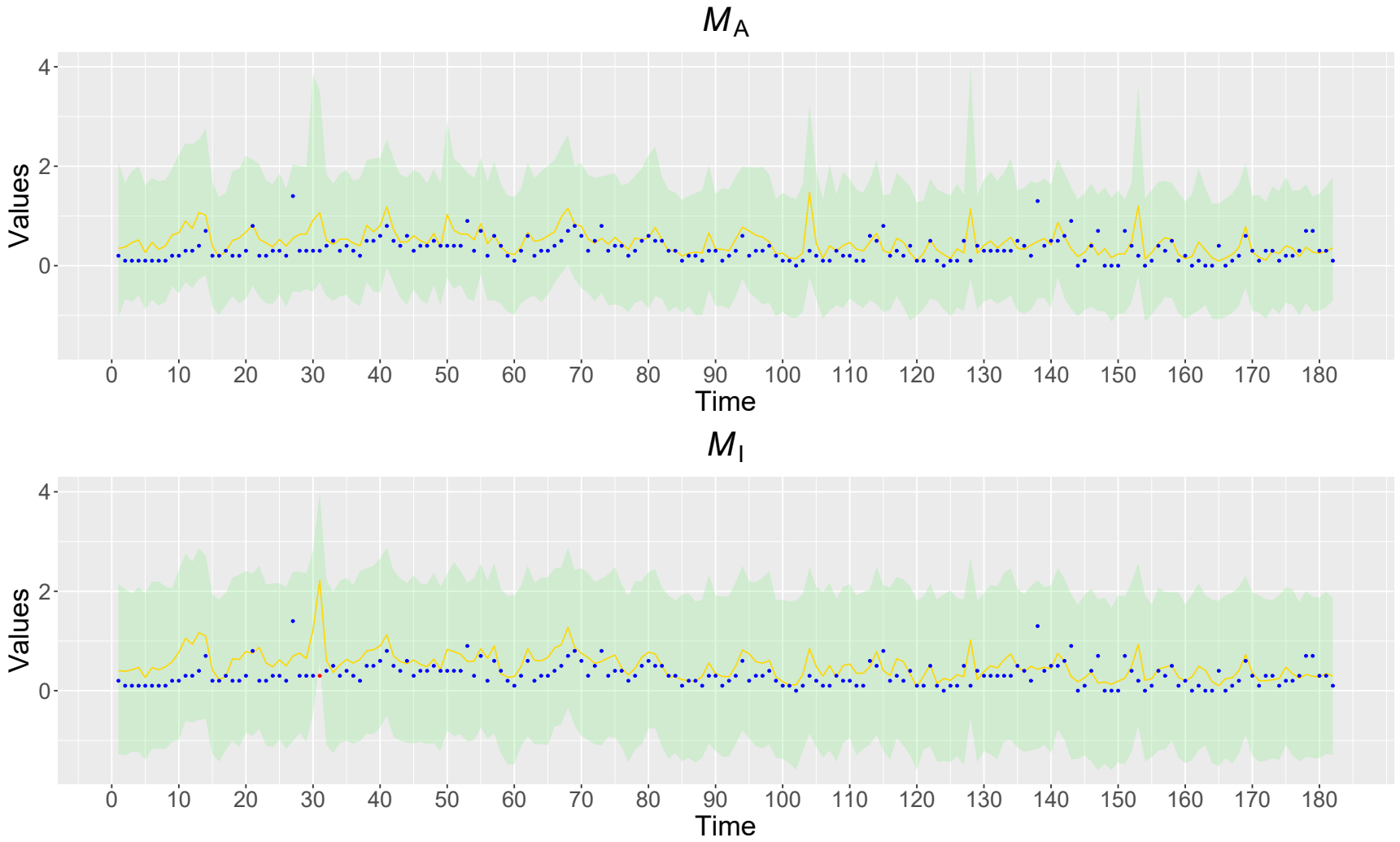


Figure 3.40: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 2$ and $t \in \{1, \ldots, 182\}$ for models $\mathcal{M}_A$ and $\mathcal{M}_I$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

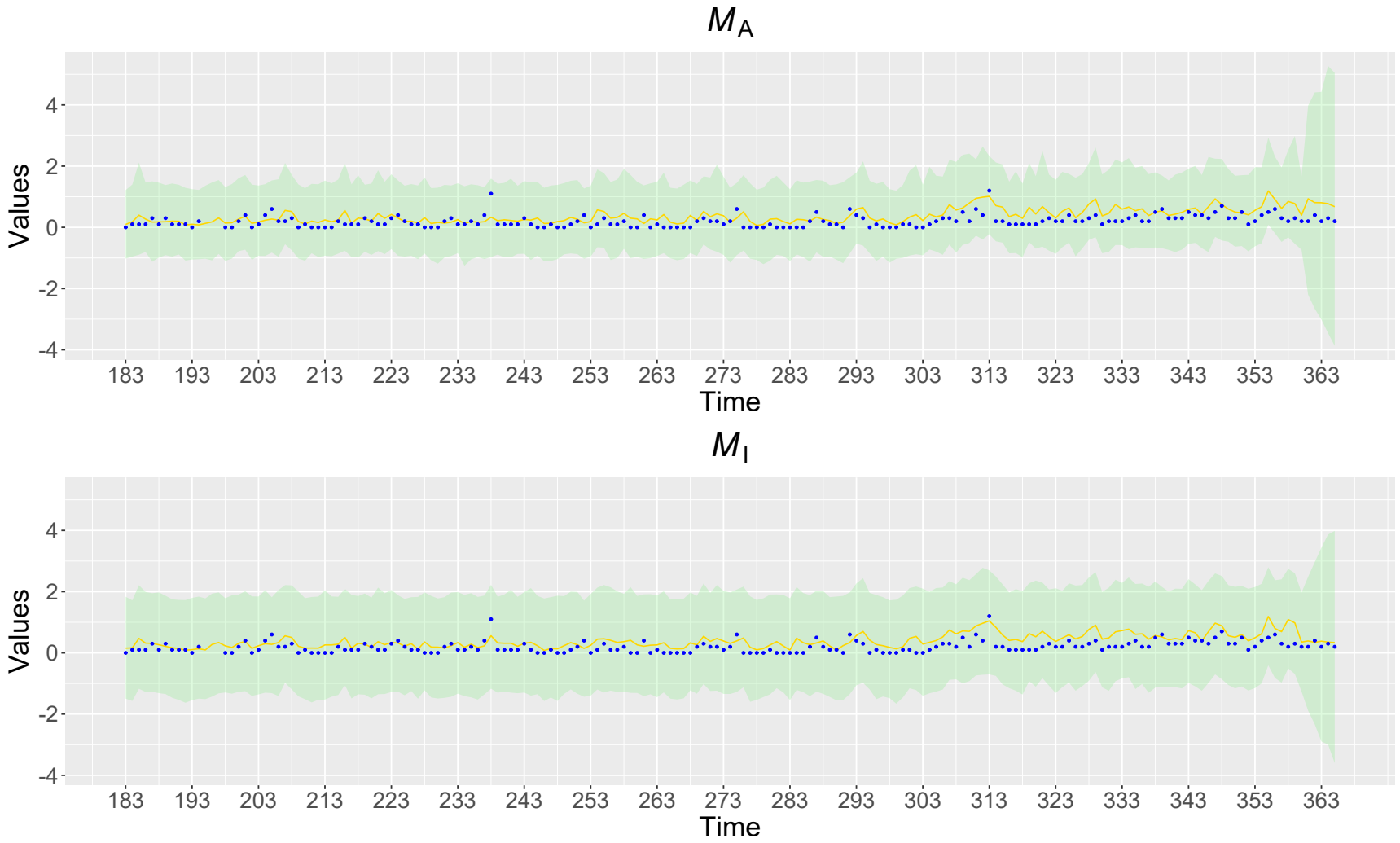


Figure 3.41: Line chart of the interpolated values of $Y_{10+n,i,t}$ for $n = 2$, $i = 2$ and $t \in \{183, \ldots, 365\}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$, resulting from the illustrative example in Chapter 3. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

3.7 Final considerations

In this chapter, we present an extension to the model proposed in Chapter 2 that accommodates incomplete response matrices. The case in which it is assumed that missing values are considered MCAR – Missing Completely at Random (Little and Rubin, 2019, Sec. 1.3).

As a main finding, we verified that the incorporation of spatial deformation brings advantages in the imputation of missing values and in interpolation. However, we saw large fractions of missing values can harm the quality of parameter estimates.

In the illustrative example discussed in Section 3.6, the appearance of missing values does not follow an MCAR pattern. Despite this, the results obtained seem reasonable when using the model with spatial deformation. Another advantage of estimating missing values using the statistical model is the possibility of quantifying their uncertainty, which is no longer possible when this occurs prior to modeling.

Chapter 4

An application with $q = 3$ response variables

The proposed model ($\mathcal{M}_{\text{A}}$) and its isotropic version ($\mathcal{M}_{\text{I}}$), given by Equations (3.1) and (3.7), respectively, were applied to a data set on the concentration levels of the following $q = 3$ air quality indices or response variables: Response 1 – Nitrogen dioxide (NO_2), Response 2 – Ozone (O_3), and Response 3 – Particulate matter 10 micrometers or less in diameter (PM_{10}). They were measured in the Northern region of Portugal, having been obtained at ten monitoring stations that measure the three pollutants (stations that measure only one or two of the three indices were not considered), collected hourly between June 1$^{\text{st}}$, 2020 and May 31$^{\text{st}}$, 2022. We computed the daily median, working with 730 days. The percentages of missing values for Responses 1, 2 and 3 are equal to 63.3%, 39.0% and 58.5%, respectively. This data set was obtained using the *get_saq_observations* command in the R package `saqgetr` (Grange, 2019).

We used $N = 9$ sites and the first $T = 725$ days to fit the model, storing $N^* = 1$ site to investigate the interpolation performance and the last $T^\star = 5$ days to study the forecasting performance. Figure 4.1 highlights these sites on the map[1]. We also worked with no explanatory variables ($p = 1$), considering $\mathbf{W} = \mathbf{I}_1$ and, for $t \in \{1, \ldots, 725\}$, $\mathbf{G}_t = \mathbf{I}_1$, $\mathbf{X}_t = \mathbf{1}_9$ and $\mathbf{X}_t^* = \mathbf{1}_1$. To run Algorithm 11, we use the fixed quantities $\mathbf{W}$, $\mathbf{G}_t$, $\mathbf{X}_t$ and $\underset{\sim}{\mathbf{Y}}_{t,\text{obs}}$ for $t = 1, \ldots, 725$. In addition, we specify the following hyperparameters: $a_V = 0.001$, $b_V = 0.001$, $a_{\boldsymbol{\Sigma}} = 0.001$, $\mathbf{b}_{\boldsymbol{\Sigma}} = 0.001 \cdot \mathbf{I}_3$, $a_\phi = 0.001$, $b_\phi = 0.001$, $\mathbf{M}_0 = \mathbf{0}_{1\times 3}$, and $\mathbf{C}_0 = \mathbf{I}_1$. To evaluate the importance of a good specification for $\boldsymbol{\sigma}_d^2$, we also consider two scenarios for the anisotropic model:

[1]The source we used to obtain the Portugal map shapefile (by region) is: https://github.com/joaopalmeiro/portugal-maps/tree/master/shapefile/ccdr_portugal_continental

- **Scenario 1.** $\boldsymbol{\sigma}_d^2 = \widehat{\text{Cov}}(\mathbf{S}) = \left[\begin{smallmatrix} 0.35649469 & 0.06978237 \\ 0.06978237 & 0.01959628 \end{smallmatrix}\right]$, choosing $\psi = 25$ after a few tries. This is indicated when the researcher does not have tools or information to specify $\boldsymbol{\sigma}_d^2$. Based on the gauged sites $(\underset{\sim}{\mathbf{s}}_1, \ldots, \underset{\sim}{\mathbf{s}}_9)$ in Figure 4.1, note that the longitudes vary between -8.721 and -6.887, while the latitudes vary between 41.097 and 41.572. Due to this difference in ranges, the sample variance of latitudes is smaller than the sample variance of longitudes. Viewing the map in Figure 4.1 as a scatter plot, the non-zero sample covariance is due to the appearance of a positive association between longitudes and latitudes; and

- **Scenario 2.** $\boldsymbol{\sigma}_d^2 = \left[\begin{smallmatrix} 0.10 & 0.00 \\ 0.00 & 0.01 \end{smallmatrix}\right]$, choosing $\psi = 40$ after a few tries. This specification was made arbitrarily, with the aim of evaluating the impact of using a non-judicious fixation for this hyperparameter.

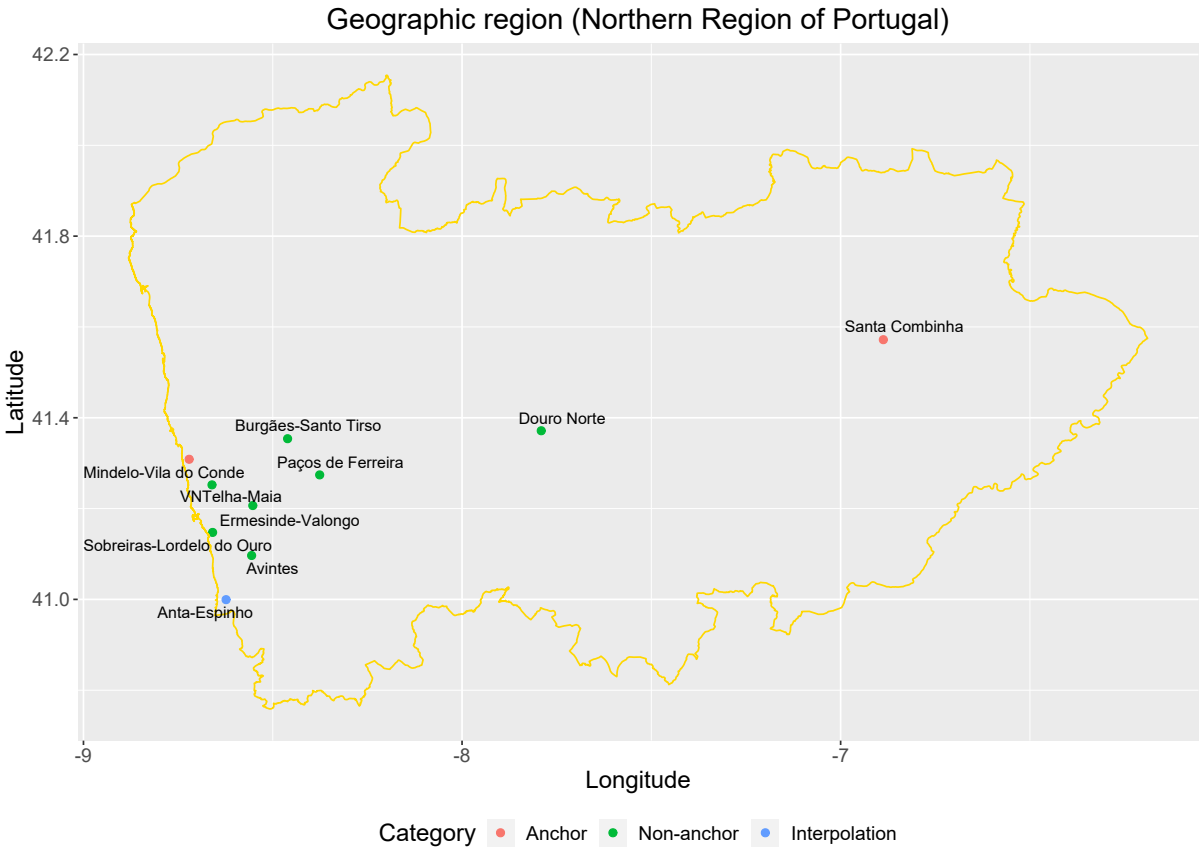


Figure 4.1: Sites in the geographic region of interest ($\mathcal{S}$) of the practical example in Chapter 4.

Algorithm 11 was run 18000 times, having converged after only $J \approx 3000$ iterations in Scenario 1. Analogously, it was run 20000 times and achieved convergence after $J \approx 10000$ iterations in Scenario 2. Similar configuration was adopted for sampling from the analogous isotropic model $\mathcal{M}_{\text{I}}$

given in Equation (3.7), where we specified $B_{n,n'} = \exp\{-\frac{1}{\phi}\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|\}$.

Figures 4.2(a) and 4.2(b) show the distances between the gauged sites and the respective posterior means of the deformations for Scenarios 1 and 2 of the anisotropic model, while Figures 4.2(c) and 4.2(d) show the estimated deformations of the respective Scenarios 1 and 2 of the anisotropic model. Deformed maps will not be displayed because the edges of the geographic map of the region of interest have 26547 pairs of coordinates, which makes the matrix inversion and multiplication calculations described in Appendix A.5 challenging. Both scenarios present more widespread deformations between them, noting that this is visibly more evident in Scenario 1. Based on the model comparison metrics presented in Table 4.1, it can be seen that the anisotropic model in Scenario 1 is more advantageous than the isotropic model and the anisotropic model in Scenario 2. The anisotropic model in Scenario 2 is less advantageous than the isotropic model, which reinforces the need for adequate specifications for hyperparameters $\boldsymbol{\sigma}_d^2$ and ψ. Because of this, we will no longer display analyzes with Scenario 2 of the anisotropic model.

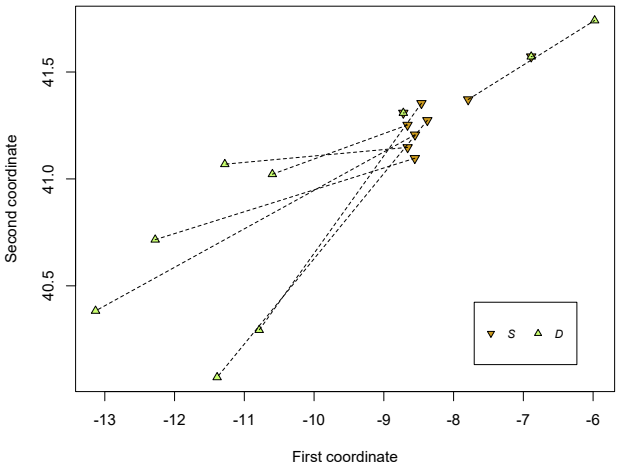
The acceptance rates of the parameter ϕ are equal to 42.77% and 46.48%, respectively for the models $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) and $\mathcal{M}_{\mathrm{I}}$. The trace plots for the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$, ϕ and $\mathbf{D}$ from this practical example are available in Appendix C.3.

For models $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) and $\mathcal{M}_{\mathrm{I}}$, Figure 4.3 displays the posterior distribution of ϕ and Figure 4.4 shows the posterior distributions of the parameters $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2, 3\})$, while the posterior distributions of $\beta_{0,1,t}$, $\beta_{0,2,t}$ and $\beta_{0,3,t}$ $(t = 0, 1, \ldots, 730)$ are shown respectively in Figures 4.5, 4.6 and 4.7. It can be noted that the posterior means of ϕ vary significantly from one another, as well as that model $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) provided wider credibility intervals than model $\mathcal{M}_{\mathrm{I}}$ for parameters $\beta_{0,1,t}$, $\beta_{0,2,t}$ and $\beta_{0,3,t}$. Among the covariances, only the 95% credibility interval of $V \cdot \Sigma_{1,2}$ does not contain the value zero. This negative association between Responses 1 and 2 is in accordance with the literature[2]. Specifically for the gauged site $n = 2$, the scatter plot presented in Figure 4.8 shows that nitrogen dioxide decreases over time, while ozone oscillates over time.

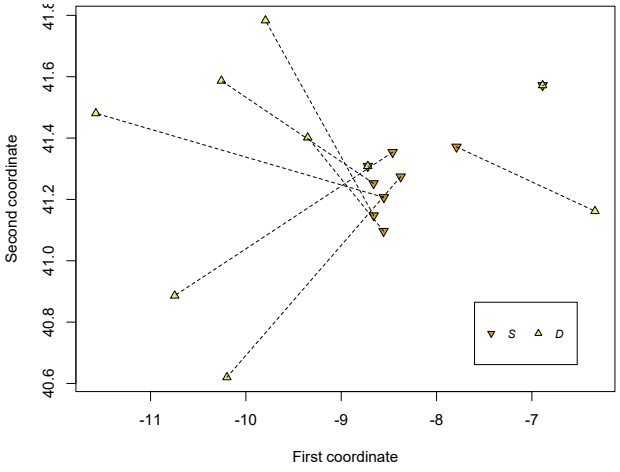
To facilitate visual comparison of missing data imputation, forecast and interpolation by model, we divided the time series into five intervals: 1 to 146, 147 to 292, 293 to 438, 439 to 584, and 585 to 730 days. Figures 4.9-4.13 show that the interpolation performance of model $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) is superior to the one of model $\mathcal{M}_{\mathrm{I}}$ at the ungauged site $\underset{\sim}{\mathbf{s}}_{12}$ for the second response variable, noting that their credibility intervals are narrower, which is in agreement with the results of the ECP and IS statistics.

[2]See, for example, Li et al. (2021).

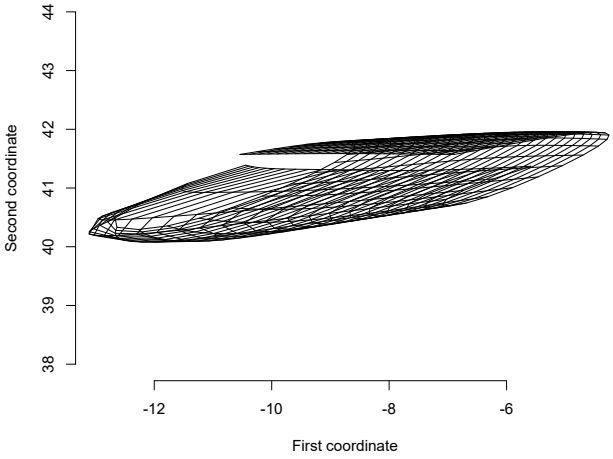
The results were similar for the two models, with a slight advantage for model $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) in relation to slightly narrower intervals in imputation. Thus, through this practical example we concluded that, to analyze real data with some anisotropy, model $\mathcal{M}_{\mathrm{A}}$ (with spatial deformation) given in Scenario 1 makes more assertive interpolation than model $\mathcal{M}_{\mathrm{I}}$ (without spatial deformation), with both having similar results for forecasting. The data analysis presented here has the limitation of exceeding safe percentages of missing values that guarantee good recovery of the parameters, as indicated in Section 2.5.1.



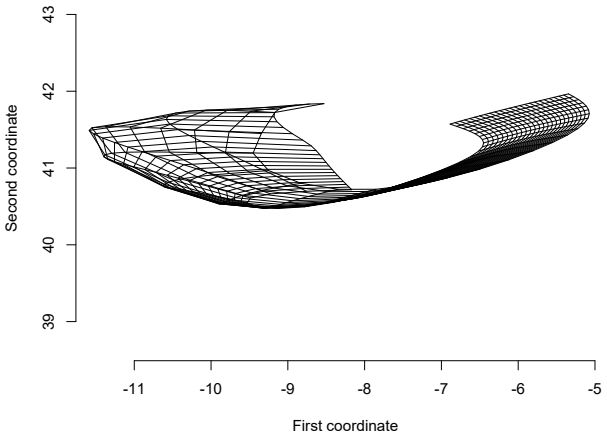
(a) Points of the geographic region and their posterior means of the corresponding deformation in Scenario 1.



(b) Points of the geographic region and their posterior means of the corresponding deformation in Scenario 2.



(c) Estimated deformation in Scenario 1.

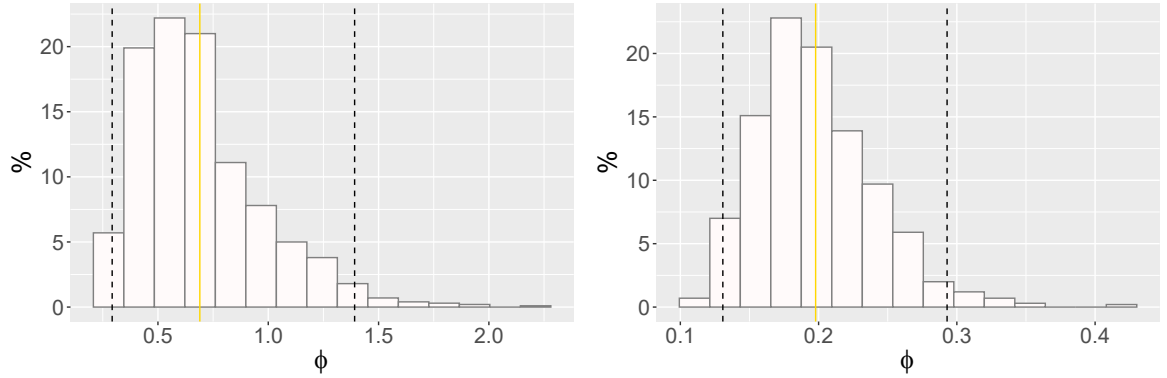


(d) Estimated deformation in Scenario 2.

Figure 4.2: Gauged sites and their estimated deformations by scenario, resulting from the practical example in Chapter 4.

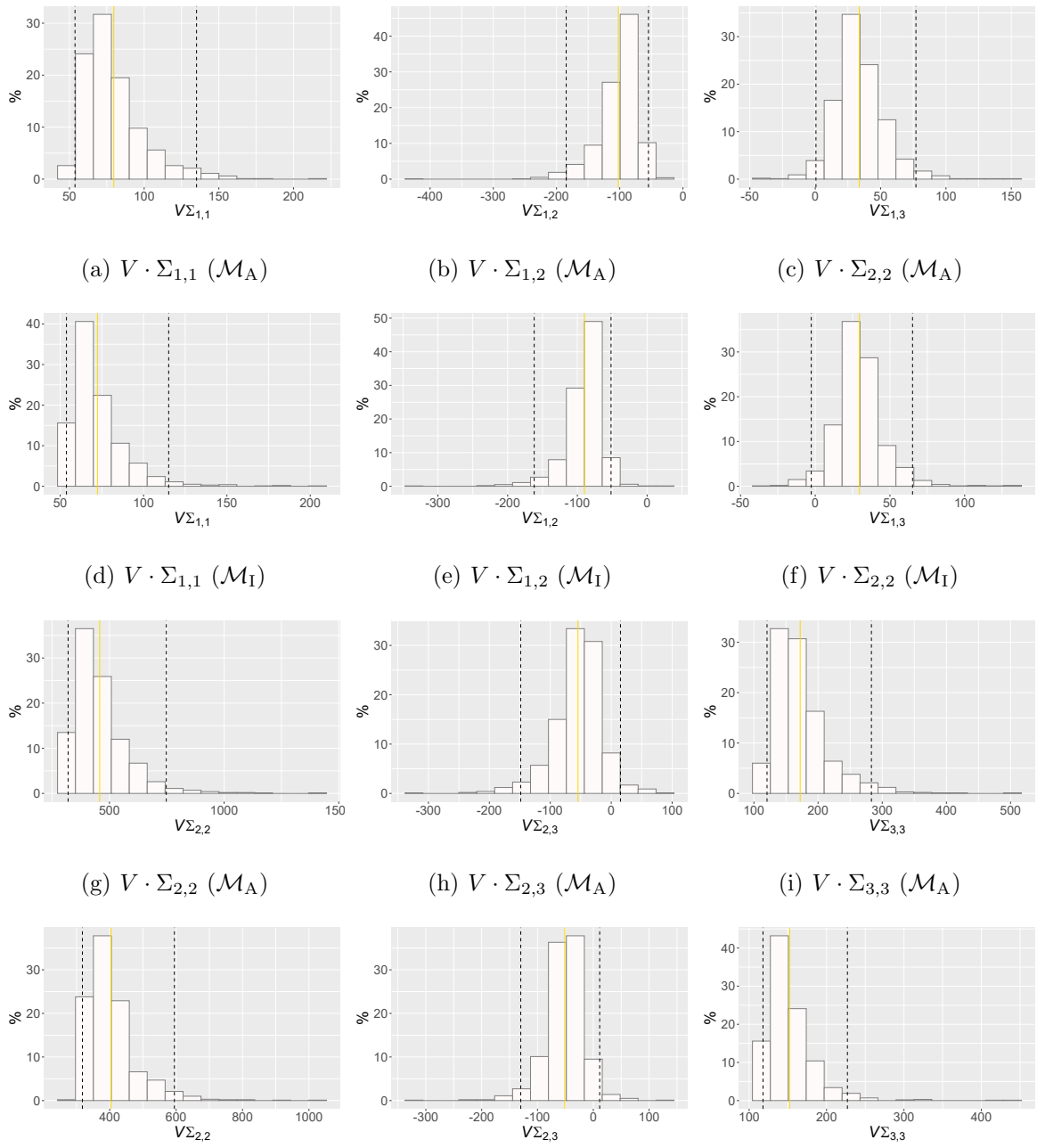
Table 4.1: Metrics for model comparison (DIC, PMSE, ECP and IS) by response variable, from the practical example in Chapter 4.

Metric	$\mathcal{M}_{\text{A}}$ in Scenario 1			$\mathcal{M}_{\text{A}}$ in Scenario 2			$\mathcal{M}_{\text{I}}$		
DIC	73111.4			72990.5			73781.9		
PMSE	179.2			425.3			336.4		
ECP (%)	**Resp. 1**	**Resp. 2**	**Resp. 3**	**Resp. 1**	**Resp. 2**	**Resp. 3**	**Resp. 1**	**Resp. 2**	**Resp. 3**
$\underset{\sim}{\mathbf{s}}_{10}$	97.3	97.7	100.0	93.2	86.9	94.6	100.0	93.4	97.3
IS	**Resp. 1**	**Resp. 2**	**Resp. 3**	**Resp. 1**	**Resp. 2**	**Resp. 3**	**Resp. 1**	**Resp. 2**	**Resp. 3**
$\underset{\sim}{\mathbf{s}}_{10}$	0.6209	1.5471	0.8756	0.8930	2.2165	2.9653	0.7045	2.0457	2.0260



(a) ϕ ($\mathcal{M}_{\text{A}}$) (b) ϕ ($\mathcal{M}_{\text{I}}$)

Figure 4.3: Histograms of the posterior distribution of ϕ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.



(a) $V \cdot \Sigma_{1,1}$ $(\mathcal{M}_{\mathrm{A}})$ (b) $V \cdot \Sigma_{1,2}$ $(\mathcal{M}_{\mathrm{A}})$ (c) $V \cdot \Sigma_{2,2}$ $(\mathcal{M}_{\mathrm{A}})$

(d) $V \cdot \Sigma_{1,1}$ $(\mathcal{M}_{\mathrm{I}})$ (e) $V \cdot \Sigma_{1,2}$ $(\mathcal{M}_{\mathrm{I}})$ (f) $V \cdot \Sigma_{2,2}$ $(\mathcal{M}_{\mathrm{I}})$

(g) $V \cdot \Sigma_{2,2}$ $(\mathcal{M}_{\mathrm{A}})$ (h) $V \cdot \Sigma_{2,3}$ $(\mathcal{M}_{\mathrm{A}})$ (i) $V \cdot \Sigma_{3,3}$ $(\mathcal{M}_{\mathrm{A}})$

(j) $V \cdot \Sigma_{2,2}$ $(\mathcal{M}_{\mathrm{I}})$ (k) $V \cdot \Sigma_{2,3}$ $(\mathcal{M}_{\mathrm{I}})$ (l) $V \cdot \Sigma_{3,3}$ $(\mathcal{M}_{\mathrm{I}})$

Figure 4.4: Histograms of the posterior distribution of $V \cdot \Sigma_{i,i'}$ $(i, i' \in \{1, 2, 3\})$ for models $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) and $\mathcal{M}_{\mathrm{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

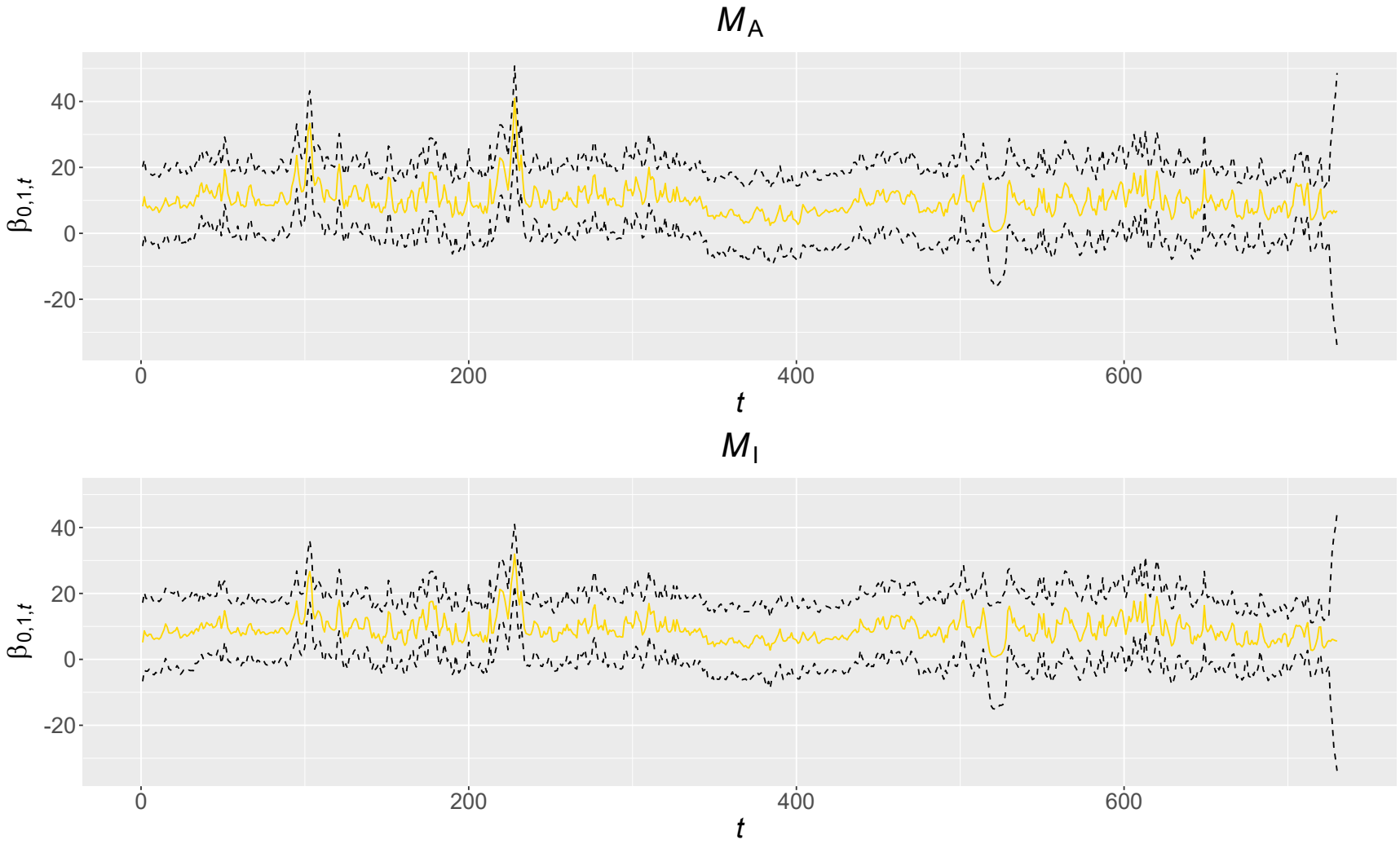


Figure 4.5: Line chart of the posterior distribution of $\beta_{0,1,t}$ for $t \in \{0, 1, \ldots, 730\}$ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

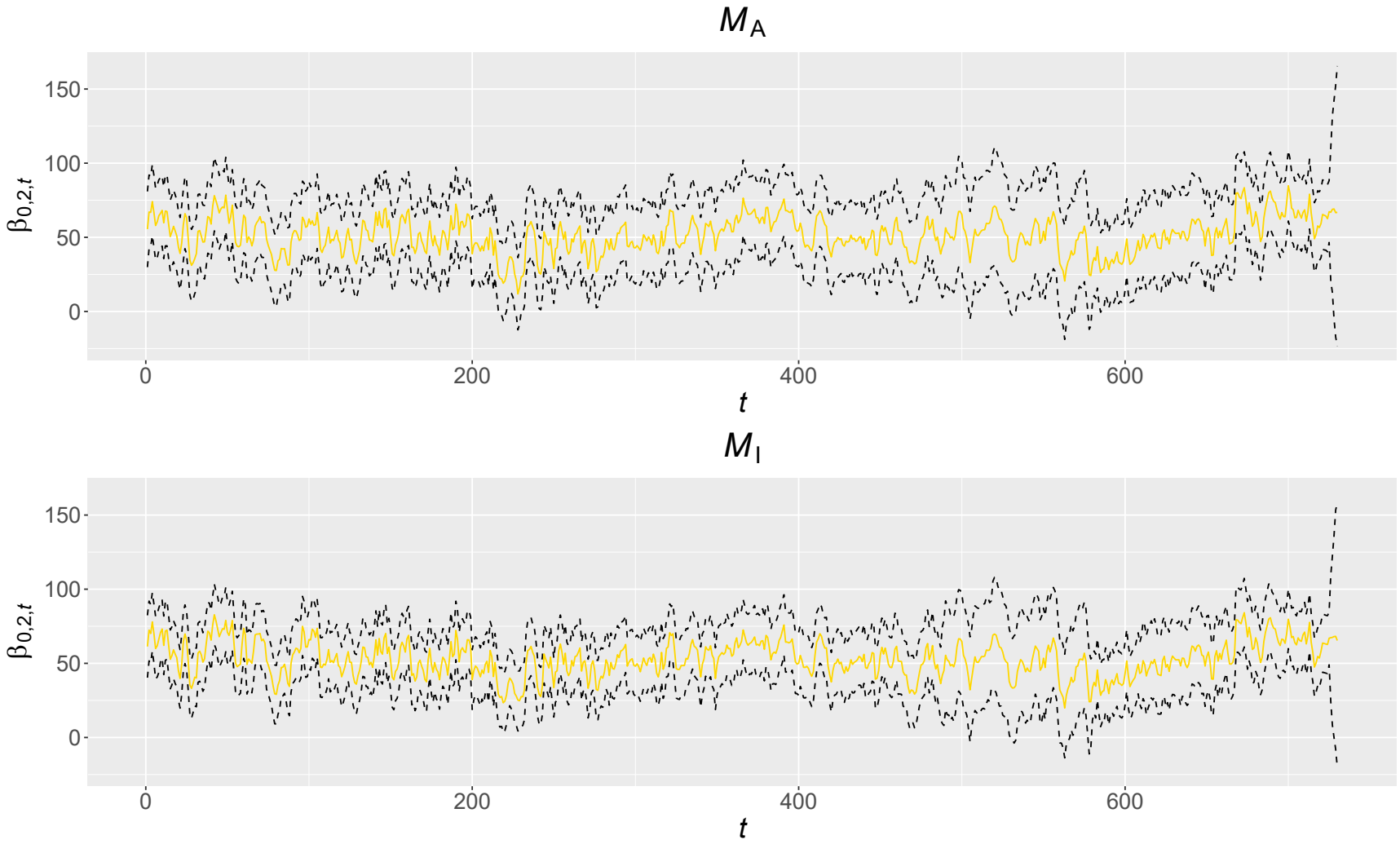


Figure 4.6: Line chart of the posterior distribution of $\beta_{0,2,t}$ for $t \in \{0, 1, \ldots, 730\}$ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

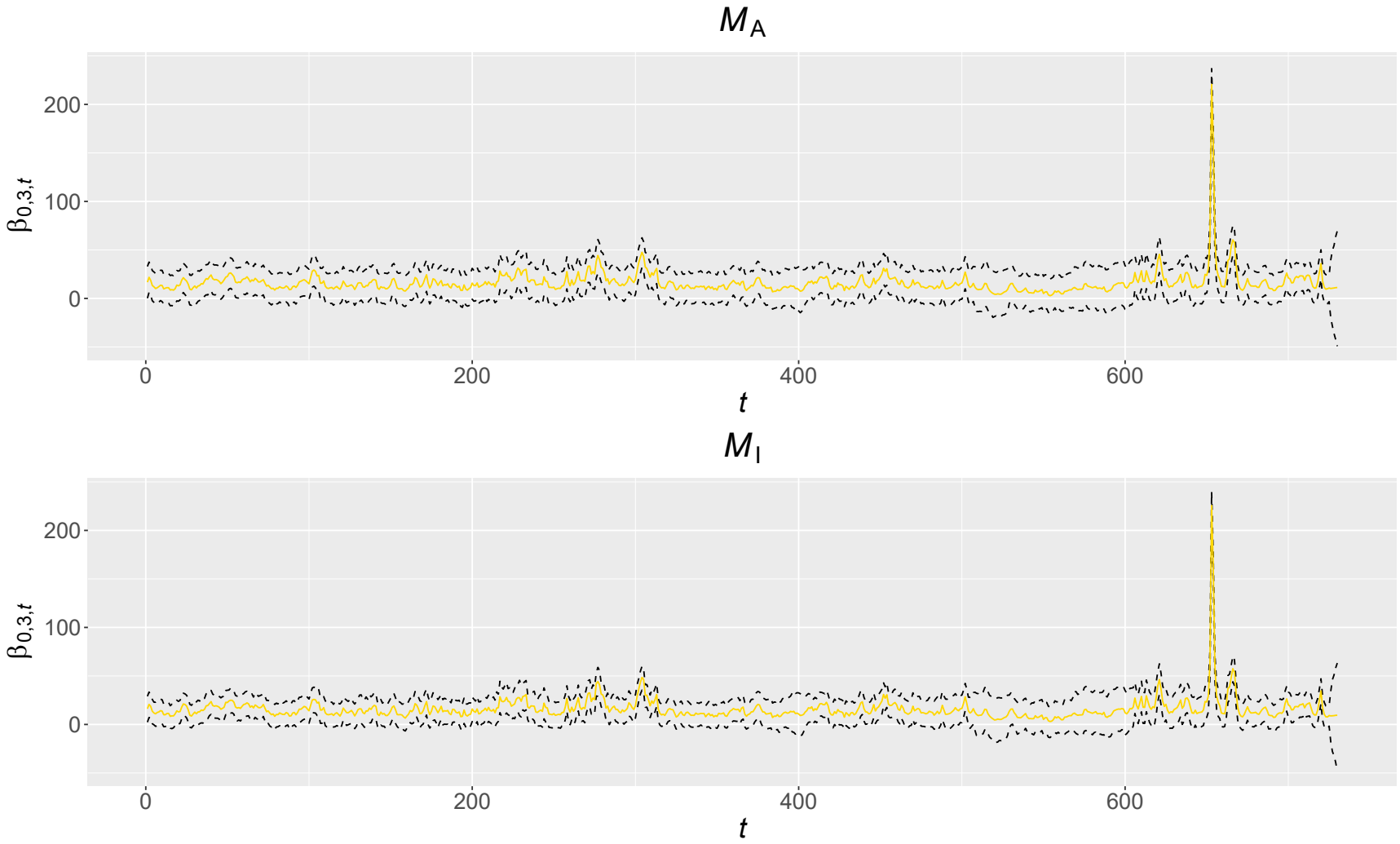


Figure 4.7: Line chart of the posterior distribution of $\beta_{0,3,t}$ for $t \in \{0, 1, \ldots, 730\}$ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines and the posterior mean is represented by the solid golden line.

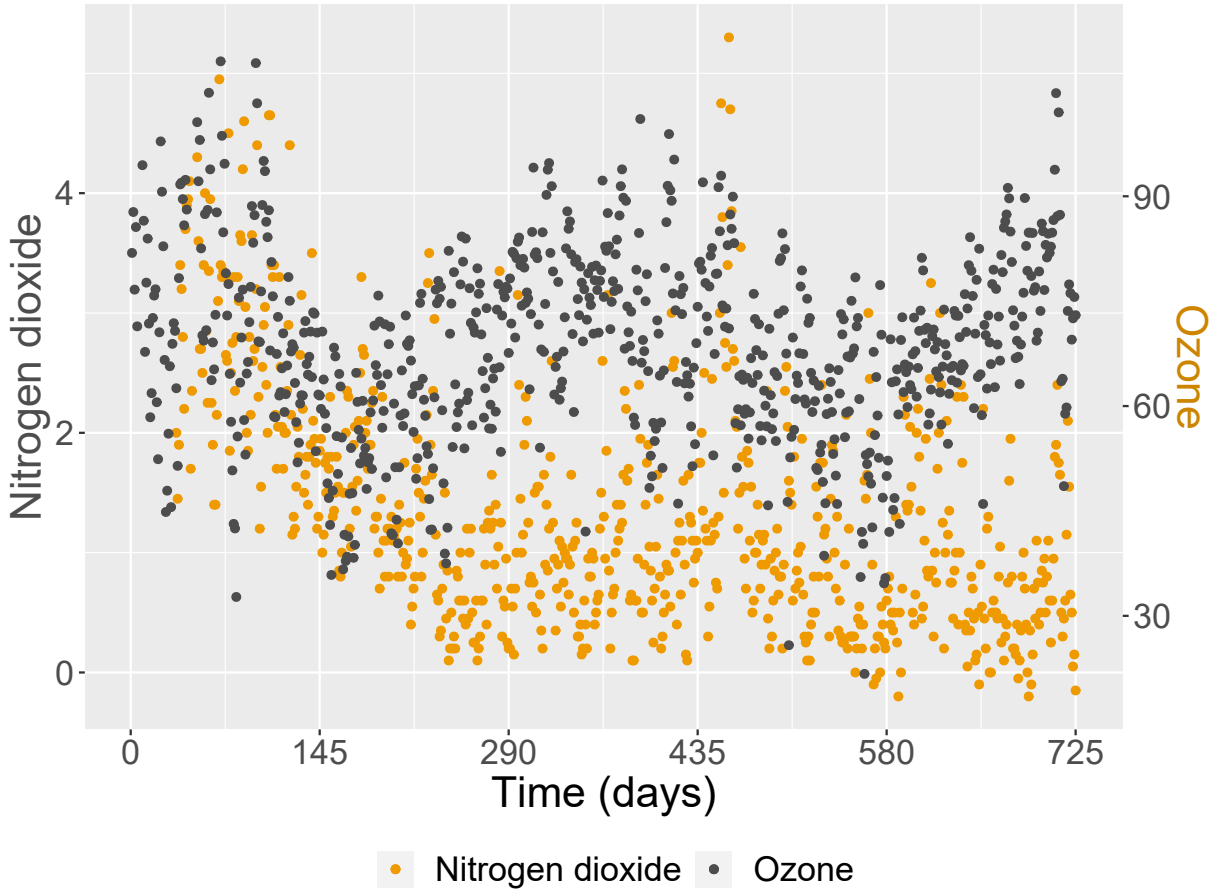


Figure 4.8: Scatter plot of time (in days) versus Responses 1 (nitrogen dioxide) and 2 (ozone) for the gauged site $n = 2$.

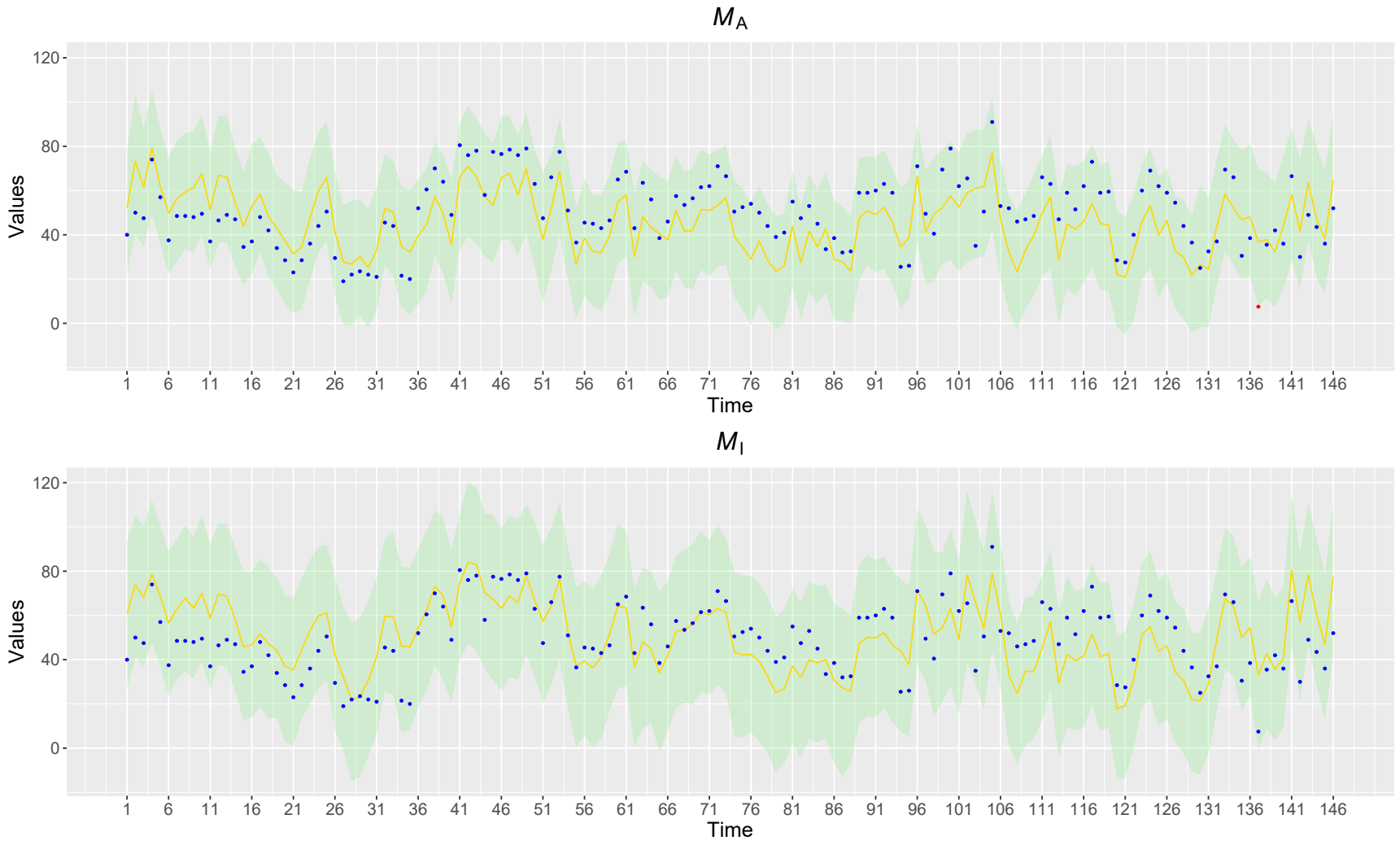


Figure 4.9: Line chart of the interpolated values of $Y_{10,i,t}$ for $i = 2$ and $t \in \{1, \ldots, 146\}$ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

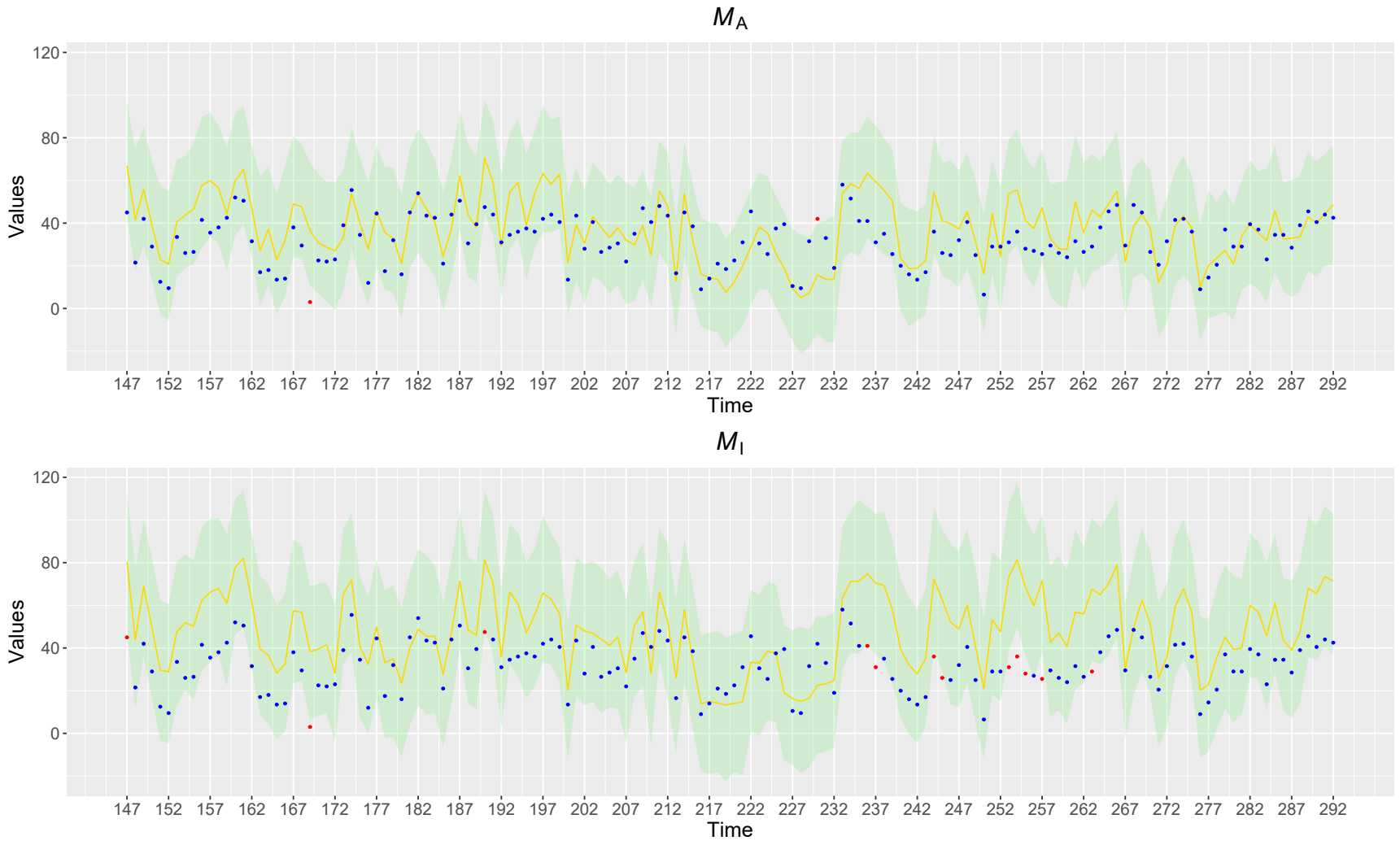


Figure 4.10: Line chart of the interpolated values of $Y_{10,i,t}$ for $i = 2$ and $t \in \{147, \ldots, 292\}$ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

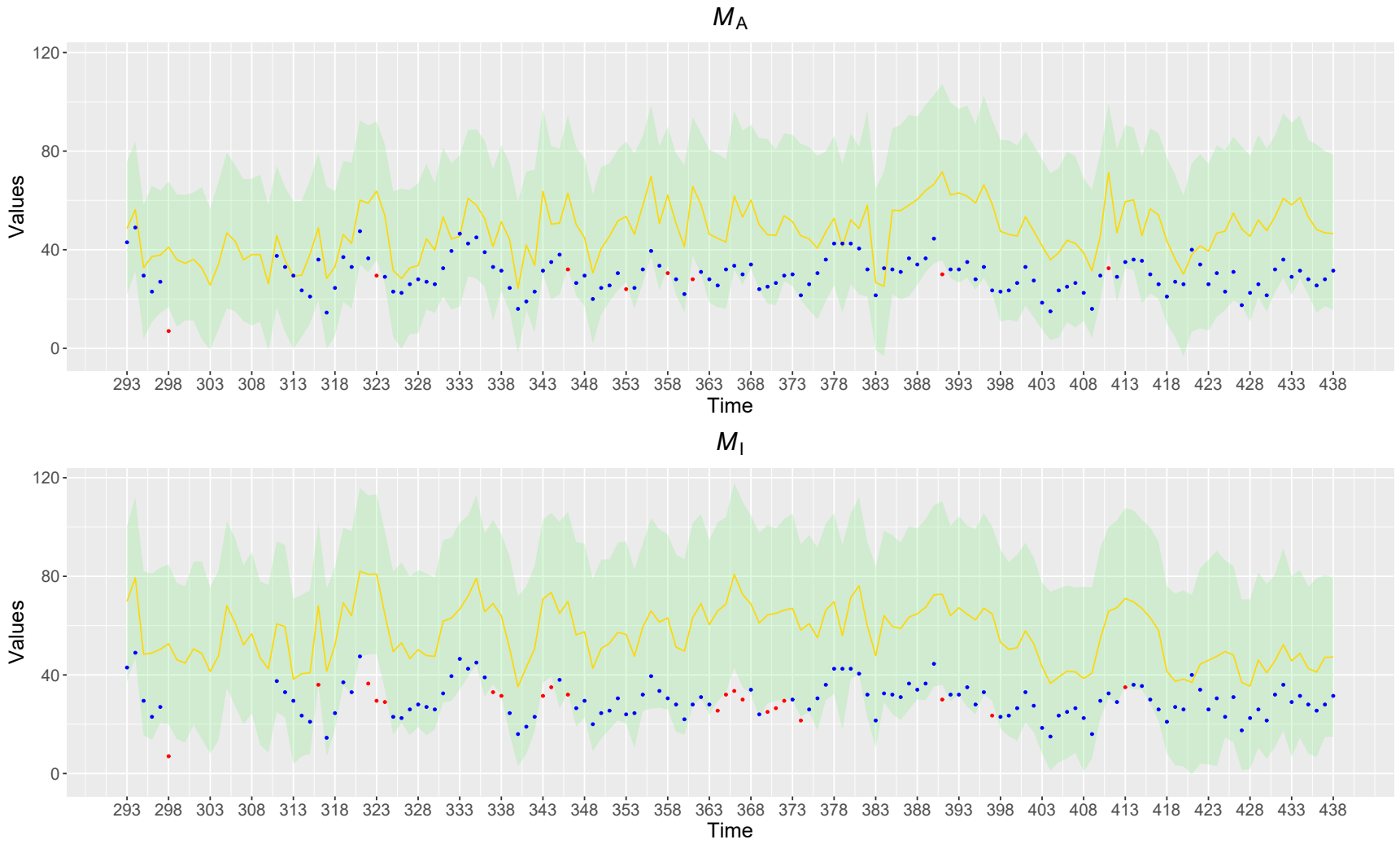


Figure 4.11: Line chart of the interpolated values of $Y_{10,i,t}$ for $i = 2$ and $t \in \{293, \ldots, 438\}$ for models $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) and $\mathcal{M}_{\mathrm{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

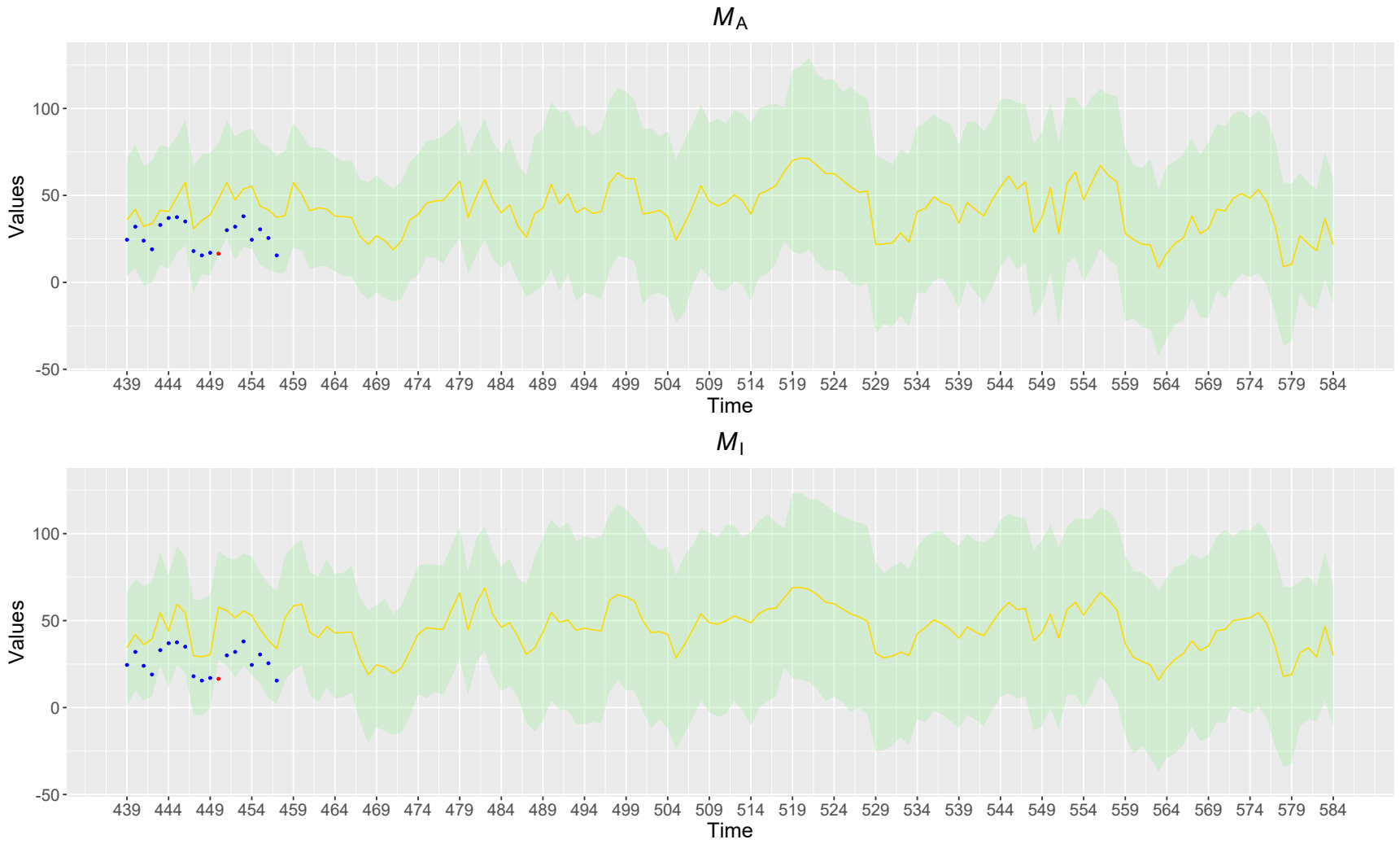


Figure 4.12: Line chart of the interpolated values of $Y_{10,i,t}$ for $i = 2$ and $t \in \{439, \ldots, 584\}$ for models $\mathcal{M}_{\text{A}}$ (Scenario 1) and $\mathcal{M}_{\text{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

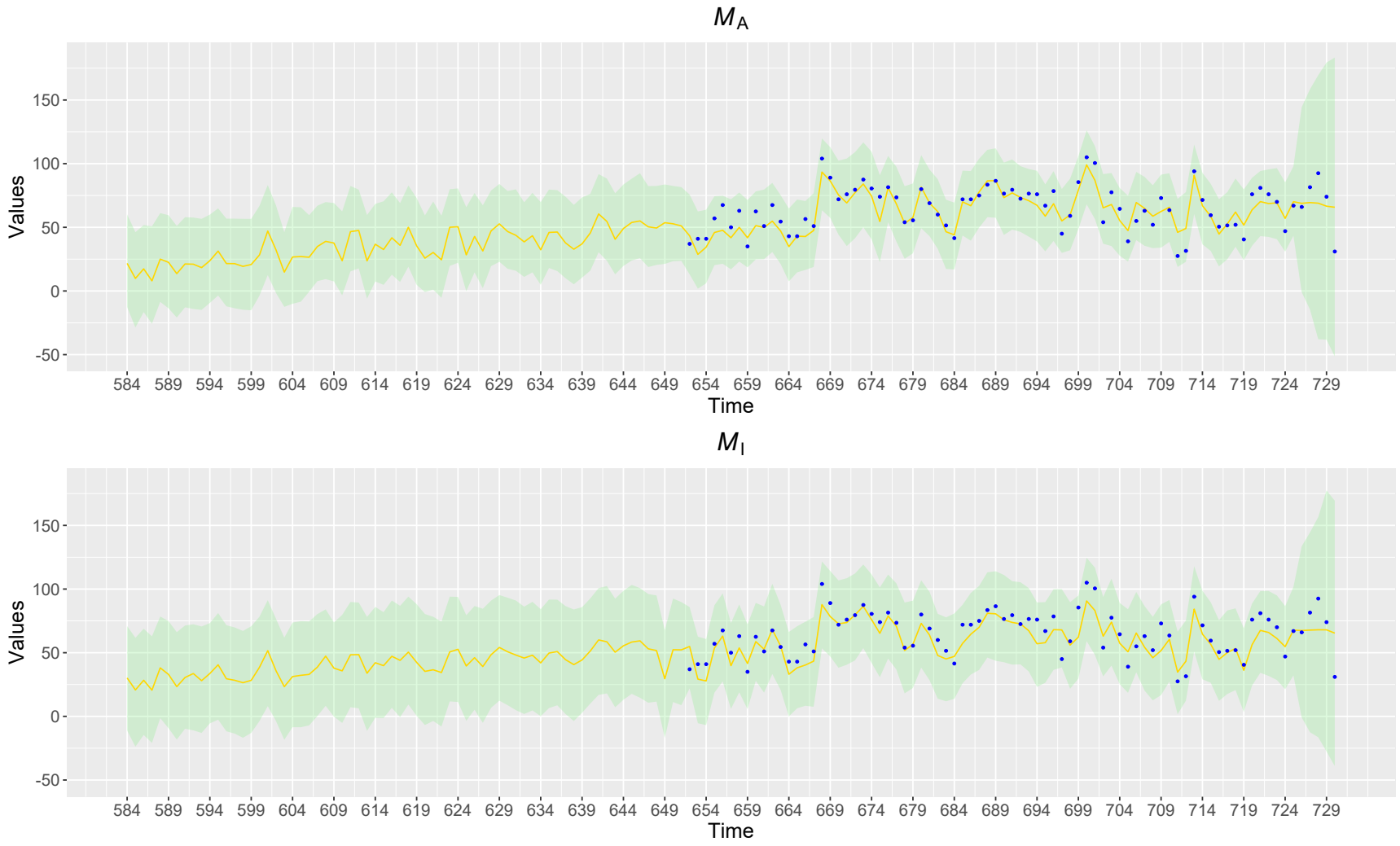


Figure 4.13: Line chart of the interpolated values of $Y_{10,i,t}$ for $i = 2$ and $t \in \{585, \ldots, 730\}$ for models $\mathcal{M}_{\mathrm{A}}$ (Scenario 1) and $\mathcal{M}_{\mathrm{I}}$, resulting from the practical example in Chapter 4. The 2.5^{th} and 97.5^{th} posterior quantiles are represented by the green shaded area and the posterior mean is represented by the solid golden line. Points represent the true values (in blue, if within range; in red, otherwise).

Chapter 5

Final remarks and future work

This doctoral thesis aimed to develop a Bayesian spatiotemporal model to adjust two or more response variables in the context in which they are measured at fixed locations in a continuous space (geostatistics) and at different discrete and equispaced points in time, having the relaxation of the hypothesis of spatial isotropy as its main highlight. To this end, we use the concept of spatial deformation (Sampson and Guttorp, 1992) to treat anisotropy in a matrix-variate spatiotemporal model inspired by the statistical models of Paez et al. (2008) and Morales et al. (2013).

In Chapter 2, we initially review the spatial deformation model of Schmidt and O'Hagan (2003) and highlight the impact of the choices of hyperparameters $\boldsymbol{\sigma}_d^2$ and ψ. We then present the statistical model proposed in this thesis in two steps. First, we modeled each multivariate response using an adapted version of the model presented in Morales et al. (2013). After that, we assume that all response variables share the same spatial dependence structure, as done in Paez et al. (2008), to then write a model for matrix-variate responses.

Since the posterior distribution of the parameters of the model proposed in Chapter 2 is not analytically tractable, we present a hybrid algorithm that allows sampling from this distribution and its implementation in the Python language. This algorithm is distinguished from other works by using the slice sampler to sample from the full conditional distribution of the spatial deformation parameter. Predictive distributions for forecasting and interpolation purposes were presented, accompanied by explanations on how to approximate them using Monte Carlo integration.

As we wish to evaluate whether the use of spatial deformation brings an advantage to the matrix-variate spatiotemporal model developed in Chapter 2, we review some metrics for model comparison purposes and present mathematical expressions carefully adapted to the notation used in the text. This comparison was made through a simulation study, in which we described in

detail how we generate deterministic deformations, as well as from the analysis of real data. We verified that there are gains in the adjustment and quality of interpolation in both cases considered, which shows that the use of spatial deformation can bring advantages to the modelling of other matrix-variate spatiotemporal datasets.

In the illustrative example discussed in Chapter 2, missing values from the response matrices were imputed before applying the model. Because of this, Chapter 3 discussed using the augmented data algorithm to simultaneously estimate model parameters and missing values. Although it seems like an immediate extension of the model discussed in Chapter 2, the matrix treatment is vectorized due to the fact that it is not possible to split response matrices into blocks in many real-world situations. Therefore, the inference procedure, predictive distributions for forecasting and interpolation, and metrics for model comparison are meticulously written for the context of incomplete data.

A first simulation study showed the impact of increasing the fraction of values on parameter recovery. The second simulation study aimed to compare models with and without spatial deformation in terms of adjustment, recovery of missing values and interpolation of anisotropic data generated via deterministic deformation, having verified that these three aspects benefit from the incorporation of spatial deformation into the suggested model. The same illustrative example considered in Chapter 2 was analyzed again in Chapter 3, with the distinction that missing values were estimated rather than previously imputed. Although the results were very similar, our approach allows us to study the uncertainty of estimates for missing values instead of having only point estimates for them.

Finally, Chapter 4 presented an analysis of real data from the Northern region of Portugal in a case in which there is interest in simultaneously modeling three measurements of pollutants that vary in time and space. The three response variables have a high fraction of missing values. Once again, it was possible to show that the use of spatial deformation improves the quality of the adjustment and interpolation.

A disadvantage of the model proposed in this thesis is due to the difficulty in specifying the values of the hyperparameters $\boldsymbol{\sigma}_d^2$ and ψ, making it difficult to guarantee good results for all four metrics considered here at the same time. Our suggestion of using $\boldsymbol{\sigma}_d^2 = \widehat{\text{Cov}}(\mathbf{S})$ turned out to be useful, but choosing ψ based on trials tends to be an exhausting work. A limitation to be noted is that we adopted the assumption that missing values occur in a random pattern, which does not accommodate the predominantly sequential pattern of appearance of missing values in the actual

analyzed data.

For future work, we understand that faster algorithms need to be implemented to facilitate the exhaustive work of specifying hyperparameters of the prior distribution of $\mathbf{D}$. In the context of the use of spatial deformation in spatial point processes, Quintana (2022) implemented the Hamiltonian Monte Carlo algorithm to obtain good approximations of the posterior distribution of the parameters with a significantly reduced number of iterations. A similar approach can be carried out with the model proposed in this thesis, since matrix differentiation results allow us to obtain analytical expressions for the derivative of the full conditional distribution of $\mathbf{D}$. More approximate Bayesian calculation methods can also be considered for implementation.

Other future research may involve new sensitivity studies of the hyperparameters of the prior distribution of $\mathbf{D}$, variation over time of certain quantities considered fixed[1], estimation of missing values that do not assume randomness and relaxation of other hypotheses, such as normality in distribution and the assumption that all response variables have the same spatial dependence structure.

[1] Morales (2010, Chapter 3) proposed time variations for spatial deformations.

Appendix A

Literature review and additional topics

A.1 Probability distributions

Definition A.1. Let a and b be real numbers such that $a < b$. The random variable X is said to have a *continuous uniform distribution* on the interval (a, b), which is denoted by $X \sim \mathsf{U}(a, b)$, if its density is, for all $a < x < b$, given by:

$$f(x) = \frac{1}{b-a}. \tag{A.1}$$

Definition A.2. Let a and b be positive numbers. The random variable X is said to have a *gamma distribution* with parameters a and b, which is denoted by $X \sim \mathsf{G}(a, b)$, if its density is, for all $x > 0$, given by:

$$f(x) = \frac{b^a}{\Gamma(a)} x^{a-1} \exp\{-bx\}. \tag{A.2}$$

Definition A.3. If $Y \sim \mathsf{G}(a, b)$, its transformation $X = 1/Y$ is said to have an *inverse gamma distribution* with parameters a and b, which is denoted by $X \sim \mathsf{IG}(a, b)$, whose density is, for all $x > 0$, given by:

$$f(x) = \frac{b^a}{\Gamma(a)} x^{-a-1} \exp\left\{-\frac{b}{x}\right\}. \tag{A.3}$$

We write $\mathbf{A} \in \mathrm{Sym}^+(p)$ to denote that a $p \times p$ matrix $\mathbf{A}$ is symmetric positive definite. This notation is also used by Jung et al. (2015).

Definition A.4. Let $\mathbf{X} = (X_1, \ldots, X_n)$ be a random vector with mean vector $\boldsymbol{\mu} = \mathrm{E}[\mathbf{X}]$ and covariance matrix $\boldsymbol{\Sigma} = \mathrm{E}[(\mathbf{X} - \boldsymbol{\mu})(\mathbf{X} - \boldsymbol{\mu})^\intercal]$, where $n \in \mathbb{N}$. The random vector $\mathbf{X}$ is said to have a

multivariate normal distribution with mean vector $\boldsymbol{\mu} \in \mathbb{R}^n$ and covariance matrix $\boldsymbol{\Sigma} \in \mathrm{Sym}^+(n)$, which is denoted by $\mathbf{X} \sim \mathsf{N}_n(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, if its density is, for all $\mathbf{x} \in \mathbb{R}^n$, given by:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{n}{2}} (\det \boldsymbol{\Sigma})^{\frac{1}{2}}} \exp\left\{ -\frac{1}{2} (\mathbf{x} - \boldsymbol{\mu})^\intercal \boldsymbol{\Sigma}^{-1} (\mathbf{x} - \boldsymbol{\mu}) \right\}. \tag{A.4}$$

When $n = 1$, we simply write $X \sim \mathsf{N}(\mu, \sigma^2)$. In this case, the probability density is, for all $x \in \mathbb{R}$, given by:

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left\{ -\frac{(x-\mu)^2}{2\sigma^2} \right\}.$$

Definition A.5. The random variable X is said to have a *inverse normal distribution* with mean $\mu > 0$ and shape parameter $\lambda > 0$, which is denoted by $X \sim \mathsf{IN}(\mu, \lambda)$, if its density is, for all $x > 0$, given by:

$$f(x) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left\{ -\frac{\lambda(x-\mu)^2}{2\mu^2 x} \right\}. \tag{A.5}$$

Definition A.6. Let $\mathbf{M}$, $\mathbf{C}$ and $\boldsymbol{\Sigma}$ be matrices with respective dimensions given by $r \times q$, $r \times r$ and $q \times q$, where $r, q \in \mathbb{N}$. Suppose further that $\mathbf{C} \in \mathrm{Sym}^+(r)$ and $\boldsymbol{\Sigma} \in \mathrm{Sym}^+(q)$. The $r \times q$ random matrix $\mathbf{X}$ is said to have a *matrix-variate normal distribution* with mean matrix $\mathbf{M}$, left covariance matrix $\mathbf{C}$ (among-row) and right covariance matrix $\boldsymbol{\Sigma}$ (among-column), which is denoted by $\mathbf{X} \sim \mathsf{N}_{r\times q}(\mathbf{M}, \mathbf{C}, \boldsymbol{\Sigma})$, if $\mathrm{vec}(\mathbf{X}) \sim \mathsf{N}_{rq}(\mathrm{vec}(\mathbf{M}), \boldsymbol{\Sigma} \otimes \mathbf{C})$.

It can be shown (see Gupta and Nagar, 2000, Theorem 2.2.1) that the density of the random matrix $\mathbf{X} \sim \mathsf{N}_{r\times q}(\mathbf{M}, \mathbf{C}, \boldsymbol{\Sigma})$ is, for all $\mathbf{x} \in \mathbb{R}^{r\times q}$, given by:

$$f(\mathbf{x}) = \frac{1}{(2\pi)^{\frac{rq}{2}} [\det \mathbf{C}]^{\frac{q}{2}} [\det \boldsymbol{\Sigma}]^{\frac{r}{2}}} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[(\mathbf{x} - \mathbf{M})^\intercal \mathbf{C}^{-1} (\mathbf{x} - \mathbf{M}) \boldsymbol{\Sigma}^{-1} \right] \right\}. \tag{A.6}$$

Definition A.7. Let $\mathbf{S}$ be a $p \times p$ nonsingular matrix. The $p \times p$ random matrix $\mathbf{X}$ is said to have an *inverse Wishart distribution* with r degrees of freedom and $p \times p$ scale matrix $\mathbf{S}$, which is denoted by $\mathbf{X} \sim \mathsf{IW}_p(r, \mathbf{S})$, if its density is, for all $\mathbf{x} \in \mathrm{Sym}^+(p)$, proportional to:

$$f(\mathbf{x}) \propto (\det \mathbf{x})^{-\left(p + \frac{r}{2}\right)} \exp\left\{ -\frac{1}{2} \operatorname{tr}\left(\mathbf{S}\mathbf{x}^{-1} \right) \right\}. \tag{A.7}$$

The inverse Wishart distribution has mean $\mathrm{E}[\mathbf{X}] = \mathbf{S}/(r-2)$. If the matrices $\mathbf{X}$ and $\mathbf{S}$ are partitioned into four blocks, say $\mathbf{X}_{i,j}$ and $\mathbf{S}_{i,j}$ for $i, j \in \{1, 2\}$, we have $\mathbf{X}_{i,i} \sim \mathsf{IW}_{p_i}(\mathbf{S}_{i,i}, r)$ for $i \in \{1, 2\}$, where $p_{1,1}$ and $p_{2,2}$ are integers such that $p_{1,1} + p_{2,2} = p$. When $p = 1$, one can write $f(x) \propto x^{-\frac{r}{2}-1} \exp\left\{ -\frac{S/2}{x} \right\}$ for all $x > 0$, implying that $X \sim \mathsf{IG}(r/2, S/2)$. See Quintana (1987) and Dawid (1981) for further details.

A.2 Geostatistics

DEFINITION A.8. A family of random variables $Y(\underset{\sim}{\mathbf{s}})$ that are defined on the same probability space and indexed by $\underset{\sim}{\mathbf{s}}$ in a subset $\mathcal{S}$ of $\mathbb{R}^r$, where $r \in \mathbb{N}$, is called a *random process*, which is denoted by $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$. The random processes take values in a subset $\mathcal{Y}$ of $\mathbb{R}^q$ known as the *state-space*, where $q \in \mathbb{N}$. If $q = 1$, $Y(\cdot)$ is said to be a *one-dimensional process*. If $q > 1$, $Y(\cdot)$ is said to be a *q-dimensional process*.

The normal distribution is the most used probabilistic model in applications. Next, we present the following analogues for random processes.

DEFINITION A.9. Let $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$ be a random process such that $\mathcal{Y} \subset \mathbb{R}$. If the N-dimensional random vector $\mathbf{Y} = (Y(\underset{\sim}{\mathbf{s}}_{n_1}), \ldots, Y(\underset{\sim}{\mathbf{s}}_{n_N}))$ follows a multivariate normal distribution for all finite subset $\{\underset{\sim}{\mathbf{s}}_{n_1}, \ldots, \underset{\sim}{\mathbf{s}}_{n_N}\} \subset \mathcal{S}$ and all $N \in \mathbb{N}$, then $Y(\cdot)$ is said to be a (unidimensional) *Gaussian process*.

DEFINITION A.10. Let $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$ be a random process such that $\mathcal{Y} \subset \mathbb{R}^q$, where $q \in \mathbb{N}$. If the $q \times N$ random matrix $\mathbf{Y} = \begin{bmatrix} Y(\underset{\sim}{\mathbf{s}}_{n_1}) & \cdots & Y(\underset{\sim}{\mathbf{s}}_{n_N}) \end{bmatrix}$ follows a matrix-variate normal distribution[1] for all finite subset $\{\underset{\sim}{\mathbf{s}}_{n_1}, \ldots, \underset{\sim}{\mathbf{s}}_{n_N}\} \subset \mathcal{S}$ and all $N \in \mathbb{N}$, then $Y(\cdot)$ is said to be a *q-dimensional Gaussian process*.

In geostatistics is usual to specify a random process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$ called *spatial process*, where $\underset{\sim}{\mathbf{s}}$ is a *site* located in the *region of interest* $\mathcal{S} \subset \mathbb{R}^r$. In most applications, spatial processes are one-dimensional (i.e. $\mathcal{Y} \subset \mathbb{R}$). When $\mathcal{S} \subset \mathbb{R}^2$ (most common case), $\underset{\sim}{\mathbf{s}}$ is commonly a site that contains the longitude and the latitude coordinates. If $\mathcal{S} \subset \mathbb{R}^3$, $\underset{\sim}{\mathbf{s}}$ is usually a site characterized by its longitude, latitude and elevation above sea level (or sea depth). The Universal Transverse Mercator (UTM) coordinate system is also widely used. It is assumed that the function $\mu : \mathbb{R}^r \to \mathbb{R}^q$ such that $\mu(\underset{\sim}{\mathbf{s}}) = \mathrm{E}[Y(\underset{\sim}{\mathbf{s}})]$ exists for all $\underset{\sim}{\mathbf{s}} \in \mathcal{S}$, where $\mu(\cdot)$ is known as the *trend* (or *drift*).

Stationarity means that some characteristics of a spatial process stay the same when shifting a given set of points from one part of the domain/region of interest to another (Wackernagel, 2003, Chapter 5). Next, we present the concepts of strict, second-order and intrinsic stationarity.

DEFINITION A.11. A one-dimensional spatial process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$ is said to be *strictly stationary* (or *strongly starionary*) if the N-dimensional random vectors $(Y(\underset{\sim}{\mathbf{s}}_{n_1}), \ldots, Y(\underset{\sim}{\mathbf{s}}_{n_N}))$

[1] Please see Definition A.6 for a review.

and $(Y(\underset{\sim}{\mathbf{s}}_{n_1} + \underset{\sim}{\mathbf{h}}), \ldots, Y(\underset{\sim}{\mathbf{s}}_{n_N} + \underset{\sim}{\mathbf{h}}))$ have the same probability distribution, for all finite subset $\{\underset{\sim}{\mathbf{s}}_{n_1}, \ldots, \underset{\sim}{\mathbf{s}}_{n_N}\} \subset \mathcal{S}$, all vector $\underset{\sim}{\mathbf{h}} \in \mathbb{R}^r$ such that $\{\underset{\sim}{\mathbf{s}}_{n_1} + \underset{\sim}{\mathbf{h}}, \ldots, \underset{\sim}{\mathbf{s}}_{n_N} + \underset{\sim}{\mathbf{h}}\} \subset \mathcal{S}$ and all $N \in \mathbb{N}$.

Strict stationarity is a strong assumption because the finite-dimensional distributions are rarely translation invariant. A mild form of stationarity assumes that the mean is constant for all sites and the covariance is a function of the *separation vector* (or *spatial lag*) $\underset{\sim}{\mathbf{h}}$.

Definition A.12. A one-dimensional spatial process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$ is said to be *second-order stationary* (or *weakly stationary*) if $\mathrm{E}[Y^2(\underset{\sim}{\mathbf{s}})] < \infty$, $\mu(\underset{\sim}{\mathbf{s}}) = \mu$ and $\mathrm{Cov}[Y(\underset{\sim}{\mathbf{s}}), Y(\underset{\sim}{\mathbf{s}}')] = C(\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}')$ for all $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$, where $C : \mathbb{R}^r \to \mathbb{R}$ is a function known as a *covariogram*.

A covariogram must be positive-definite (Cressie, 1993, Sec. 2.3.2). Moveover, it holds that $C(\underset{\sim}{\mathbf{0}}) \geq 0$, $C(\underset{\sim}{\mathbf{h}}) = C(-\underset{\sim}{\mathbf{h}})$ and $|C(\underset{\sim}{\mathbf{h}})| < C(\underset{\sim}{\mathbf{0}})$ for all $\underset{\sim}{\mathbf{h}} = \underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'$ and $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$ (van Lieshout, 2019, Prop. 2.2). Next, we present a concept that is directly related to the covariogram function.

Definition A.13. Let $C(\cdot)$ be a covariogram function of a one-dimensional second-order stationary spatial process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$, such that $C(\underset{\sim}{\mathbf{0}}) > 0$. For all $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$ and $\underset{\sim}{\mathbf{h}} = \underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'$, a *correlogram* is a function $\rho : \mathbb{R}^r \to \mathbb{R}$ given by $\rho(\underset{\sim}{\mathbf{h}}) = C(\underset{\sim}{\mathbf{h}})/C(\underset{\sim}{\mathbf{0}})$.

Just as a covariogram, a correlogram must be positive-definite in order to define a legitimate model (Diggle and Ribeiro-Jr., 2007, Sec. 2.2). In addition, $\rho(\cdot)$ verifies that $\rho(\underset{\sim}{\mathbf{0}}) = 1$, $\rho(\underset{\sim}{\mathbf{h}}) = \rho(-\underset{\sim}{\mathbf{h}})$ and $|\rho(\underset{\sim}{\mathbf{h}})| \leq 1$ for all $\underset{\sim}{\mathbf{h}} = \underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'$ and $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$ as consequences of the properties of $C(\cdot)$.

An even lighter version of stationarity assumes that the increments are zero-mean random variables with variance defined by a function that depends only on the separation vector.

Definition A.14. A one-dimensional spatial process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$ is said to be *intrinsically stationary* if $\mathrm{E}[Y^2(\underset{\sim}{\mathbf{s}})] < \infty$, $\mathrm{E}[Y(\underset{\sim}{\mathbf{s}}) - Y(\underset{\sim}{\mathbf{s}}')] = 0$ and $\mathrm{Var}[Y(\underset{\sim}{\mathbf{s}}) - Y(\underset{\sim}{\mathbf{s}}')] = 2\gamma(\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}')$ for all $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$, where $\gamma : \mathbb{R}^r \to \mathbb{R}$ is a function known as a *semivariogram*.

A semivariogram must be negative-definite (Cressie, 1993, Sec. 2.5.2). For all $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$ and $\underset{\sim}{\mathbf{h}} = \underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'$, it turns out that $\gamma(\underset{\sim}{\mathbf{0}}) = 0$, $\gamma(\underset{\sim}{\mathbf{h}}) = \gamma(-\underset{\sim}{\mathbf{h}})$ and $\gamma(\underset{\sim}{\mathbf{h}}) = C(\underset{\sim}{\mathbf{0}}) - C(\underset{\sim}{\mathbf{h}})$ (Carvalho and Natário, 2008, Secs. 2.1.2 and 2.1.3). This last result implies $C(\underset{\sim}{\mathbf{0}}) = \lim_{\|\underset{\sim}{\mathbf{h}}\| \to \infty} \gamma(\underset{\sim}{\mathbf{h}})$ if we assume that $\lim_{\|\underset{\sim}{\mathbf{h}}\| \to \infty} C(\underset{\sim}{\mathbf{h}}) = 0$ (Banerjee et al., 2014, Sec. 2.1.2), a condition that can be understood as a weak form of asymptotic independence, where $\|\cdot\|$ denotes the Euclidean distance. Next, we present some elements of semivariograms.

Definition A.15. Let $\gamma(\cdot)$ be a semivariogram function of a one-dimensional intrinsically stationary spatial process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$. If $\gamma(\underset{\sim}{\mathbf{0}}^+) \equiv \lim_{\|\underset{\sim}{\mathbf{h}}\| \to 0^+} \gamma(\underset{\sim}{\mathbf{h}}) = c_0^2 > 0$, c_0^2 is called the *nugget effect*. The asymptotic value $\lim_{\|\underset{\sim}{\mathbf{h}}\| \to \infty} \gamma(\underset{\sim}{\mathbf{h}}) = c_0^2 + c_1^2$ is named the *sill*, where $c_1^2 > 0$ is known as the *partial sill*. The value $\|\underset{\sim}{\mathbf{h}}\| = 1/\phi$ at which $\gamma(\underset{\sim}{\mathbf{h}})$ first reaches its ultimate level (the sill) is said to be the *range*, where $\phi > 0$ is known as the *decay*.

The concepts of stationarity presented are related. Strict stationarity implies weak stationarity and weak stationarity implies intrinsic stationarity (Cressie, 1993, Sec. 2.3). Since a Gaussian process is characterized by its mean and covariance functions, it holds that second-order stationarity implies strict stationarity in this case (van Lieshout, 2019, Sec. 2.3). Intrinsic stationarity implies second-order stationarity when the sill is finite (Carvalho and Natário, 2008, Sec. 2.1.2). See Murteira et al. (1993, Chapter 2), Gonçalves and Mendes-Lopes (2008, Chapter 1) and Morettin and Toloi (2018, Chapter 2) for a review about stationary stochastic processes.

Isotropy and anisotropy are crucial concepts for this work. Intuitively, the interest is to study whether or not the semivariogram has the same behaviour for all directions (Soares, 2006, Sec. 3.5.6).

Definition A.16. Let $\gamma(\cdot)$ be a semivariogram function of a one-dimensional intrinsically stationary spatial process $Y(\cdot) = \{Y(\underset{\sim}{\mathbf{s}}) : \underset{\sim}{\mathbf{s}} \in \mathcal{S}\}$. If $\gamma(\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}') = \gamma^\circ(\|\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'\|)$ for all $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$, $Y(\cdot)$ is said to be *isotropic*. Otherwise, $Y(\cdot)$ is said to be *anisotropic*.

A semivariogram is anisotropic if at least two directional semivariograms differ (Gaetan and Guyon, 2010, Sec. 1.3.4). Kitanidis (1997, Sec. 5.3) presents the steps for obtaining directional experimental variograms. Anisotropy subtypes are discussed in texts such as Zimmerman (1993), Eriksson and Siska (2000) and Zhu and Zhang (2013). When a one-dimensional second-order stationary spatial process is isotropic, its covariogram and correlogram functions are written as $C(\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}') = C^\circ(\|\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'\|)$ and $\rho(\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}') = \rho^\circ(\|\underset{\sim}{\mathbf{s}} - \underset{\sim}{\mathbf{s}}'\|)$ for all $\underset{\sim}{\mathbf{s}}, \underset{\sim}{\mathbf{s}}' \in \mathcal{S}$, respectively (Cressie, 1993; Carvalho and Natário, 2008). Formulas for the empirical covariogram, correlogram and semivariogram are presented in Isaaks and Srivastava (1989, Chapter 4) and Schuenemeyer and Drew (2011, Sec. 6.4).

A.3 Marginal likelihood

Recall that the likelihood function of $\boldsymbol{\theta} = \{\boldsymbol{\beta}_0, \boldsymbol{\beta}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}\}$, denoted by $l(\boldsymbol{\theta}; \mathbf{y})$, is presented in Equation (2.5), where $\boldsymbol{\beta} = \{\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T\}$ and $\mathbf{Y} = \{\mathbf{Y}_1, \ldots, \mathbf{Y}_T\}$. Based on Migon et al. (2014,

Sec. 2.6.1), the marginal likelihood of $\{V, \phi, \mathbf{D}, \boldsymbol{\Sigma}\} = \boldsymbol{\theta} \backslash \{\boldsymbol{\beta}_0, \boldsymbol{\beta}\}$ is given by:

$$\begin{aligned} l(V, \phi, \mathbf{D}, \boldsymbol{\Sigma}; \mathbf{y}) &= f(\mathbf{y} \mid V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\ &= \int_{\mathbb{R}^{p \times q}} \int_{\mathbb{R}^{p \times q}} \cdots \int_{\mathbb{R}^{p \times q}} f(\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T, \mathbf{y} \mid V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \partial\boldsymbol{\beta}_0 \partial\boldsymbol{\beta}_1 \cdots \partial\boldsymbol{\beta}_T \\ &= \int_{\mathbb{R}^{p \times q}} \int_{\mathbb{R}^{p \times q}} \cdots \int_{\mathbb{R}^{p \times q}} f(\boldsymbol{\beta}_0, \boldsymbol{\beta} \mid V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) f(\mathbf{y} \mid \boldsymbol{\theta}) \partial\boldsymbol{\beta}_0 \partial\boldsymbol{\beta}_1 \cdots \partial\boldsymbol{\beta}_T \\ &= \int_{\mathbb{R}^{p \times q}} \int_{\mathbb{R}^{p \times q}} \cdots \int_{\mathbb{R}^{p \times q}} f(\boldsymbol{\beta}_0, \boldsymbol{\beta} \mid V, \boldsymbol{\Sigma}) l(\boldsymbol{\theta}; \mathbf{y}) \partial\boldsymbol{\beta}_0 \partial\boldsymbol{\beta}_1 \cdots \partial\boldsymbol{\beta}_T, \end{aligned}$$

where

$$\begin{aligned} f(\boldsymbol{\beta}_0, \boldsymbol{\beta} \mid V, \boldsymbol{\Sigma}) &= f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_1 \mid \boldsymbol{\beta}_0, V, \boldsymbol{\Sigma}) f(\boldsymbol{\beta}_2 \mid \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, V, \boldsymbol{\Sigma}) \cdots f(\boldsymbol{\beta}_T \mid \boldsymbol{\beta}_0, \ldots, \boldsymbol{\beta}_{T-1}, V, \boldsymbol{\Sigma}) \\ &= f(\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \prod_{t=1}^{T} f(\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}) \\ &\propto [\det(V \cdot \mathbf{W})]^{-\frac{Tq}{2}} [\det \boldsymbol{\Sigma}]^{-\frac{Tp}{2}} [\det(V \cdot \mathbf{C}_0)]^{-\frac{q}{2}} [\det \boldsymbol{\Sigma}]^{-\frac{p}{2}} \\ &\times \exp\Bigg\{ -\frac{1}{2} \operatorname{tr} \Big[\Big\{ \boldsymbol{\beta}_0^\intercal (V \cdot \mathbf{C}_0)^{-1} \boldsymbol{\beta}_0 + \sum_{t=1}^{T} \boldsymbol{\beta}_t^\intercal (V \cdot \mathbf{W})^{-1} \boldsymbol{\beta}_t - \\ &\qquad - \boldsymbol{\beta}_0^\intercal (V \cdot \mathbf{C}_0)^{-1} \mathbf{M}_0 - \sum_{t=1}^{T} \boldsymbol{\beta}_t^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_t \boldsymbol{\beta}_{t-1} - \\ &\qquad - \mathbf{M}_0^\intercal (V \cdot \mathbf{C}_0)^{-1} \boldsymbol{\beta}_0 - \sum_{t=1}^{T} \boldsymbol{\beta}_{t-1}^\intercal \mathbf{G}_t^\intercal (V \cdot \mathbf{W})^{-1} \boldsymbol{\beta}_t + \\ &\qquad + \mathbf{M}_0^\intercal (V \mathbf{C}_0)^{-1} \mathbf{M}_0 + \sum_{t=1}^{T} \boldsymbol{\beta}_{t-1}^\intercal \mathbf{G}_t^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_t \boldsymbol{\beta}_{t-1} \Big\} \boldsymbol{\Sigma}^{-1} \Big] \Bigg\} \end{aligned}$$

and

$$\begin{aligned} l(\boldsymbol{\theta}; \mathbf{y}) &= \prod_{t=1}^{T} f(\mathbf{y}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\ &\propto [\det(V \cdot \mathbf{B})]^{-\frac{Tq}{2}} [\det \boldsymbol{\Sigma}]^{-\frac{TN}{2}} \\ &\times \exp\Bigg\{ -\frac{1}{2} \operatorname{tr} \Big[\Big\{ \sum_{t=1}^{T} \mathbf{y}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{y}_t - \sum_{t=1}^{T} \mathbf{y}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{X}_t \boldsymbol{\beta}_t - \\ &\qquad - \sum_{t=1}^{T} \boldsymbol{\beta}_t^\intercal \mathbf{X}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{y}_t + \sum_{t=1}^{T} \boldsymbol{\beta}_t^\intercal \mathbf{X}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{X}_t \boldsymbol{\beta}_t \Big\} \boldsymbol{\Sigma}^{-1} \Big] \Bigg\}. \end{aligned}$$

To simplify the notation, for $t \in \{1, \ldots, T\}$ define the following quantities:

$$\mathbf{v}_t = \mathbf{G}_t^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{U}_t \Leftrightarrow \mathbf{v}_t^\intercal = \mathbf{U}_t (V \cdot \mathbf{W})^{-1} \mathbf{G}_t$$

and

$$\mathbf{w}_t = \mathbf{X}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{y}_t \Leftrightarrow \mathbf{w}_t^\intercal = \mathbf{y}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{X}_t,$$

where $\mathbf{U}_1, \ldots, \mathbf{U}_T$ are $p \times p$ symmetric matrices that will be presented in the next subsections.

As one may write

$$\sum_{t=1}^{T} \boldsymbol{\beta}_{t-1}^\intercal \mathbf{G}_t^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_t \boldsymbol{\beta}_{t-1} = \boldsymbol{\beta}_0^\intercal \mathbf{G}_1^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_1 \boldsymbol{\beta}_0 + \sum_{t=1}^{T-1} \boldsymbol{\beta}_t^\intercal \mathbf{G}_{t+1}^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_{t+1} \boldsymbol{\beta}_t,$$

we have the following factorization:

$$f(\boldsymbol{\beta}_0, \boldsymbol{\beta} \mid V, \boldsymbol{\Sigma}) l(\boldsymbol{\theta}; \mathbf{y}) \propto \gamma \cdot \lambda_0 \cdot \left[\prod_{t=1}^{T-1} \lambda_t \right] \cdot \lambda_T,$$

where:

$$\begin{aligned}
\gamma &= \gamma(V, \phi, \mathbf{D}, \boldsymbol{\Sigma}; \mathbf{y}) \\
&= [\det(V \cdot \mathbf{B})]^{-\frac{Tq}{2}} [\det(V \cdot \mathbf{W})]^{-\frac{Tq}{2}} [\det(V \cdot \mathbf{C}_0)]^{-\frac{q}{2}} [\det \boldsymbol{\Sigma}]^{-\frac{TN+Tp+p}{2}} \\
&\times \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[\left\{ \mathbf{M}_0^\intercal (V \cdot \mathbf{C}_0)^{-1} \mathbf{M}_0 + \sum_{t=1}^{T} \mathbf{y}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{y}_t \right\} \boldsymbol{\Sigma}^{-1} \right] \right\},
\end{aligned}$$

$$\begin{aligned}
\lambda_0 &= \lambda(\boldsymbol{\beta}_0; V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\
&= \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[\left\{ \boldsymbol{\beta}_0^\intercal \left[(V \cdot \mathbf{C}_0)^{-1} + \mathbf{G}_1^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_1 \right] \boldsymbol{\beta}_0 - \right. \right. \right. \\
&\qquad \left. \left. \left. - \boldsymbol{\beta}_0^\intercal (V \cdot \mathbf{C}_0)^{-1} \mathbf{M}_0 - \mathbf{M}_0^\intercal (V \cdot \mathbf{C}_0)^{-1} \boldsymbol{\beta}_0 \right\} \boldsymbol{\Sigma}^{-1} \right] \right\},
\end{aligned}$$

$$\begin{aligned}
\lambda_t &= \lambda(\boldsymbol{\beta}_t; \boldsymbol{\beta}_{t-1}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_t) \\
&= \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[\left\{ \boldsymbol{\beta}_t^\intercal \left[\mathbf{X}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{X}_t + (V \cdot \mathbf{W})^{-1} + \mathbf{G}_{t+1}^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_{t+1} \right] \boldsymbol{\beta}_t - \right. \right. \right. \\
&\qquad - \boldsymbol{\beta}_t^\intercal \left[\mathbf{X}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{y}_t + (V \cdot \mathbf{W})^{-1} \mathbf{G}_t \boldsymbol{\beta}_{t-1} \right] - \\
&\qquad \left. \left. \left. - \left[\mathbf{y}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{X}_t + \boldsymbol{\beta}_{t-1}^\intercal \mathbf{G}_t^\intercal (V \cdot \mathbf{W})^{-1} \right] \boldsymbol{\beta}_t \right\} \boldsymbol{\Sigma}^{-1} \right] \right\}, \quad t \in \{1, \ldots, T-1\}, \\
&= \exp\left\{ -\frac{1}{2} \operatorname{tr}\left[\left\{ \boldsymbol{\beta}_t^\intercal \left[\mathbf{X}_t^\intercal (V \cdot \mathbf{B})^{-1} \mathbf{X}_t + (V \cdot \mathbf{W})^{-1} + \mathbf{G}_{t+1}^\intercal (V \cdot \mathbf{W})^{-1} \mathbf{G}_{t+1} \right] \boldsymbol{\beta}_t - \right. \right. \right. \\
&\qquad - \boldsymbol{\beta}_t^\intercal \left[(V \cdot \mathbf{W})^{-1} \mathbf{G}_t \boldsymbol{\beta}_{t-1} + \mathbf{w}_t \right] - \\
&\qquad \left. \left. \left. - \left[\boldsymbol{\beta}_{t-1}^\intercal \mathbf{G}_t^\intercal (V \cdot \mathbf{W})^{-1} + \mathbf{w}_t^\intercal \right] \boldsymbol{\beta}_t \right\} \boldsymbol{\Sigma}^{-1} \right] \right\}, \quad t \in \{1, \ldots, T-1\},
\end{aligned}$$

and

$$\begin{aligned}
\lambda_T &= \lambda(\boldsymbol{\beta}_T; \boldsymbol{\beta}_{T-1}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_T) \\
&= \exp\Bigg\{ -\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_T^\intercal\left[\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T + (V\cdot\mathbf{W})^{-1}\right]\boldsymbol{\beta}_T - \\
&\qquad\qquad -\boldsymbol{\beta}_T^\intercal\left[\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_T + (V\cdot\mathbf{W})^{-1}\mathbf{G}_T\boldsymbol{\beta}_{T-1}\right] - \\
&\qquad\qquad -\left[\mathbf{y}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T + \boldsymbol{\beta}_{T-1}^\intercal\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\right]\boldsymbol{\beta}_T\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\} \\
&= \exp\Bigg\{ -\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_T^\intercal\left[\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T + (V\cdot\mathbf{W})^{-1}\right]\boldsymbol{\beta}_T - \\
&\qquad\qquad -\boldsymbol{\beta}_T^\intercal\left[(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\boldsymbol{\beta}_{T-1} + \mathbf{w}_T\right] - \\
&\qquad\qquad -\left[\boldsymbol{\beta}_{T-1}^\intercal\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1} + \mathbf{w}_T^\intercal\right]\boldsymbol{\beta}_T\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}.
\end{aligned}$$

Thus, the marginal likelihood of $\{V, \phi, \mathbf{D}, \boldsymbol{\Sigma}\} = \boldsymbol{\theta}\backslash\{\boldsymbol{\beta}_0, \boldsymbol{\beta}\}$ may be written as the following nested integrals to solve:

$$l(V, \phi, \mathbf{D}, \boldsymbol{\Sigma}; \mathbf{y}) \propto \gamma \cdot \int_{\mathbb{R}^{p\times q}} \lambda_0 \left[\int_{\mathbb{R}^{p\times q}} \lambda_1 \left[\cdots \left[\int_{\mathbb{R}^{p\times q}} \lambda_{T-1} \left[\int_{\mathbb{R}^{p\times q}} \lambda_T \boldsymbol{\partial}\boldsymbol{\beta}_T \right] \boldsymbol{\partial}\boldsymbol{\beta}_{T-1} \right] \cdots \right] \boldsymbol{\partial}\boldsymbol{\beta}_1 \right] \boldsymbol{\partial}\boldsymbol{\beta}_0.$$

Integrating out with respect to $\boldsymbol{\beta}_T$

The term λ_T can be identified as the kernel of the density of a matrix-normal distribution with mean matrix $\mathbf{U}_T\mathbf{u}_T$, left covariance matrix $\mathbf{U}_T$ and right covariance matrix $\boldsymbol{\Sigma}$, where

$$\mathbf{U}_T = \left[\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T + (V\cdot\mathbf{W})^{-1}\right]^{-1}$$

and

$$\begin{aligned}
\mathbf{u}_T &= \mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_T + (V\cdot\mathbf{W})^{-1}\mathbf{G}_T\boldsymbol{\beta}_{T-1} \\
&= (V\cdot\mathbf{W})^{-1}\mathbf{G}_T\boldsymbol{\beta}_{T-1} + \mathbf{w}_T.
\end{aligned}$$

Since

$$\begin{aligned}
(\boldsymbol{\beta}_T - \mathbf{U}_T\mathbf{u}_T)^\intercal\mathbf{U}_T^{-1}(\boldsymbol{\beta}_T - \mathbf{U}_T\mathbf{u}_T)\boldsymbol{\Sigma}^{-1} &= \{(\boldsymbol{\beta}_T^\intercal - \mathbf{u}_T^\intercal\mathbf{U}_T^\intercal)\mathbf{U}_T^{-1}(\boldsymbol{\beta}_T - \mathbf{U}_T\mathbf{u}_T)\}\boldsymbol{\Sigma}^{-1} \\
&= \{\boldsymbol{\beta}_T^\intercal\mathbf{U}_T^{-1}\boldsymbol{\beta}_T - \boldsymbol{\beta}_T^\intercal\mathbf{u}_T - \mathbf{u}_T^\intercal\boldsymbol{\beta}_T + \mathbf{u}_T^\intercal\mathbf{U}_T\mathbf{u}_T\}\boldsymbol{\Sigma}^{-1},
\end{aligned}$$

multiplying and dividing λ_T by $\exp\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_T^\intercal\mathbf{U}_T\mathbf{u}_T\}\boldsymbol{\Sigma}^{-1}\big]\}$, we have:

$$\begin{aligned}
\int\limits_{\mathbb{R}^{p\times q}} \lambda_T\boldsymbol{\partial}\boldsymbol{\beta}_T &= \frac{\int\limits_{\mathbb{R}^{p\times q}} \lambda_T \exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_T^\intercal\mathbf{U}_T\mathbf{u}_T\}\boldsymbol{\Sigma}^{-1}\big]\right\}\boldsymbol{\partial}\boldsymbol{\beta}_T}{\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_T^\intercal\mathbf{U}_T\mathbf{u}_T\}\boldsymbol{\Sigma}^{-1}\big]\right\}} \\
&\propto [\det\mathbf{U}_T]^{\frac{q}{2}}\,[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{-\mathbf{u}_T^\intercal\mathbf{U}_T\mathbf{u}_T\}\boldsymbol{\Sigma}^{-1}\big]\right\} \\
&= [\det\mathbf{U}_T]^{\frac{q}{2}}\,[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\,\gamma_T^*\lambda_{T-1}^*,
\end{aligned}$$

where

$$\begin{aligned}
\gamma_T^* &= \gamma^*(V,\phi,\mathbf{D},\boldsymbol{\Sigma};\mathbf{y}_T) \\
&= \exp\left\{-\frac{1}{2}\operatorname{tr}\left[\left\{-\mathbf{y}_T^\intercal\left[(V\cdot\mathbf{B})^{-1}\mathbf{X}_T\mathbf{U}_T\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\right]\mathbf{y}_T\right\}\boldsymbol{\Sigma}^{-1}\right]\right\} \\
&= \exp\left\{-\frac{1}{2}\operatorname{tr}\left[\left\{-\mathbf{w}_T^\intercal\mathbf{U}_T\mathbf{w}_T\right\}\boldsymbol{\Sigma}^{-1}\right]\right\}
\end{aligned}$$

and

$$\begin{aligned}
\lambda_{T-1}^* &= \lambda^*(\boldsymbol{\beta}_{T-1};V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_T) \\
&= \exp\Bigg\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_{T-1}^\intercal\left[-\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\right]\boldsymbol{\beta}_{T-1} - \\
&\qquad -\boldsymbol{\beta}_{T-1}^\intercal\left[\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_T\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_T\right] - \\
&\qquad -\left[\mathbf{y}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\right]\boldsymbol{\beta}_{T-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\} \\
&= \exp\Bigg\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_{T-1}^\intercal\left[-\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\right]\boldsymbol{\beta}_{T-1} - \\
&\qquad -\boldsymbol{\beta}_{T-1}^\intercal\left[\mathbf{v}_T\mathbf{w}_T\right] - \left[\mathbf{w}_T^\intercal\mathbf{v}_T^\intercal\right]\boldsymbol{\beta}_{T-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}.
\end{aligned}$$

Integrating out with respect to $\boldsymbol{\beta}_{T-1}$

The term

$$\begin{aligned}
\lambda_{T-1}\lambda^*_{T-1} &= \lambda(\boldsymbol{\beta}_{T-1};\boldsymbol{\beta}_{T-2},V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_{T-1})\lambda^*(\boldsymbol{\beta}_{T-1};V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_T) \\
&\propto \exp\Bigg\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}^\intercal_{T-1}\Big[\mathbf{X}^\intercal_{T-1}(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}+(V\cdot\mathbf{W})^{-1}+ \\
&\qquad +\mathbf{G}^\intercal_T\left\{(V\cdot\mathbf{W})^{-1}-(V\cdot\mathbf{W})^{-1}\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\right\}\mathbf{G}_T\Big]\boldsymbol{\beta}_{T-1}- \\
&\qquad -\boldsymbol{\beta}^\intercal_{T-1}\Big[\mathbf{X}^\intercal_{T-1}(V\cdot\mathbf{B})^{-1}\mathbf{y}_{T-1}+(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\boldsymbol{\beta}_{T-2}+ \\
&\qquad +\mathbf{G}^\intercal_T(V\cdot\mathbf{W})^{-1}\mathbf{U}_T\mathbf{X}^\intercal_T(V\cdot\mathbf{B})^{-1}\mathbf{y}_T\Big]- \\
&\qquad -\Big[\mathbf{y}^\intercal_{T-1}(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}+\boldsymbol{\beta}^\intercal_{T-2}\mathbf{G}^\intercal_{T-1}(V\cdot\mathbf{W})^{-1}+ \\
&\qquad +\mathbf{y}^\intercal_T(V\cdot\mathbf{B})^{-1}\mathbf{X}_T\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\Big]\boldsymbol{\beta}_{T-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\} \\
&= \exp\Bigg\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}^\intercal_{T-1}\Big[\mathbf{X}^\intercal_{T-1}(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}+(V\cdot\mathbf{W})^{-1}+ \\
&\qquad +\mathbf{G}^\intercal_T\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p-\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_T\Big]\boldsymbol{\beta}_{T-1}- \\
&\qquad -\boldsymbol{\beta}^\intercal_{T-1}\Big[(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\boldsymbol{\beta}_{T-2}+\mathbf{w}_{T-1}+\mathbf{v}_T\mathbf{w}_T\Big]- \\
&\qquad -\Big[\boldsymbol{\beta}^\intercal_{T-2}\mathbf{G}^\intercal_{T-1}(V\cdot\mathbf{W})^{-1}+\mathbf{w}^\intercal_{T-1}+\mathbf{w}^\intercal_T\mathbf{v}^\intercal_T\Big]\boldsymbol{\beta}_{T-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}
\end{aligned}$$

can be identified as the kernel of the density of a matrix-normal distribution with mean matrix $\mathbf{U}_{T-1}\mathbf{u}_{T-1}$, left covariance matrix $\mathbf{U}_{T-1}$ and right covariance matrix $\boldsymbol{\Sigma}$, where

$$\mathbf{U}_{T-1}=\left[\mathbf{X}^\intercal_{T-1}(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}+(V\cdot\mathbf{W})^{-1}+\mathbf{G}^\intercal_T\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p-\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_T\right]^{-1}$$

and

$$\begin{aligned}
\mathbf{u}_{T-1} &= \mathbf{X}^\intercal_{T-1}(V\cdot\mathbf{B})^{-1}\mathbf{y}_{T-1}+(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\boldsymbol{\beta}_{T-2}+\mathbf{G}^\intercal_T(V\cdot\mathbf{W})^{-1}\mathbf{U}_T\mathbf{X}^\intercal_T(V\cdot\mathbf{B})^{-1}\mathbf{y}_T \\
&= (V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\boldsymbol{\beta}_{T-2}+\mathbf{w}_{T-1}+\mathbf{v}_T\mathbf{w}_T.
\end{aligned}$$

Multiplying and dividing $\lambda_{T-1}\lambda^*_{T-1}$ by $\exp\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}^\intercal_{T-1}\mathbf{U}_{T-1}\mathbf{u}_{T-1}\}\boldsymbol{\Sigma}^{-1}\big]\}$, we have:

$$\begin{aligned}
\int\limits_{\mathbb{R}^{p\times q}}\lambda_{T-1}\lambda^*_{T-1}\boldsymbol{\partial}\boldsymbol{\beta}_{T-1} &= \frac{\int\limits_{\mathbb{R}^{p\times q}}\lambda_{T-1}\lambda^*_{T-1}\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}^\intercal_{T-1}\mathbf{U}_{T-1}\mathbf{u}_{T-1}\}\boldsymbol{\Sigma}^{-1}\big]\right\}\boldsymbol{\partial}\boldsymbol{\beta}_{T-1}}{\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}^\intercal_{T-1}\mathbf{U}_{T-1}\mathbf{u}_{T-1}\}\boldsymbol{\Sigma}^{-1}\big]\right\}} \\
&\propto [\det\mathbf{U}_{T-1}]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{-\mathbf{u}^\intercal_{T-1}\mathbf{U}_{T-1}\mathbf{u}_{T-1}\}\boldsymbol{\Sigma}^{-1}\big]\right\} \\
&= [\det\mathbf{U}_{T-1}]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\gamma^*_{T-1}\lambda^*_{T-2},
\end{aligned}$$

where

$$
\begin{aligned}
\gamma^*_{T-1} &= \gamma^*(V,\phi,\mathbf{D},\boldsymbol{\Sigma};\mathbf{y}_{T-1},\mathbf{y}_T) \\
&= \exp\Bigg\{-\frac{1}{2}\,\mathrm{tr}\Big[\Big\{-\mathbf{y}_{T-1}^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}\mathbf{U}_{T-1}\mathbf{X}_{T-1}^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_{T-1} - \\
&\qquad -\mathbf{y}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\mathbf{U}_{T-1}\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_T\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_T - \\
&\qquad -\mathbf{y}_{T-1}^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}\mathbf{U}_{T-1}\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_T\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_T - \\
&\qquad -\mathbf{y}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\mathbf{U}_{T-1}\mathbf{X}_{T-1}^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_{T-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\} \\
&= \exp\Bigg\{-\frac{1}{2}\,\mathrm{tr}\Big[-\Big\{\mathbf{w}_{T-1}^\intercal\mathbf{U}_{T-1}\mathbf{w}_{T-1} + \mathbf{w}_{T-1}^\intercal\mathbf{U}_{T-1}\mathbf{v}_T\mathbf{w}_T + \\
&\qquad +\mathbf{w}_T^\intercal\mathbf{v}_T^\intercal\mathbf{U}_{T-1}\mathbf{w}_{T-1} + \mathbf{w}_T^\intercal\mathbf{v}_T^\intercal\mathbf{U}_{T-1}\mathbf{v}_T\mathbf{w}_T\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}
\end{aligned}
$$

and

$$
\begin{aligned}
\lambda^*_{T-2} &= \lambda^*(\boldsymbol{\beta}_{T-2};V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_{T-1},\mathbf{y}_T) \\
&= \exp\Bigg\{-\frac{1}{2}\,\mathrm{tr}\Big[\Big\{\boldsymbol{\beta}_{T-2}^\intercal\left[-\mathbf{G}_{T-1}^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_{T-1}(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\right]\boldsymbol{\beta}_{T-2} - \\
&\qquad -\boldsymbol{\beta}_{T-2}^\intercal\Big[\mathbf{G}_{T-1}^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_{T-1}\mathbf{X}_{T-1}^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_{T-1} + \\
&\qquad\qquad +\mathbf{G}_{T-1}^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_{T-1}\mathbf{G}_T^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_T\mathbf{X}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_T\Big] - \\
&\qquad -\Big[\mathbf{y}_T^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_T\mathbf{U}_T(V\cdot\mathbf{W})^{-1}\mathbf{G}_T\mathbf{U}_{T-1}(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1} + \\
&\qquad\quad +\mathbf{y}_{T-1}^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-1}\mathbf{U}_{T-1}^{-1}(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\Big]\boldsymbol{\beta}_{T-2}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\} \\
&= \exp\Bigg\{-\frac{1}{2}\,\mathrm{tr}\Big[\Big\{\boldsymbol{\beta}_{T-2}^\intercal\left[-\mathbf{G}_{T-1}^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_{T-1}(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-1}\right]\boldsymbol{\beta}_{T-2} - \\
&\qquad -\boldsymbol{\beta}_{T-2}^\intercal\left[\mathbf{v}_{T-1}\mathbf{w}_{T-1} + \mathbf{v}_{T-1}\mathbf{v}_T\mathbf{w}_T\right] \\
&\qquad -\left[\mathbf{w}_{T-1}^\intercal\mathbf{v}_{T-1}^\intercal + \mathbf{w}_T^\intercal\mathbf{v}_T^\intercal\mathbf{v}_{T-1}^\intercal\right]\boldsymbol{\beta}_{T-2}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}.
\end{aligned}
$$

Integrating out with respect to $\boldsymbol{\beta}_{T-2}$

The term

$$\begin{aligned}
\lambda_{T-2}\lambda^*_{T-2} &= \lambda(\boldsymbol{\beta}_{T-2};\boldsymbol{\beta}_{T-3},V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_{T-2})\lambda^*(\boldsymbol{\beta}_{T-2};V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_{T-1},\mathbf{y}_T)\\
&\propto \exp\Bigg\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}^\intercal_{T-2}\Big[\mathbf{X}^\intercal_{T-2}(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-2}+(V\cdot\mathbf{W})^{-1}+\\
&\qquad +\mathbf{G}^\intercal_{T-1}\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p-\mathbf{U}_{T-1}(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_{T-1}\Big]\boldsymbol{\beta}_{T-2}-\\
&\qquad -\boldsymbol{\beta}^\intercal_{T-2}\Big[(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-2}\boldsymbol{\beta}_{T-3}+\mathbf{w}_{T-2}+\sum_{t=T-1}^{T}\left(\prod_{t'=T-1}^{t}\mathbf{v}_{t'}\right)\mathbf{w}_t\Big]-\\
&\qquad -\Big[\boldsymbol{\beta}^\intercal_{T-3}\mathbf{G}^\intercal_{T-2}(V\cdot\mathbf{W})^{-1}+\mathbf{w}^\intercal_{T-2}+\sum_{t=T-1}^{T}\mathbf{w}^\intercal_t\left(\prod_{t'=t}^{T-1}\mathbf{v}^\intercal_{t'}\right)\Big]\boldsymbol{\beta}_{T-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}
\end{aligned}$$

can be identified as the kernel of the density of a matrix-normal distribution with mean matrix $\mathbf{U}_{T-2}\mathbf{u}_{T-2}$, left covariance matrix $\mathbf{U}_{T-2}$ and right covariance matrix $\boldsymbol{\Sigma}$, where

$$\mathbf{U}_{T-2}=\Big[\mathbf{X}^\intercal_{T-2}(V\cdot\mathbf{B})^{-1}\mathbf{X}_{T-2}+(V\cdot\mathbf{W})^{-1}+\mathbf{G}^\intercal_{T-1}\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p-\mathbf{U}_{T-1}(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_{T-1}\Big]^{-1}$$

and

$$\mathbf{u}_{T-2}=(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-2}\boldsymbol{\beta}_{T-3}+\mathbf{w}_{T-2}+\sum_{t=T-1}^{T}\left(\prod_{t'=T-1}^{t}\mathbf{v}_{t'}\right)\mathbf{w}_t.$$

Multiplying and dividing $\lambda_{T-2}\lambda^*_{T-2}$ by $\exp\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}^\intercal_{T-2}\mathbf{U}_{T-2}\mathbf{u}_{T-2}\}\boldsymbol{\Sigma}^{-1}\big]\}$, we have:

$$\begin{aligned}
\int_{\mathbb{R}^{p\times q}}\lambda_{T-2}\lambda^*_{T-2}\boldsymbol{\partial}\boldsymbol{\beta}_{T-2} &= \frac{\int_{\mathbb{R}^{p\times q}}\lambda_{T-2}\lambda^*_{T-2}\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}^\intercal_{T-2}\mathbf{U}_{T-2}\mathbf{u}_{T-2}\}\boldsymbol{\Sigma}^{-1}\big]\right\}\boldsymbol{\partial}\boldsymbol{\beta}_{T-2}}{\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}^\intercal_{T-2}\mathbf{U}_{T-2}\mathbf{u}_{T-2}\}\boldsymbol{\Sigma}^{-1}\big]\right\}}\\
&\propto [\det\mathbf{U}_{T-2}]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{-\mathbf{u}^\intercal_{T-2}\mathbf{U}_{T-2}\mathbf{u}_{T-2}\}\boldsymbol{\Sigma}^{-1}\big]\right\}\\
&= [\det\mathbf{U}_{T-2}]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\gamma^*_{T-2}\lambda^*_{T-3},
\end{aligned}$$

where

$$\begin{aligned}
\gamma^*_{T-2} &= \gamma^*(V,\phi,\mathbf{D},\boldsymbol{\Sigma};\mathbf{y}_{T-2},\mathbf{y}_{T-1},\mathbf{y}_T)\\
&= \exp\Bigg\{-\frac{1}{2}\operatorname{tr}\Big[-\Big\{\mathbf{w}^\intercal_{T-2}\mathbf{U}_{T-2}\mathbf{w}_{T-2}+\mathbf{w}^\intercal_{T-2}\mathbf{U}_{T-2}\left[\sum_{t=T-1}^{T}\left(\prod_{t'=T-1}^{t}\mathbf{v}_{t'}\right)\mathbf{w}_t\right]+\\
&\qquad +\left[\sum_{t=T-1}^{T}\mathbf{w}^\intercal_t\left(\prod_{t'=t}^{T-1}\mathbf{v}^\intercal_{t'}\right)\right]\mathbf{U}_{T-2}\left[\sum_{t=T-1}^{T}\left(\prod_{t'=T-1}^{t}\mathbf{v}_{t'}\right)\mathbf{w}_t\right]+\\
&\qquad +\left[\sum_{t=T-1}^{T}\mathbf{w}^\intercal_t\left(\prod_{t'=t}^{T-1}\mathbf{v}^\intercal_{t'}\right)\right]\mathbf{U}_{T-2}\mathbf{w}_{T-2}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}
\end{aligned}$$

and

$$\begin{aligned}
\lambda^*_{T-3} &= \lambda^*(\boldsymbol{\beta}_{T-3}; V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_{T-2}, \mathbf{y}_{T-1}, \mathbf{y}_T) \\
&= \exp\Bigg\{ -\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}^\intercal_{T-3}\left[-\mathbf{G}^\intercal_{T-2}(V\cdot\mathbf{W})^{-1}\mathbf{U}_{T-2}(V\cdot\mathbf{W})^{-1}\mathbf{G}_{T-2}\right]\boldsymbol{\beta}_{T-3} - \\
&\qquad -\boldsymbol{\beta}^\intercal_{T-3}\left[\sum_{t=T-2}^{T}\left(\prod_{t'=T-2}^{t}\mathbf{v}_{t'}\right)\mathbf{w}_t\right] - \\
&\qquad -\left[\sum_{t=T-2}^{T}\mathbf{w}^\intercal_t\left(\prod_{t'=t}^{T-2}\mathbf{v}^\intercal_{t'}\right)\right]\boldsymbol{\beta}_{T-3}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}.
\end{aligned}$$

Integrating out with respect to $\boldsymbol{\beta}_t$ for $t \in \{1, \ldots, T-1\}$

For $t \in \{0, 1, \ldots, T-1\}$, we state the following:

$$\begin{aligned}
\lambda^*_t &= \lambda^*(\boldsymbol{\beta}_t; V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_{t+1}, \ldots, \mathbf{y}_T) \\
&= \exp\Bigg\{ -\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}^\intercal_t\left[-\mathbf{G}^\intercal_{t+1}(V\cdot\mathbf{W})^{-1}\mathbf{U}_{t+1}(V\cdot\mathbf{W})^{-1}\mathbf{G}_{t+1}\right]\boldsymbol{\beta}_t - \\
&\qquad -\boldsymbol{\beta}^\intercal_t\left[\sum_{t'=t+1}^{T}\left(\prod_{t''=t+1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\right] - \\
&\qquad -\left[\sum_{t'=t+1}^{T}\mathbf{w}^\intercal_{t'}\left(\prod_{t''=t'}^{t+1}\mathbf{v}^\intercal_{t''}\right)\right]\boldsymbol{\beta}_t\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}.
\end{aligned}$$

It can be proved by induction. Suppose that λ^*_t is true for $t = k \in \{1, \ldots, T-1\}$, in order to verify its validity for $t = k-1$. The term

$$\begin{aligned}
\lambda_k\lambda^*_k &= \lambda(\boldsymbol{\beta}_k; \boldsymbol{\beta}_{k-1}, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_k)\lambda^*(\boldsymbol{\beta}_k; V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_{k+1}, \ldots, \mathbf{y}_T) \\
&\propto \exp\Bigg\{ -\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}^\intercal_k\Big[\mathbf{X}^\intercal_k(V\cdot\mathbf{B})^{-1}\mathbf{X}_k + (V\cdot\mathbf{W})^{-1} + \\
&\qquad +\mathbf{G}^\intercal_{k+1}\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p - \mathbf{U}_{k+1}(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_{k+1}\Big]\boldsymbol{\beta}_k - \\
&\qquad -\boldsymbol{\beta}^\intercal_k\Big[(V\cdot\mathbf{W})^{-1}\mathbf{G}_k\boldsymbol{\beta}_{k-1} + \mathbf{w}_k + \sum_{t'=k+1}^{T}\left(\prod_{t''=k+1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\Big] - \\
&\qquad -\Big[\boldsymbol{\beta}^\intercal_{k-1}\mathbf{G}^\intercal_k(V\cdot\mathbf{W})^{-1} + \mathbf{w}^\intercal_k + \sum_{t'=k+1}^{T}\mathbf{w}^\intercal_{t'}\left(\prod_{t''=t'}^{k+1}\mathbf{v}^\intercal_{t''}\right)\Big]\boldsymbol{\beta}_k\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}
\end{aligned}$$

can be identified as the kernel of the density of a matrix-normal distribution with mean matrix $\mathbf{U}_k\mathbf{u}_k$, left covariance matrix $\mathbf{U}_k$ and right covariance matrix $\boldsymbol{\Sigma}$, where

$$\mathbf{U}_k = \Big[\mathbf{X}^\intercal_k(V\cdot\mathbf{B})^{-1}\mathbf{X}_k + (V\cdot\mathbf{W})^{-1} + \mathbf{G}^\intercal_{k+1}\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p - \mathbf{U}_{k+1}(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_{k+1}\Big]^{-1}$$

and

$$\mathbf{u}_k = (V \cdot \mathbf{W})^{-1}\mathbf{G}_k\boldsymbol{\beta}_{k-1} + \mathbf{w}_k + \sum_{t'=k+1}^{T}\left(\prod_{t''=k+1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}.$$

Multiplying and dividing $\lambda_k\lambda_k^*$ by $\exp\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_k^\intercal\mathbf{U}_k\mathbf{u}_k\}\boldsymbol{\Sigma}^{-1}\big]\}$, we have:

$$\begin{aligned}
\int\limits_{\mathbb{R}^{p\times q}} \lambda_k\lambda_k^*\boldsymbol{\partial}\boldsymbol{\beta}_k &= \frac{\int\limits_{\mathbb{R}^{p\times q}} \lambda_k\lambda_k^* \exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_k^\intercal\mathbf{U}_k\mathbf{u}_k\}\boldsymbol{\Sigma}^{-1}\big]\right\}\boldsymbol{\partial}\boldsymbol{\beta}_k}{\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_k^\intercal\mathbf{U}_k\mathbf{u}_k\}\boldsymbol{\Sigma}^{-1}\big]\right\}} \\
&\propto [\det\mathbf{U}_k]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{-\mathbf{u}_k^\intercal\mathbf{U}_k\mathbf{u}_k\}\boldsymbol{\Sigma}^{-1}\big]\right\} \\
&= [\det\mathbf{U}_k]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\gamma_k^*\lambda_{k-1}^*,
\end{aligned}$$

where

$$\begin{aligned}
\gamma_k^* &= \gamma^*(V,\phi,\mathbf{D},\boldsymbol{\Sigma};\mathbf{y}_k,\ldots,\mathbf{y}_T) \\
&= \exp\left\{-\frac{1}{2}\operatorname{tr}\Big[-\Big\{\mathbf{w}_k^\intercal\mathbf{U}_k\mathbf{w}_k + \mathbf{w}_k^\intercal\mathbf{U}_k\left[\sum_{t'=k+1}^{T}\left(\prod_{t''=k+1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\right] + \right. \right. \\
&\qquad + \left[\sum_{t'=k+1}^{T}\mathbf{w}_{t'}^\intercal\left(\prod_{t''=t'}^{k+1}\mathbf{v}_{t''}^\intercal\right)\right]\mathbf{U}_k\left[\sum_{t'=k+1}^{T}\left(\prod_{t''=k+1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\right] + \\
&\qquad \left.\left. + \left[\sum_{t'=k+1}^{T}\mathbf{w}_{t'}^\intercal\left(\prod_{t''=t'}^{k+1}\mathbf{v}_{t''}^\intercal\right)\right]\mathbf{U}_k\mathbf{w}_k\Big\}\boldsymbol{\Sigma}^{-1}\Big]\right\}
\end{aligned}$$

and

$$\begin{aligned}
\lambda_{k-1}^* &= \lambda^*(\boldsymbol{\beta}_{k-1};V,\phi,\mathbf{D},\boldsymbol{\Sigma},\mathbf{y}_k,\mathbf{y}_{k+1},\ldots,\mathbf{y}_T) \\
&= \exp\left\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_{k-1}^\intercal\left[-\mathbf{G}_k^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_k(V\cdot\mathbf{W})^{-1}\mathbf{G}_k\right]\boldsymbol{\beta}_{k-1} - \right. \right. \\
&\qquad -\boldsymbol{\beta}_{k-1}^\intercal\left[\mathbf{v}_k\mathbf{w}_k + \mathbf{v}_k\left\{\sum_{t'=k+1}^{T}\left(\prod_{t''=k+1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\right\}\right] - \\
&\qquad \left.\left. -\left[\mathbf{w}_k^\intercal\mathbf{v}_k^\intercal + \left\{\sum_{t'=k+1}^{T}\mathbf{w}_{t'}^\intercal\left(\prod_{t''=t'}^{k+1}\mathbf{v}_{t''}^\intercal\right)\right\}\mathbf{v}_k^\intercal\right]\boldsymbol{\beta}_{k-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\right\} \\
&= \exp\left\{-\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_{k-1}^\intercal\left[-\mathbf{G}_k^\intercal(V\cdot\mathbf{W})^{-1}\mathbf{U}_k(V\cdot\mathbf{W})^{-1}\mathbf{G}_k\right]\boldsymbol{\beta}_{k-1} - \right. \right. \\
&\qquad -\boldsymbol{\beta}_{k-1}^\intercal\left[\sum_{t'=k}^{T}\left(\prod_{t''=k}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\right] - \\
&\qquad \left.\left. -\left[\sum_{t'=k}^{T}\mathbf{w}_{t'}^\intercal\left(\prod_{t''=t'}^{k}\mathbf{v}_{t''}^\intercal\right)\right]\boldsymbol{\beta}_{k-1}\Big\}\boldsymbol{\Sigma}^{-1}\Big]\right\}.
\end{aligned}$$

Integrating out with respect to $\boldsymbol{\beta}_0$ and writing the final expression

The term

$$\begin{aligned}
\lambda_0\lambda_0^* &= \lambda(\boldsymbol{\beta}_0; V, \phi, \mathbf{D}, \boldsymbol{\Sigma})\lambda^*(\boldsymbol{\beta}_0; V, \phi, \mathbf{D}, \boldsymbol{\Sigma}, \mathbf{y}_1, \ldots, \mathbf{y}_T) \\
&\propto \exp\Bigg\{ -\frac{1}{2}\operatorname{tr}\Big[\Big\{\boldsymbol{\beta}_0^\intercal\Big[(V\cdot\mathbf{C}_0)^{-1} + \\
&\qquad +\mathbf{G}_1^\intercal\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p - \mathbf{U}_1(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_1\Big]\boldsymbol{\beta}_0 - \\
&\qquad -\boldsymbol{\beta}_0^\intercal\Big[(V\cdot\mathbf{C}_0)^{-1}\mathbf{M}_0 + \sum_{t'=1}^{T}\left(\prod_{t''=1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}\Big] - \\
&\qquad -\Big[\mathbf{M}_0^\intercal(V\cdot\mathbf{C}_0)^{-1} + \sum_{t'=1}^{T}\mathbf{w}_{t'}^\intercal\left(\prod_{t''=t'}^{1}\mathbf{v}_{t''}^\intercal\right)\Big]\boldsymbol{\beta}_0\Big\}\boldsymbol{\Sigma}^{-1}\Big]\Bigg\}
\end{aligned}$$

can be identified as the kernel of the density of a matrix-normal distribution with mean matrix $\mathbf{U}_0\mathbf{u}_0$, left covariance matrix $\mathbf{U}_0$ and right covariance matrix $\boldsymbol{\Sigma}$, where

$$\mathbf{U}_0 = \Big[(V\cdot\mathbf{C}_0)^{-1} + \mathbf{G}_1^\intercal\left\{(V\cdot\mathbf{W})^{-1}\left[\mathbf{I}_p - \mathbf{U}_1(V\cdot\mathbf{W})^{-1}\right]\right\}\mathbf{G}_1\Big]^{-1}$$

and

$$\mathbf{u}_0 = (V\cdot\mathbf{C}_0)^{-1}\mathbf{M}_0 + \sum_{t'=1}^{T}\left(\prod_{t''=1}^{t'}\mathbf{v}_{t''}\right)\mathbf{w}_{t'}.$$

Multiplying and dividing $\lambda_0\lambda_0^*$ by $\exp\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_0^\intercal\mathbf{U}_0\mathbf{u}_0\}\boldsymbol{\Sigma}^{-1}\big]\}$, we have:

$$\begin{aligned}
\int_{\mathbb{R}^{p\times q}}\lambda_0\lambda_0^*\boldsymbol{\partial}\boldsymbol{\beta}_0 &= \frac{\int_{\mathbb{R}^{p\times q}}\lambda_0\lambda_0^*\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_0^\intercal\mathbf{U}_0\mathbf{u}_0\}\boldsymbol{\Sigma}^{-1}\big]\right\}\boldsymbol{\partial}\boldsymbol{\beta}_0}{\exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{\mathbf{u}_0^\intercal\mathbf{U}_0\mathbf{u}_0\}\boldsymbol{\Sigma}^{-1}\big]\right\}} \\
&\propto [\det\mathbf{U}_0]^{\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{\frac{p}{2}}\gamma_0^*,
\end{aligned}$$

where

$$\gamma_0^* = \gamma^*(V, \phi, \mathbf{D}, \boldsymbol{\Sigma}; \mathbf{y}) = \exp\left\{-\frac{1}{2}\operatorname{tr}\big[\{-\mathbf{u}_0^\intercal\mathbf{U}_0\mathbf{u}_0\}\boldsymbol{\Sigma}^{-1}\big]\right\}.$$

Finally, the marginal likelihood of $\{V, \phi, \mathbf{D}, \boldsymbol{\Sigma}\}$ is:

$$\begin{aligned}
l(V, \phi, \mathbf{D}, \boldsymbol{\Sigma}; \mathbf{y}) &\propto [\det(V\cdot\mathbf{B})]^{-\frac{Tq}{2}}[\det(V\cdot\mathbf{W})]^{-\frac{Tq}{2}}[\det(V\cdot\mathbf{C}_0)]^{-\frac{q}{2}}[\det\boldsymbol{\Sigma}]^{-\frac{TN}{2}} \\
&\times \exp\left\{-\frac{1}{2}\operatorname{tr}\left[\left\{\mathbf{M}_0^\intercal(V\cdot\mathbf{C}_0)^{-1}\mathbf{M}_0 + \sum_{t=1}^{T}\mathbf{y}_t^\intercal(V\cdot\mathbf{B})^{-1}\mathbf{y}_t\right\}\boldsymbol{\Sigma}^{-1}\right]\right\} \\
&\times [\det\mathbf{U}_0]^{\frac{q}{2}}\gamma_0^*\cdot\left\{\prod_{t=1}^{T-1}[\det\mathbf{U}_t]^{\frac{q}{2}}\gamma_t^*\right\}\cdot[\det\mathbf{U}_T]^{\frac{q}{2}}\gamma_T^*.
\end{aligned}$$

A.4 Estimating the entries of the 2×2 matrix $\mathbf{\Lambda}$

In this section we will show a way to estimate the four entries of the matrix $\mathbf{\Lambda}$, introduced in Sections 2.5.2 and 3.5.2, using the MCMC samples of the unknown model parameters. Recall that $\underset{\sim}{\mathbf{s}}_n = (\text{lon}_n, \text{lat}_n)$ and $\underset{\sim}{\mathbf{d}}_n = d(\underset{\sim}{\mathbf{s}}_n) = (d_1(\underset{\sim}{\mathbf{s}}_n), d_2(\underset{\sim}{\mathbf{s}}_n)) = (D_{1,n}, D_{2,n})$ for all $n \in \{1, \ldots, N\}$. Since we imposed $\underset{\sim}{\mathbf{d}}_1 = \underset{\sim}{\mathbf{s}}_1$ and $\underset{\sim}{\mathbf{d}}_2 = \underset{\sim}{\mathbf{s}}_2$ to avoid unidentifiability issues, these two points will be ignored.

We will write the indices $j_1, \ldots, j_K$ to emphasize that $\Lambda_{1,1}^{(j_k)}$, $\Lambda_{1,2}^{(j_k)}$, $\Lambda_{2,1}^{(j_k)}$, and $\Lambda_{2,2}^{(j_k)}$ will be obtained through each MCMC sample $\mathbf{D}^{(j_k)} = \left[\begin{array}{c} D_{m,n}^{(j_k)} \end{array} \right]_{2 \times N} = \left[\begin{array}{cccc} \underset{\sim}{\mathbf{s}}_1 & \underset{\sim}{\mathbf{s}}_2 & \underset{\sim}{\mathbf{d}}_3^{(j_k)} & \cdots & \underset{\sim}{\mathbf{d}}_N^{(j_k)} \end{array} \right]$. For two distinct sites $\underset{\sim}{\mathbf{s}}_n$ and $\underset{\sim}{\mathbf{s}}_{n'}$ such that $n, n' \in \{3, \ldots, N\}$, $n \neq n'$, we have:

$$\underset{\sim}{\mathbf{d}}_n^{(j_k)} = \mathbf{\Lambda}^{(j_k)} \cdot \underset{\sim}{\mathbf{s}}_n \Longrightarrow \left\{ \begin{array}{rcl} D_{1,n}^{(j_k)} & = & \Lambda_{1,1}^{(j_k)} \cdot \text{lon}_n + \Lambda_{1,2}^{(j_k)} \cdot \text{lat}_n \\ D_{2,n}^{(j_k)} & = & \Lambda_{2,1}^{(j_k)} \cdot \text{lon}_n + \Lambda_{2,2}^{(j_k)} \cdot \text{lat}_n \end{array} \right. \tag{A.8}$$

and

$$\underset{\sim}{\mathbf{d}}_{n'}^{(j_k)} = \mathbf{\Lambda}^{(j_k)} \cdot \underset{\sim}{\mathbf{s}}_{n'} \Longrightarrow \left\{ \begin{array}{rcl} D_{1,n'}^{(j_k)} & = & \Lambda_{1,1}^{(j_k)} \cdot \text{lon}_{n'} + \Lambda_{1,2}^{(j_k)} \cdot \text{lat}_{n'} \\ D_{2,n'}^{(j_k)} & = & \Lambda_{2,1}^{(j_k)} \cdot \text{lon}_{n'} + \Lambda_{2,2}^{(j_k)} \cdot \text{lat}_{n'} \end{array} \right. . \tag{A.9}$$

Using the first lines of the systems given by Equations (A.8) and (A.9), obtain $\Lambda_{1,1}^{(j_k)}$ and $\Lambda_{1,2}^{(j_k)}$ by solving this new system using Cramer's rule (Lay et al., 2022, Sec. 3.3) as follows:

$$\left\{ \begin{array}{rcl} D_{1,n}^{(j_k)} & = & \text{lon}_n \cdot \Lambda_{1,1}^{(j_k)} + \text{lat}_n \cdot \Lambda_{1,2}^{(j_k)} \\ D_{1,n'}^{(j_k)} & = & \text{lon}_{n'} \cdot \Lambda_{1,1}^{(j_k)} + \text{lat}_{n'} \cdot \Lambda_{1,2}^{(j_k)} \end{array} \right. \Longrightarrow \left\{ \begin{array}{rcl} \Lambda_{1,1}^{(j_k)} & = & \dfrac{D_{1,n}^{(j_k)} \cdot \text{lat}_{n'} - D_{1,n'}^{(j_k)} \cdot \text{lat}_n}{\text{lon}_n \cdot \text{lat}_{n'} - \text{lon}_{n'} \cdot \text{lat}_n} \\ \Lambda_{1,2}^{(j_k)} & = & \dfrac{\text{lon}_n \cdot D_{1,n'}^{(j_k)} - \text{lon}_{n'} \cdot D_{1,n}^{(j_k)}}{\text{lon}_n \cdot \text{lat}_{n'} - \text{lon}_{n'} \cdot \text{lat}_n} \end{array} \right. .$$

Using the second lines of the systems given by Equations (A.8) and (A.9), obtain $\Lambda_{2,1}^{(j_k)}$ and $\Lambda_{2,2}^{(j_k)}$ by solving this new system again using Cramer's rule as follows:

$$\left\{ \begin{array}{rcl} D_{2,n}^{(j_k)} & = & \text{lon}_n \cdot \Lambda_{2,1}^{(j_k)} + \text{lat}_n \cdot \Lambda_{2,2}^{(j_k)} \\ D_{2,n'}^{(j_k)} & = & \text{lon}_{n'} \cdot \Lambda_{2,1}^{(j_k)} + \text{lat}_{n'} \cdot \Lambda_{2,2}^{(j_k)} \end{array} \right. \Longrightarrow \left\{ \begin{array}{rcl} \Lambda_{2,1}^{(j_k)} & = & \dfrac{D_{2,n}^{(j_k)} \cdot \text{lat}_{n'} - D_{2,n'}^{(j_k)} \cdot \text{lat}_n}{\text{lon}_n \cdot \text{lat}_{n'} - \text{lon}_{n'} \cdot \text{lat}_n} \\ \Lambda_{2,2}^{(j_k)} & = & \dfrac{\text{lon}_n \cdot D_{2,n'}^{(j_k)} - \text{lon}_{n'} \cdot D_{2,n}^{(j_k)}}{\text{lon}_n \cdot \text{lat}_{n'} - \text{lon}_{n'} \cdot \text{lat}_n} \end{array} \right. .$$

These four solutions are valid for the pairs $(n, n') \in \{3, \ldots, N\}^2$, $n < n'$, under the restriction $\text{lon}_n \cdot \text{lat}_{n'} \neq \text{lon}_{n'} \cdot \text{lat}_n$. In Sections 2.5.2 and 3.5.2 we worked with $N = 16$ gauged sites and $K = 1000$ MCMC samples of the parameters. Among the fourteen non-anchoring points plotted in Figures 2.10 and 3.8, we discard only the subset of pairs $\{(5, 10), (7, 15), (8, 13)\}$ due to the constraint we imposed. Thus, we can compute summary statistics for the quantities $\Lambda_{1,1}$, $\Lambda_{1,2}$, $\Lambda_{2,1}$, and $\Lambda_{2,2}$ from the following MCMC samples:

$$\left\{ \Lambda_{1,1}^{(j_k)}(n, n'), \Lambda_{1,2}^{(j_k)}(n, n'), \Lambda_{2,1}^{(j_k)}(n, n'), \Lambda_{2,2}^{(j_k)}(n, n') : (n, n') \in \mathcal{N}, k \in \mathcal{K} \right\},$$

where $\mathcal{N} = \{(n, n') \in \mathbb{N}^2 : n, n' \in \{3, \ldots, N\}, n < n', \text{lon}_n \cdot \text{lat}_{n'} \neq \text{lon}_{n'} \cdot \text{lat}_n\}$ and $\mathcal{K} = \{1, \ldots, K\}$.

A.5 Obtaining the borders of the estimated deformed map

Let $\mathbf{S}$ be the $2 \times N$ matrix with the points of the $\mathcal{S}$-space used to fit the model given by Equation (2.3). Let $\mathbf{S}^{\mathsf{map}}$ be the $2 \times N^{\mathsf{map}}$ matrix with the borders of the $\mathcal{S}$-space. Applying Equation (2.14), we have the following expression:

$$\mathbf{D}^{\mathsf{map}} \mid \mathbf{D} \sim \mathsf{N}_{2\times N^{\mathsf{map}}}(\mathbf{S}^{\mathsf{map}} + (\mathbf{D} - \mathbf{S})\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^{\mathsf{map}} - \mathbf{R}_{\mathsf{u},\mathsf{g}}\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}),$$

where

$$\left[\exp\{-\psi\|\underset{\sim}{\mathbf{s}}_n - \underset{\sim}{\mathbf{s}}_{n'}\|^2\} \right]_{(N+N^{\mathsf{map}})\times(N+N^{\mathsf{map}})} = \begin{bmatrix} \mathbf{R}_d & \mathbf{R}_{\mathsf{g},\mathsf{u}} \\ \mathbf{R}_{\mathsf{u},\mathsf{g}} & \mathbf{R}_d^{\mathsf{map}} \end{bmatrix}.$$

Using K MCMC samples, where $\boldsymbol{\theta}^{(j_k)} = \left\{\boldsymbol{\beta}_0^{(j_k)}, \boldsymbol{\beta}_1^{(j_k)}, \ldots, \boldsymbol{\beta}_T^{(j_k)}, V^{(j_k)}, \phi^{(j_k)}, \mathbf{D}^{(j_k)}, \boldsymbol{\Sigma}^{(j_k)}\right\}$ is the k^{th} MCMC sample, for $k \in \{1, \ldots, K\}$ sample $\mathbf{D}_{(j_k)}^{\mathsf{map}}$ from the distribution $\mathsf{N}_{2\times N^{\mathsf{map}}}(\mathbf{S}^{\mathsf{map}} + (\mathbf{D}^{(j_k)} - \mathbf{S})\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d^{\mathsf{map}} - \mathbf{R}_{\mathsf{u},\mathsf{g}}\mathbf{R}_d^{-1}\mathbf{R}_{\mathsf{g},\mathsf{u}})$. Finally, compute the estimated deformed map by doing $\bar{\mathbf{D}}^{\mathsf{map}} = \frac{1}{K}\sum_{k=1}^{K} \mathbf{D}_{(j_k)}^{\mathsf{map}}$ and use it as the corresponding estimated borders of the $\mathcal{D}$-space to plot.

A.6 Vector representation of the model given in Equation (2.3)

A.6.1 Formulation

Let $\text{vec}(\cdot)$ be the *vectorization operator*, where $\text{vec}\,\mathbf{A}$ creates a $p \cdot q$ column vector from a $p \times q$ matrix $\mathbf{A}$ by stacking its q columns. The Kronecker product between two matrices is denoted by $\otimes$. Let $\underset{\sim}{\mathbf{Y}}_t = \text{vec}\,\mathbf{Y}_t$ for all $t \in \{1, \ldots, T\}$. Based on Quintana (1987, Sec. 3.3.2), the vector representation of the model given in Equation (2.3) is given as follows:

$$\begin{aligned}
\underset{\sim}{\mathbf{Y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma} &\sim \mathsf{N}_{Nq}([\mathbf{I}_q \otimes \mathbf{X}_t]\,\text{vec}\,\boldsymbol{\beta}_t, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{B}]),\ t \in \{1, \ldots, T\}, \\
\mathbf{B} &= \left[B_{n,n'} \right]_{N\times N}, \\
B_{n,n'} &= \begin{cases} \exp\{-\phi\|d(\underset{\sim}{\mathbf{s}}_n) - d(\underset{\sim}{\mathbf{s}}_{n'})\|\}, & \text{if } n \neq n' \\ 1, & \text{if } n = n' \end{cases}, \\
\text{vec}\,\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{pq}([\mathbf{I}_q \otimes \mathbf{G}_t]\,\text{vec}\,\boldsymbol{\beta}_{t-1}, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{W}]),\ t \in \{1, \ldots, T\}, \\
\text{vec}\,\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma} &\sim \mathsf{N}_{pq}(\text{vec}\,\mathbf{M}_0, V \cdot [\boldsymbol{\Sigma} \otimes \mathbf{C}_0]), \\
V &\sim \mathsf{IG}(a_V, b_V), \\
\boldsymbol{\Sigma} &\sim \mathsf{IW}_q(a_{\boldsymbol{\Sigma}}, \mathbf{b}_{\boldsymbol{\Sigma}}), \\
\phi &\sim \mathsf{G}(a_\phi, b_\phi), \\
\mathbf{D} &\sim \mathsf{N}_{2\times N}(\mathbf{S}, \boldsymbol{\sigma}_d^2, \mathbf{R}_d).
\end{aligned} \tag{A.10}$$

Assuming prior independence for the parameters ϕ, $\mathbf{D}$, V and $\boldsymbol{\Sigma}$, by the law of total probability and also by the Markovian property, we write the following joint prior density:

$$f(\boldsymbol{\theta}) = f(\phi)f(\mathbf{D})f(V)f(\boldsymbol{\Sigma})f(\operatorname{vec}\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \prod_{t=1}^{T} f(\operatorname{vec}\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}). \tag{A.11}$$

Define $\underset{\sim}{\mathbf{Y}} = \{\underset{\sim}{\mathbf{Y}}_1, \ldots, \underset{\sim}{\mathbf{Y}}_T\}$ the set of multivariate responses, assuming that $\underset{\sim}{\mathbf{Y}}_1, \ldots, \underset{\sim}{\mathbf{Y}}_T$ are T conditionally independent observations given $\boldsymbol{\theta}$. Thus, the likelihood function of $\boldsymbol{\theta}$ is:

$$l(\boldsymbol{\theta}; \underset{\sim}{\mathbf{y}}) = f(\underset{\sim}{\mathbf{y}} \mid \boldsymbol{\theta}) = \prod_{t=1}^{T} f(\underset{\sim}{\mathbf{y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}). \tag{A.12}$$

From Equations (A.11) and (A.12), by the Bayes' theorem we have the following posterior density:

$$\begin{aligned} f(\boldsymbol{\theta} \mid \underset{\sim}{\mathbf{y}}) \quad &\propto \quad f(\phi)f(\mathbf{D})f(V)f(\boldsymbol{\Sigma}) \\ &\times \quad f(\operatorname{vec}\boldsymbol{\beta}_0 \mid V, \boldsymbol{\Sigma}) \left[\prod_{t=1}^{T} f(\operatorname{vec}\boldsymbol{\beta}_t \mid \boldsymbol{\beta}_{t-1}, V, \boldsymbol{\Sigma}) f(\underset{\sim}{\mathbf{y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \right]. \end{aligned} \tag{A.13}$$

A.6.2 Bayesian inference via Markov chain Monte Carlo

The posterior density given in Equation (A.13) does not have a closed-form. For this reason, we resort the Markov chain Monte Carlo (MCMC) method to obtain samples from the model parameters. Next, we will obtain the full conditional distributions of the parameters in order to implement a hybrid algorithm. As the calculations are very similar to those already done in Section 2.2.3, we will focus on presenting final results and algorithms.

A.6.2.1 Sampling from the full conditional distribution of V

The full conditional distribution of V is given by

$$V \mid \underset{\sim}{\mathbf{Y}} = \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, \phi, \mathbf{D}, \boldsymbol{\Sigma} \sim \mathsf{IG}(a'_V, b'_V), \tag{A.14}$$

where

$$a'_V = a_V + \frac{pq}{2} + \frac{Tpq}{2} + \frac{TNq}{2}$$

and

$$\begin{aligned} b'_V \quad = \quad b_V \quad &+ \quad \frac{1}{2}(\operatorname{vec}\boldsymbol{\beta}_0 - \operatorname{vec}\mathbf{M}_0)^{\intercal}[\boldsymbol{\Sigma} \otimes \mathbf{C}_0]^{-1}(\operatorname{vec}\boldsymbol{\beta}_0 - \operatorname{vec}\mathbf{M}_0) \\ &+ \quad \frac{1}{2}\sum_{t=1}^{T}(\operatorname{vec}\boldsymbol{\beta}_t - [\mathbf{I}_q \otimes \mathbf{G}_t]\operatorname{vec}\boldsymbol{\beta}_{t-1})^{\intercal}[\boldsymbol{\Sigma} \otimes \mathbf{W}]^{-1}(\operatorname{vec}\boldsymbol{\beta}_t - [\mathbf{I}_q \otimes \mathbf{G}_t]\operatorname{vec}\boldsymbol{\beta}_{t-1}) \\ &+ \quad \frac{1}{2}\sum_{t=1}^{T}(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t)^{\intercal}[\boldsymbol{\Sigma} \otimes \mathbf{B}]^{-1}(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\operatorname{vec}\boldsymbol{\beta}_t). \end{aligned}$$

In Appendix B.2.2.1, we present a program written in Python to sample from the full conditional distribution of V given in Equation (A.14).

A.6.2.2 Sampling from the full conditional distribution of $\boldsymbol{\Sigma}$

Based on Gupta and Nagar (2000, Theorem 1.2.22 (iii)), the full conditional distribution of $\boldsymbol{\Sigma}$ is given by

$$\boldsymbol{\Sigma} \mid \underset{\sim}{\mathbf{Y}} = \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}_0, \boldsymbol{\beta}, V, \phi, \mathbf{D} \sim \mathsf{IW}_q(a'_{\boldsymbol{\Sigma}}, \mathbf{b}'_{\boldsymbol{\Sigma}}), \tag{A.15}$$

where

$$a'_{\boldsymbol{\Sigma}} = a_{\boldsymbol{\Sigma}} + p + Tp + TN$$

and

$$\begin{aligned} \mathbf{b}'_{\boldsymbol{\Sigma}} &= \mathbf{b}_{\boldsymbol{\Sigma}} + (\boldsymbol{\beta}_0 - \mathbf{M}_0)^\intercal (V \cdot \mathbf{C}_0)^{-1} (\boldsymbol{\beta}_0 - \mathbf{M}_0) \\ &\quad + \sum_{t=1}^{T} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1})^\intercal (V \cdot \mathbf{W})^{-1} (\boldsymbol{\beta}_t - \mathbf{G}_t \boldsymbol{\beta}_{t-1}) \\ &\quad + \sum_{t=1}^{T} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t)^\intercal (V \cdot \mathbf{B})^{-1} (\mathbf{y}_t - \mathbf{X}_t \boldsymbol{\beta}_t) \end{aligned}$$

are the same two quantities obtained in Section 2.2.3.2 and

$$\begin{aligned} \mathbf{M}_0 &= \{[\operatorname{vec} \mathbf{I}_q]^\intercal \otimes \mathbf{I}_p\} \cdot [\mathbf{I}_q \otimes \operatorname{vec} \mathbf{M}_0], \\ \boldsymbol{\beta}_t &= \{[\operatorname{vec} \mathbf{I}_q]^\intercal \otimes \mathbf{I}_p\} \cdot [\mathbf{I}_q \otimes \operatorname{vec} \boldsymbol{\beta}_t], t \in \{0, 1, \ldots, T\}, \\ \mathbf{y}_t &= \{[\operatorname{vec} \mathbf{I}_q]^\intercal \otimes \mathbf{I}_N\} \cdot \left[\mathbf{I}_q \otimes \underset{\sim}{\mathbf{y}}_t\right], t \in \{1, \ldots, T\}. \end{aligned}$$

In Appendix B.2.2.2, we present a program written in Python to sample from the full conditional distribution of $\boldsymbol{\Sigma}$ given in Equation (A.15).

A.6.2.3 Sampling from the full conditional distribution of $\operatorname{vec} \boldsymbol{\beta}_t, \operatorname{vec} \boldsymbol{\beta}_1, \ldots, \operatorname{vec} \boldsymbol{\beta}_T$

We use the Forward-Filtering Backward-Sampling (FFBS) algorithm (Frühwirth-Schnatter, 1994; Carter and Kohn, 1994; Chib and Greenberg, 1995) to sample from the posterior distribution of the parameters $\operatorname{vec} \boldsymbol{\beta}_0, \operatorname{vec} \boldsymbol{\beta}_1, \ldots, \operatorname{vec} \boldsymbol{\beta}_T$. This method is described in Algorithm 15 and its implementation is shown in Appendix B.2.2.3.

A.6.2.4 Sampling from the full conditional distribution of ϕ

For all $\phi > 0$, the natural logarithm of the full conditional density of ϕ is given by:

$$\begin{aligned} \ln f(\phi \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \mathbf{D}, \boldsymbol{\Sigma}) &= \text{constant} + \ln f(\phi) + \sum_{t=1}^{T} \ln f(\underset{\sim}{\mathbf{y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\ &\propto \{(a_\phi - 1) \cdot \ln \phi - b_\phi \cdot \phi\} - \frac{Tq}{2} \ln \det \mathbf{B} \\ &\quad - \frac{1}{2V} \sum_{t=1}^{T} (\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t)^\intercal [\boldsymbol{\Sigma} \otimes \mathbf{B}]^{-1} (\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t). \end{aligned} \tag{A.16}$$

Algorithm 15 FFBS algorithm to sample from the full conditional distribution of the parameters $\text{vec}\,\boldsymbol{\beta}_0, \text{vec}\,\boldsymbol{\beta}_1, \ldots, \text{vec}\,\boldsymbol{\beta}_T$.

Start with $\text{vec}\,\mathbf{M}_0$, $\mathbf{C}_0$ and $\boldsymbol{\theta}^{(j-1)} = \left\{\boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right\}$.

1: Compute $\tilde{\mathbf{C}}_0 = V^{(j-1)} \cdot [\boldsymbol{\Sigma}^{(j-1)} \otimes \mathbf{C}_0]$ and the matrix $\mathbf{B}_{(j-1)} = \left[\ B_{n,n'}^{(j-1)}\ \right]_{N\times N}$, where:

$$B_{n,n'}^{(j-1)} = \exp\{-\phi^{(j-1)}\|d^{(j-1)}(\underset{\sim}{\mathbf{s}}_n) - d^{(j-1)}(\underset{\sim}{\mathbf{s}}_{n'})\|\}, \quad n, n' \in \{1, \ldots, N\}.$$

2: **for** $t \leftarrow 1$ to T **do**

$$\begin{aligned}
\mathbf{E}_t &= V^{(j-1)} \cdot [\boldsymbol{\Sigma}^{(j-1)} \otimes \mathbf{W}] + [\mathbf{I}_q \otimes \mathbf{G}_t]\tilde{\mathbf{C}}_{t-1}[\mathbf{I}_q \otimes \mathbf{G}_t]^\intercal, \\
\mathbf{Q}_t &= V^{(j-1)} \cdot [\boldsymbol{\Sigma}^{(j-1)} \otimes \mathbf{B}_{(j-1)}] + [\mathbf{I}_q \otimes \mathbf{X}_t]\mathbf{E}_t[\mathbf{I}_q \otimes \mathbf{X}_t]^\intercal, \\
\text{vec}\,\mathbf{a}_t &= [\mathbf{I}_q \otimes \mathbf{G}_t]\,\text{vec}\,\mathbf{M}_{t-1}, \\
\text{vec}\,\mathbf{M}_t &= \text{vec}\,\mathbf{a}_t + \mathbf{E}_t[\mathbf{I}_q \otimes \mathbf{X}_t]^\intercal \mathbf{Q}_t^{-1}(\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t]\,\text{vec}\,\mathbf{a}_t), \\
\tilde{\mathbf{C}}_t &= \mathbf{E}_t - \mathbf{E}_t[\mathbf{I}_q \otimes \mathbf{X}_t]^\intercal \mathbf{Q}_t^{-1}[\mathbf{I}_q \otimes \mathbf{X}_t]\mathbf{E}_t.
\end{aligned}$$

3: **end for**

4: Sample $\text{vec}\,\boldsymbol{\beta}_T^{(j)}$ from the distribution $\mathsf{N}_{pq}(\text{vec}\,\mathbf{M}_T, \tilde{\mathbf{C}}_T)$.

5: **for** $t \leftarrow T-1$ to 0 **do**

6: Sample $\text{vec}\,\boldsymbol{\beta}_t^{(j)}$ from the distribution $\mathsf{N}_{pq}(\mathbf{H}_t\underset{\sim}{\mathbf{h}}_t, \mathbf{H}_t)$, where:

$$\begin{aligned}
\mathbf{H}_t &= \left\{\tilde{\mathbf{C}}_t^{-1} + [\mathbf{I}_q \otimes \mathbf{G}_t]^\intercal \left(V^{(j-1)} \cdot [\boldsymbol{\Sigma}^{(j-1)} \otimes \mathbf{W}]\right)^{-1} [\mathbf{I}_q \otimes \mathbf{G}_t]\right\}^{-1}, \\
\underset{\sim}{\mathbf{h}}_t &= \tilde{\mathbf{C}}_t^{-1}\,\text{vec}\,\mathbf{M}_t + [\mathbf{I}_q \otimes \mathbf{G}_{t+1}]^\intercal \left\{V^{(j-1)} \cdot [\boldsymbol{\Sigma}^{(j-1)} \otimes \mathbf{W}]\right\}^{-1} \text{vec}\,\boldsymbol{\beta}_{t+1}^{(j)}.
\end{aligned}$$

7: **end for**

Since $f(\phi \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \mathbf{D}, \boldsymbol{\Sigma})$ does not have a known form, we sample from this density through the Metropolis-Hastings algorithm. The method is described in Algorithm 16 and its implementation is presented in Appendix B.2.2.4.

Algorithm 16 Metropolis-Hastings algorithm to sample from $f(\phi \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \mathbf{D}, \boldsymbol{\Sigma})$.

Start with a_ϕ, b_ϕ and $\boldsymbol{\theta}^{(j-1)} = \left\{\boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right\}$, where $j \in \mathbb{N}$ and $\boldsymbol{\beta}^{(j-1)} = \left\{\boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}\right\}$.

1: Generate a random number $u \in (0,1)$ and propose a new value for ϕ by sampling from the inverse normal distribution given by $\mathsf{IN}(\phi^{(j-1)}, \delta)$, where δ is a tuning parameter and the proposal density is:

$$g(a \mid b, \delta) = \sqrt{\frac{\delta}{2\pi a^3}} \exp\left\{-\frac{\delta(a-b)^2}{2b^2 a}\right\} \mathbb{1}_{(0,+\infty)}(a).$$

Let ϕ_{prop} be the proposed value for the j^{th} MCMC iteration.

2: **if** $u \leq \min\left\{1, \dfrac{f(\phi_{\mathsf{prop}} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}) g(\phi^{(j-1)} \mid \phi_{\mathsf{prop}}, \delta)}{f(\phi^{(j-1)} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}) g(\phi_{\mathsf{prop}} \mid \phi^{(j-1)}, \delta)}\right\}$ **then**

3: $\quad \phi^{(j)} \leftarrow \phi_{\mathsf{prop}}$.

4: **else**

5: $\quad \phi^{(j)} \leftarrow \phi^{(j-1)}$.

6: **end if**

A.6.2.5 Sampling from the full conditional distribution of D

The natural logarithm of the full conditional density of $\mathbf{D}$ is given by:

$$\begin{aligned}
\ln f(\mathbf{D} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma}) &= \text{constant} + \ln f(\mathbf{D}) + \sum_{t=1}^{T} \ln f(\underset{\sim}{\mathbf{y}}_t \mid \boldsymbol{\beta}_t, V, \phi, \mathbf{D}, \boldsymbol{\Sigma}) \\
&\propto \left\{-\frac{1}{2}\operatorname{tr}\left[(\mathbf{D}-\mathbf{S})^\intercal \boldsymbol{\sigma}_d^{-2} (\mathbf{D}-\mathbf{S}) \mathbf{R}_d^{-1}\right]\right\} - \frac{Tq}{2} \ln\det \mathbf{B} \\
&- \frac{1}{2V} \sum_{t=1}^{T} (\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t)^\intercal [\boldsymbol{\Sigma} \otimes \mathbf{B}]^{-1} (\underset{\sim}{\mathbf{y}}_t - [\mathbf{I}_q \otimes \mathbf{X}_t] \operatorname{vec} \boldsymbol{\beta}_t).
\end{aligned} \tag{A.17}$$

Since $f(\mathbf{D} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$ does not have a known form, we sample from this density through the slice sampler. The method is described in Algorithm 17-18 and its implementation is presented in Appendix B.2.2.5.

Algorithm 17 Slice sampling algorithm to sample from $f(\mathbf{D} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$ – Part 1.

Start with $\sigma^2_{d_{1,1}}$, $\sigma^2_{d_{2,2}}$, ψ and $\boldsymbol{\theta}^{(j-1)} = \left\{\boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right\}$, where $j \in \mathbb{N}$ and $\boldsymbol{\beta}^{(j-1)} = \left\{\boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}\right\}$.
Update $\mathbf{D}^{(j-1)} = \left[\begin{array}{c} D^{(j-1)}_{m,n} \end{array}\right]_{2\times N}$ by $\mathbf{D}^{(j)} = \left[\begin{array}{c} D^{(j)}_{m,n} \end{array}\right]_{2\times N}$ proceeding as follows:

- Do $D^{(j)}_{1,1} \leftarrow D^{(j-1)}_{1,1}$ and $D^{(j)}_{1,2} \leftarrow D^{(j-1)}_{1,2}$.
- Obtain $D^{(j)}_{1,3}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\left[\begin{array}{ccccc} D^{(j)}_{1,1} & D^{(j)}_{1,2} & D^{(j-1)}_{1,3} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & D^{(j-1)}_{2,3} & \cdots & D^{(j-1)}_{2,N} \end{array}\right] \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{1,3} : f\left(\left[\begin{array}{ccccc} D^{(j)}_{1,1} & D^{(j)}_{1,2} & D_{1,3} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & D^{(j-1)}_{2,3} & \cdots & D^{(j-1)}_{2,N} \end{array}\right] \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D^{(j-1)}_{1,3}$ that contains all, or much, of the slice.
  3. Draw the new point, $D^{(j)}_{1,3}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.
- $\vdots$
- Obtain $D^{(j)}_{1,N}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\left[\begin{array}{ccccc} D^{(j)}_{1,1} & D^{(j)}_{1,2} & D^{(j)}_{1,3} & \cdots & D^{(j-1)}_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & D^{(j-1)}_{2,3} & \cdots & D^{(j-1)}_{2,N} \end{array}\right] \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{1,N} : f\left(\left[\begin{array}{ccccc} D^{(j)}_{1,1} & D^{(j)}_{1,2} & D^{(j)}_{1,3} & \cdots & D_{1,N} \\ D^{(j-1)}_{2,1} & D^{(j-1)}_{2,2} & D^{(j-1)}_{2,3} & \cdots & D^{(j-1)}_{2,N} \end{array}\right] \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D^{(j-1)}_{1,N}$ that contains all, or much, of the slice.
  3. Draw the new point, $D^{(j)}_{1,N}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.

Algorithm 18 Slice sampling algorithm to sample from $f(\mathbf{D} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}, V, \phi, \boldsymbol{\Sigma})$ – Part 2.

- Do $D_{2,1}^{(j)} \leftarrow D_{2,1}^{(j-1)}$ and $D_{2,2}^{(j)} \leftarrow D_{2,2}^{(j-1)}$.
- Obtain $D_{2,3}^{(j)}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j)} & \cdots & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & D_{2,3}^{(j-1)} & \cdots & D_{2,N}^{(j-1)} \end{bmatrix} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{2,3} : f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & D_{1,3}^{(j)} & \cdots & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & D_{2,3} & \cdots & D_{2,N}^{(j-1)} \end{bmatrix} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D_{2,3}^{(j-1)}$ that contains all, or much, of the slice.
  3. Draw the new point, $D_{2,3}^{(j)}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.
- $\vdots$
- Obtain $D_{2,N}^{(j)}$ by doing the following:
  1. Draw u uniformly from the interval
  $$\left(0, f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & \cdots & D_{1,N-1}^{(j)} & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & \cdots & D_{1,N-1}^{(j)} & D_{2,N}^{(j-1)} \end{bmatrix} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right)\right),$$
  thus defining a horizontal "slice"
  $$\mathcal{H} = \left\{D_{2,N} : f\left(\begin{bmatrix} D_{1,1}^{(j)} & D_{1,2}^{(j)} & \cdots & D_{1,N-1}^{(j)} & D_{1,N}^{(j)} \\ D_{2,1}^{(j)} & D_{2,2}^{(j)} & \cdots & D_{2,N-1}^{(j)} & D_{2,N} \end{bmatrix} \mid \underset{\sim}{\mathbf{y}}, \boldsymbol{\beta}^{(j-1)}, V^{(j-1)}, \phi^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}\right) \geq u\right\}.$$
  2. Find an interval, $\mathcal{I} = (I_{\mathrm{L}}, I_{\mathrm{R}})$, around $D_{2,N}^{(j-1)}$ that contains all, or much, of the slice.
  3. Draw the new point, $D_{2,N}^{(j)}$, uniformly from the interval $\mathcal{H} \cap \mathcal{I}$.

A.6.2.6 Hybrid MCMC algorithm

We propose the use of a hybrid MCMC algorithm to generate samples from the posterior distribution of the unknown model parameters $\boldsymbol{\theta} = \{\boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T, V, \boldsymbol{\Sigma}, \phi, \mathbf{D}\}$, adding FFBS, slice sampler and Metropolis-Hastings steps in the Gibbs sampler. Algorithm 19 describes how to iterate from $\boldsymbol{\theta}^{(j-1)}$ to $\boldsymbol{\theta}^{(j)}$.

Algorithm 19 Hybrid MCMC algorithm to sample from $f(V, \boldsymbol{\Sigma}, \boldsymbol{\beta}_0, \boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_T, \phi, \mathbf{D} \mid \underset{\sim}{\mathbf{y}})$.

Start with $\boldsymbol{\theta}^{(j-1)} = \{V^{(j-1)}, \boldsymbol{\Sigma}^{(j-1)}, \boldsymbol{\beta}_0^{(j-1)}, \boldsymbol{\beta}_1^{(j-1)}, \ldots, \boldsymbol{\beta}_T^{(j-1)}, \phi^{(j-1)}, \mathbf{D}^{(j-1)}\} \in \boldsymbol{\Theta}$.

1: **for** $t \leftarrow 0$ to T **do**

2: Compute $\operatorname{vec} \boldsymbol{\beta}_t^{(j-1)}$ to write the $p \times q$ matrix $\boldsymbol{\beta}_t^{(j-1)}$ as a vector.

3: **end for**

4: Update the parameters:

a: Generate $V^{(j)}$ and $\boldsymbol{\Sigma}^{(j)}$ from their known full conditional distributions given in Equations (A.14) and (A.15), respectively.

b: Run Algorithm 15 to sample $\operatorname{vec} \boldsymbol{\beta}_0^{(j)}, \operatorname{vec} \boldsymbol{\beta}_1^{(j)}, \ldots, \operatorname{vec} \boldsymbol{\beta}_T^{(j)}$ through the FFBS algorithm.

c: Run Algorithm 16 to sample $\phi^{(j)}$ through the Metropolis-Hastings algorithm.

d: Run Algorithms 17-18 to sample $\mathbf{D}^{(j)}$ through the slice sampler.

5: **for** $t \leftarrow 0$ to T **do**

6: Compute $\boldsymbol{\beta}_t^{(j)} = \{[\operatorname{vec} \mathbf{I}_q]^\intercal \otimes \mathbf{I}_p\} \cdot \left[\mathbf{I}_q \otimes \operatorname{vec} \boldsymbol{\beta}_t^{(j)}\right]$ to write the vector $\operatorname{vec} \boldsymbol{\beta}_t^{(j)}$ as a $p \times q$ matrix.

7: **end for**

Appendix B

Python implementation

In this appendix, we provide some codes written in Python 3.8 (van Rossum and Drake, 2011) with the aim of contributing to a more reproducible science. These codes are also available in the following repository: https://github.com/rbulhoes/dsc-thesis

Using the `Numba` library (Lam et al., 2015), we accelerate several parts of the code and also some `NumPy` functions (Harris et al., 2020) to perform in Python at a similar speed to other faster languages (e.g. C and Fortran). Some `SciPy` functions (Virtanen et al., 2020) were used in the implementation.

B.1 Implementation of the pseudocodes of Chapter 2

B.1.1 Libraries and auxiliary functions

```
############################ Python libraries
import numpy as np
from scipy.stats import uniform, matrix_normal
from timeit import default_timer as timer
from math import pi, exp, log, sqrt, isfinite, inf, dist
from numba import njit, float64, boolean, uint8, int64, prange
from numba.core.errors import NumbaPerformanceWarning
from warnings import simplefilter
from scipy.spatial import distance, distance_matrix
simplefilter('ignore', category = NumbaPerformanceWarning)

############################ Auxiliary functions
# Obtaining the matrices B or R
```

```
# Obtaining the matrices B or R
@njit(parallel = True)
def nb_BR(matrix: float64[:,:], need_transp: boolean, scalar: float64,
          squared = False) -> float64[:,:]:
  if need_transp is True:
    X = matrix.T
  else:
    X = matrix
  n, m = X.shape
  M = np.ones((n, n))
  for i in prange(0, n - 1):
    for j in prange(i + 1, n):
      dist_ij = 0.0
      for k in prange(0, m):
        dist_ij += pow(X[i, k] - X[j, k], 2)
      if squared is False:
        M[i, j] = exp(-scalar * sqrt(dist_ij)) # This is for B
      else:
        M[i, j] = exp(-scalar * dist_ij) # This is for R
      M[j, i] = M[i, j]

  return M

# Kronecker product
@njit
def nb_kp(X: float64[:,:], Y: float64[:,:]) -> float64[:,:]:
  return np.kron(X, Y)

# Sampling from matrix-variate normal distribution (slow, but more robust)
def rMVN(n: np.uint8, avg: np.array, left: np.array,
         right: np.array, n_seed: np.int64) -> np.array:
  p, q = avg.shape
  vec_avg = np.matrix.flatten(avg, 'F')
  kron_cov = nb_kp(right, left)

  np.random.seed(seed = n_seed)
  vec_sample = np.random.multivariate_normal(mean = vec_avg,
                                             cov = kron_cov,
                                             size = n,
```

```
                                                check_valid = 'ignore')

  return np.reshape(vec_sample, (p, q), 'F')

# Sampling from matrix-variate normal distribution (fast, but less robust)
@njit
def chol_rMVN(n_seed: int64, avg: float64[:,:],
              left: float64[:,:], right: float64[:,:]) -> float64[:,:]:
  p, q = avg.shape
  vec_avg = (avg.T).flatten()
  kron_cov = np.kron(right, left)
  np.random.seed(n_seed)
  gen = np.linalg.cholesky(kron_cov) @ np.random.standard_normal(vec_avg.size)
  vec_sample = vec_avg + gen
  return vec_sample.reshape((q, p)).T

# Sampling from the inverse Wishart distribution
@njit
def chol_rIW(nu: float64, M: float64[:,:], n_seed: uint8) -> float64[:,:]:
  np.random.seed(n_seed)
  dim = M.shape[0]
  inv_M = np.linalg.inv(M)
  chol = np.linalg.cholesky(inv_M)
  df = nu + dim - 1
  foo = np.zeros((dim, dim))
  for i in range(dim):
    for j in range(i + 1):
      if i == j:
        foo[i, j] = sqrt(np.random.chisquare(df - (i + 1) + 1))
      else:
        foo[i, j] = np.random.standard_normal()
  return np.linalg.inv(chol @ (foo @ (foo.T @ chol.T)))

# Sampling from the Inverse-Normal Distribution
@njit
def rIN(n_seed: int64, mu: float64, sigma2: float64) -> float64:
  np.random.seed(n_seed)
  return np.random.wald(mu, sigma2)
```

```
# Sampling from the uniform distribution and applying the natural logarithm
@njit
def log_rUnif(n_seed: int64) -> float64:
  np.random.seed(n_seed)
  return log(np.random.rand())

# Logarithm of the PDF of the Inverse-Normal distribution
@njit
def logpdf_IN(x: float64, mean: float64, sigma2: float64) -> float64:
  term1 = 0.5*(log(sigma2) - log(2*pi*pow(x, 3)))
  term2 = -0.5*sigma2*pow(x - mean, 2) / (pow(mean, 2) * x)
  return term1 + term2
```

B.1.2 Full conditional distributions

B.1.2.1 Sampling from the full conditional distribution of V

```
@njit(parallel = True)
def updt_V(it: int64, M_Y: float64[:,:,:], M_X: float64[:,:,:],
           M_G: float64[:,:,:], W_usage: float64[:,:], init_a_V: float64,
           init_b_V: float64, init_M0: float64[:,:], init_C0: float64[:,:],
           D_usage: float64[:,:], phi_usage: float64,
           Beta_usage: float64[:,:,:], Sigma_usage: float64[:,:]) -> float64:

  T_obs = M_Y.shape[0] - 1
  N_obs = M_Y.shape[1]
  q_obs = M_Y.shape[2]
  p_obs = M_X.shape[2]

  B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)
  inv_B_usage = np.linalg.inv(B_usage)
  inv_init_C0 = np.linalg.inv(init_C0)
  inv_W_usage = np.linalg.inv(W_usage)
  inv_Sigma_usage = np.linalg.inv(Sigma_usage)

  df = init_a_V + 0.5*T_obs*N_obs*q_obs + 0.5*p_obs*q_obs*(T_obs + 1)
  par, aux = np.zeros((q_obs, q_obs)), np.zeros((q_obs, q_obs))
  for t in prange(1, T_obs + 1):
    dif_t = M_Y[t] - (M_X[t] @ Beta_usage[t])
```

```
    par += (dif_t.T @ inv_B_usage) @ (dif_t @ inv_Sigma_usage)
    sub_t = Beta_usage[t] - (M_G[t] @ Beta_usage[t - 1])
    aux += (sub_t.T @ inv_W_usage) @ (sub_t @ inv_Sigma_usage)
  sub_0 = Beta_usage[0] - init_M0
  aux_0 = (sub_0.T @ inv_init_C0) @ (sub_0 @ inv_Sigma_usage)

  quantity = init_b_V + 0.5*np.trace(aux_0 + par + aux)

  np.random.seed(it)
  inv_sample = np.random.gamma(shape = df, scale = 1/quantity)

  return 1/inv_sample
```

B.1.2.2 Sampling from the full conditional distribution of Σ

```
@njit(parallel = True)
def updt_Sigma(it: int64, Beta_usage: float64[:,:,:], M_Y: float64[:,:,:],
               M_G: float64[:,:,:], M_X: float64[:,:,:], W_usage: float64[:,:],
               init_M0: float64[:,:], init_C0: float64[:,:], V_usage: float64,
               init_a_Sigma: float64, init_b_Sigma: float64[:,:],
               D_usage: float64[:,:], phi_usage: float64) -> float64[:,:]:

  T_obs = M_Y.shape[0] - 1
  N_obs = M_Y.shape[1]
  q_obs = M_Y.shape[2]
  p_obs = M_X.shape[2]

  B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)

  inv_V_usage = 1 / V_usage
  inv_W_usage = np.linalg.inv(W_usage)
  inv_VW_usage = inv_V_usage * inv_W_usage
  inv_B_usage = np.linalg.inv(B_usage)
  inv_VB_usage = inv_V_usage * inv_B_usage
  inv_init_C0 = np.linalg.inv(init_C0)
  dif_0 = Beta_usage[0] - init_M0
  par_init = init_b_Sigma + ((dif_0.T @ (inv_V_usage * inv_init_C0)) @ dif_0)
  par_Beta = np.zeros((q_obs, q_obs))
  par_Y = np.zeros((q_obs, q_obs))
```

```
  for t in prange(1, T_obs + 1):
    dif_Beta_t = Beta_usage[t] - (M_G[t] @ Beta_usage[t - 1])
    dif_Y_t = M_Y[t] - (M_X[t] @ Beta_usage[t])
    par_Beta += ((dif_Beta_t.T @ inv_VW_usage) @ dif_Beta_t)
    par_Y += ((dif_Y_t.T @ inv_VB_usage) @ dif_Y_t)
  df = init_a_Sigma + p_obs + T_obs*p_obs + N_obs*T_obs

  sample = chol_rIW(M = par_init + par_Beta + par_Y,
                    nu = df,
                    n_seed = it)

  return sample
```

B.1.2.3 Sampling from the full conditional distribution of $\beta_0, \beta_1, \ldots, \beta_T$

```
@njit
def FF(init_M0: float64[:,:], init_C0: float64[:,:], M_Y: float64[:,:,:],
       M_X: float64[:,:,:], M_G: float64[:,:,:], V_usage: float64,
       phi_usage: float64, D_usage: float64[:,:],
       W_usage: float64[:,:]) -> (float64[:,:,:], float64[:,:,:]):

  T_obs = M_Y.shape[0] - 1
  q_obs, p_obs = M_Y.shape[2], M_X.shape[2]

  B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)
  cov_usage = V_usage * B_usage
  M_M = np.zeros((T_obs + 1, p_obs, q_obs))
  M_C = np.zeros((T_obs + 1, p_obs, p_obs))
  for t in range(0, T_obs + 1):
    if t == 0:
      M_M[t] = init_M0
      M_C[t] = V_usage * init_C0
    else:
      E_t = V_usage * W_usage + (M_G[t] @ M_C[t - 1]) @ M_G[t].T
      Q_t = cov_usage + (M_X[t] @ E_t) @ M_X[t].T
      inv_Q_t = np.linalg.inv(Q_t)
      aux_prod_t = (E_t @ M_X[t].T) @ inv_Q_t
      aux_dif_t = M_Y[t] - (M_X[t] @ M_G[t]) @ M_M[t - 1]
      M_M[t] = (M_G[t] @ M_M[t - 1]) + (aux_prod_t @ aux_dif_t)
```

```
        M_C[t] = E_t - (aux_prod_t @ M_X[t]) @ E_t

    return M_M, M_C

@njit
def BS_t(C_now: float64[:,:], M_now: float64[:,:], G_now: float64[:,:],
         G_next: float64[:,:], Beta_next: float64[:,:],
         V: float64, W: float64[:,:]) -> (float64[:,:], float64[:,:]):
    inv_VW = (1 / V) * np.linalg.inv(W)
    inv_C_now = np.linalg.inv(C_now)
    inv_H_now = inv_C_now + ((G_now.T @ inv_VW) @ G_now)
    H_now = np.linalg.inv(inv_H_now)
    h_now = (inv_C_now @ M_now) + ((G_next @ inv_VW) @ Beta_next)
    avg_now = H_now @ h_now
    return avg_now, H_now

@njit
def updt_Beta(it: int64, init_M0: float64[:,:], init_C0: float64[:,:],
              M_Y: float64[:,:,:], M_X: float64[:,:,:], M_G: float64[:,:,:],
              D_usage: float64[:,:], W_usage: float64[:,:], V_usage: float64,
              phi_usage: float64, Sigma_usage: float64[:,:]) -> float64[:,:,:]:

    T_obs = M_Y.shape[0] - 1
    q_obs, p_obs = M_Y.shape[2], M_X.shape[2]

    M_M, M_C = FF(init_M0 = init_M0,
                  init_C0 = init_C0,
                  M_Y = M_Y,
                  M_X = M_X,
                  M_G = M_G,
                  V_usage = V_usage,
                  W_usage = W_usage,
                  D_usage = D_usage,
                  phi_usage = phi_usage)

    sample = np.zeros((T_obs + 1, p_obs, q_obs))
    for x in range(0, T_obs + 1):
      t = T_obs - x
      if t == T_obs:
```

```
      sample[t] = chol_rMVN(avg = M_M[t],
                            left = M_C[t],
                            right = Sigma_usage,
                            n_seed = it)
    else:
      aux_BS_t = BS_t(C_now = M_C[t],
                      M_now = M_M[t],
                      G_now = M_G[t],
                      G_next = M_G[t + 1],
                      Beta_next = sample[t + 1],
                      W = W_usage,
                      V = V_usage)
      sample[t] = chol_rMVN(avg = aux_BS_t[0],
                            left = aux_BS_t[1],
                            right = Sigma_usage,
                            n_seed = it)

  return sample
```

B.1.2.4 Sampling from the full conditional distribution of ϕ

```
@njit(parallel = True)
def log_krnl_phi(par: float64, init_a_phi: float64, init_b_phi: float64,
                 M_Y: float64[:,:,:], M_X: float64[:,:,:],
                 D_usage: float64[:,:], V_usage: float64,
                 Beta_usage: float64[:,:,:],
                 Sigma_usage: float64[:,:]) -> float64:
  T_obs = M_Y.shape[0] - 1
  q_obs = M_Y.shape[2]
  if par <= 0:
    return -inf
  else:
    log_prior = (init_a_phi - 1)*log(par) - init_b_phi*par
    B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = par)
    log_det_B_usage = np.linalg.slogdet(B_usage)[1]
    aux = np.zeros((q_obs, q_obs))
    left = V_usage * B_usage
    inv_left, inv_right = np.linalg.inv(left), np.linalg.inv(Sigma_usage)
    for t in prange(1, T_obs + 1):
```

```
      dif_t = M_Y[t] - (M_X[t] @ Beta_usage[t])
      aux += (dif_t.T @ inv_left) @ (dif_t @ inv_right)
    log_LH = - 0.5*np.trace(aux) - 0.5*T_obs*q_obs*log_det_B_usage
    return log_prior + log_LH

@njit
def updt_phi(it: int64, init_a_phi: float64, init_b_phi: float64,
             D_usage: float64[:,:], M_Y: float64[:,:,:], M_X: float64[:,:,:],
             Beta_usage: float64[:,:,:], V_usage: float64, phi_usage: float64,
             Sigma_usage: float64[:,:], delta2: float64) -> float64:

  # Proposing a new value from the inverse-normal distribution
  phi_prop = rIN(n_seed = it, mu = phi_usage, sigma2 = delta2)

  # Generating a random number between zero and one in log scale
  log_u = log_rUnif(it)

  # Computing the log of the Metropolis ratio
  log_num = +logpdf_IN(mean = phi_prop,
                       sigma2 = delta2,
                       x = phi_usage) + log_krnl_phi(par = phi_prop,
                                                     init_a_phi = init_a_phi,
                                                     init_b_phi = init_b_phi,
                                                     D_usage = D_usage,
                                                     M_Y = M_Y,
                                                     M_X = M_X,
                                                     Beta_usage = Beta_usage,
                                                     V_usage = V_usage,
                                                     Sigma_usage = Sigma_usage)
  log_den = +logpdf_IN(mean = phi_usage,
                       sigma2 = delta2,
                       x = phi_prop) + log_krnl_phi(par = phi_usage,
                                                    init_a_phi = init_a_phi,
                                                    init_b_phi = init_b_phi,
                                                    D_usage = D_usage,
                                                    M_Y = M_Y,
                                                    M_X = M_X,
                                                    Beta_usage = Beta_usage,
                                                    V_usage = V_usage,
```

```
                                                    Sigma_usage = Sigma_usage)
  log_rho = log_num - log_den
  log_alpha = min([0, log_rho])

  if isfinite(log_alpha) and log_u <= log_alpha:
    return phi_prop
  else:
    return phi_usage
```

B.1.2.5 Sampling from the full conditional distribution of D

```
@njit(parallel = True)
def log_krnl_D(par: float64[:,:], S_obs: float64[:,:], M_X: float64[:,:,:],
               M_Y: float64[:,:,:], R_usage: float64[:,:],
               sigma2d_usage: float64[:,:], Sigma_usage: float64[:,:],
               V_usage: float64, Beta_usage: float64[:,:,:],
               phi_usage: float64) -> float64:
  T_obs = M_Y.shape[0] - 1
  q_obs = M_Y.shape[2]
  dif = par - S_obs
  inv_R_usage = np.linalg.inv(R_usage)
  inv_sigma2d_usage = np.linalg.inv(sigma2d_usage)
  krnl_prior = (dif.T @ inv_sigma2d_usage) @ (dif @ inv_R_usage)
  log_prior = -0.5*np.trace(krnl_prior)

  B_usage = nb_BR(matrix = par, need_transp = True, scalar = phi_usage)
  log_det_B_usage = np.linalg.slogdet(B_usage)[1]

  aux = np.zeros((q_obs, q_obs))
  left = V_usage * B_usage
  inv_left, inv_right = np.linalg.inv(left), np.linalg.inv(Sigma_usage)
  for t in prange(1, T_obs + 1):
    dif_t = M_Y[t] - (M_X[t] @ Beta_usage[t])
    aux += (dif_t.T @ inv_left) @ (dif_t @ inv_right)
  log_LH = - 0.5*np.trace(aux) - 0.5*T_obs*q_obs*log_det_B_usage

  return log_prior + log_LH
```

B.1.2.5.1 Using the Metropolis-Hastings algorithm

```
@njit
def updt_D(it: int64, S_obs: float64[:,:], M_X: float64[:,:,:],
           M_Y: float64[:,:,:], Beta_usage: float64[:,:,:], V_usage: float64,
           phi_usage: float64, sigma2d_usage: float64[:,:],
           D_usage: float64[:,:], R_usage: float64[:,:],
           Sigma_usage: float64[:,:], warm_up: int64, acc_now: int64,
           delta_usage: float64[:,:], size = 50, rate = 0.44,
           cte = 0.01) -> (float64[:,:], float64[:,:]):
  D_prev, delta_prev = D_usage.copy(), delta_usage.copy()
  N_obs = M_Y.shape[1]
  for i in range(0, 2):
    for j in range(2, N_obs):
      D_prop = D_prev.copy()
      delta_now = delta_prev.copy()
      np.random.seed(it + i + j)
      D_prop_ij = np.random.normal(loc = D_prev[i,j],
                                   scale = delta_now[i,j])
      D_prop[i,j] = D_prop_ij
      lcd_D_prop = log_krnl_D(par = D_prop,
                              S_obs = S_obs,
                              M_X = M_X,
                              M_Y = M_Y,
                              Sigma_usage = Sigma_usage,
                              Beta_usage = Beta_usage,
                              V_usage = V_usage,
                              phi_usage = phi_usage,
                              sigma2d_usage = sigma2d_usage,
                              R_usage = R_usage)
      lcd_D_prev = log_krnl_D(par = D_prev,
                              S_obs = S_obs,
                              M_X = M_X,
                              M_Y = M_Y,
                              Sigma_usage = Sigma_usage,
                              Beta_usage = Beta_usage,
                              V_usage = V_usage,
                              phi_usage = phi_usage,
                              sigma2d_usage = sigma2d_usage,
                              R_usage = R_usage)
```

```
      log_rho = lcd_D_prop - lcd_D_prev
      log_alpha = min([0, log_rho])
      log_u = log_rUnif(it + i + j)

      seq = [j for j in range(0, warm_up + 1, size)]
      if it in seq:
        frac = acc_now[i,j] / it
        if frac < rate:
          delta_now_ij = exp(log(delta_now[i,j]) - min(cte, 1/sqrt(it)))
          delta_now[i,j] = delta_now_ij
        else:
          delta_now_ij = exp(log(delta_now[i,j]) + min(cte, 1/sqrt(it)))
          delta_now[i,j] = delta_now_ij
        if isfinite(log_alpha) and log_u <= log_alpha:
          D_now = D_prop
        else:
          D_now = D_prev
      else:
        if isfinite(log_alpha) and log_u <= log_alpha:
          D_now = D_prop
        else:
          D_now = D_prev
      D_prev = D_now.copy()
      delta_prev = delta_now.copy()

  return D_now, delta_now
```

B.1.2.5.2 Using the slice sampler

```
@njit
def updt_D(it: int64, S_obs: float64[:,:], M_X: float64[:,:,:],
           M_Y: float64[:,:,:], Beta_usage: float64[:,:,:], V_usage: float64,
           phi_usage: float64, sigma2d_usage: float64[:,:],
           D_usage: float64[:,:], R_usage: float64[:,:],
           Sigma_usage: float64[:,:]) -> float64[:,:]:
  N_obs = M_Y.shape[1]
  D_prev = D_usage.copy()
  w = 1
  m = +inf
```

```
lower = -inf
upper = +inf
for i in range(0, 2):
  for j in range(2, N_obs):
    np.random.seed(it + i + j)
    D_prop = D_prev.copy()

    x0 = D_prop[i,j]
    x0_D = D_prop.copy()

    # Find the log density at the initial point, if not already known.
    gx0 = log_krnl_D(par = x0_D,
                     S_obs = S_obs,
                     M_X = M_X,
                     M_Y = M_Y,
                     Sigma_usage = Sigma_usage,
                     Beta_usage = Beta_usage,
                     V_usage = V_usage,
                     phi_usage = phi_usage,
                     sigma2d_usage = sigma2d_usage,
                     R_usage = R_usage)
    # Determine the slice level, in log terms.
    logy = gx0 - np.random.exponential(scale = 1.0,
                                       size = 1).item()

    # Find the initial interval to sample from.
    u = np.random.uniform(0, w)
    L = x0 - u
    R = x0 + (w - u) # should guarantee that x0 is in [L,R], even with roundoff

    L_D = D_prop.copy()
    L_D[i,j] = L
    R_D = D_prop.copy()
    R_D[i,j] = R

    # Expand the interval until its ends are outside the slice, or until
    # the limit on steps is reached.
    if isfinite(m) is False:
      while L > lower and logy < log_krnl_D(par = L_D,
```

```
                                        S_obs = S_obs,
                                        M_X = M_X,
                                        M_Y = M_Y,
                                        Sigma_usage = Sigma_usage,
                                        Beta_usage = Beta_usage,
                                        V_usage = V_usage,
                                        phi_usage = phi_usage,
                                        sigma2d_usage = sigma2d_usage,
                                        R_usage = R_usage):
      L = L - w
      L_D[i,j] = L
    while R < upper and logy < log_krnl_D(par = R_D,
                                        S_obs = S_obs,
                                        M_X = M_X,
                                        M_Y = M_Y,
                                        Sigma_usage = Sigma_usage,
                                        Beta_usage = Beta_usage,
                                        V_usage = V_usage,
                                        phi_usage = phi_usage,
                                        sigma2d_usage = sigma2d_usage,
                                        R_usage = R_usage):
      R = R + w
      R_D[i,j] = R
  else:
    pass

  # Shrink interval to lower and upper bounds.
  if L < lower:
    L = lower
    L_D[i,j] = L
  else:
    pass
  if R > upper:
    R = upper
    R_D[i,j] = R
  else:
    pass

  # Sample from the interval, shrinking it on each rejection.
```

```
    x1_D = D_prop.copy()
    x1 = np.random.uniform(L, R)
    x1_D[i,j] = x1
    gx1 = log_krnl_D(par = x1_D,
                     S_obs = S_obs,
                     M_X = M_X,
                     M_Y = M_Y,
                     Sigma_usage = Sigma_usage,
                     Beta_usage = Beta_usage,
                     V_usage = V_usage,
                     phi_usage = phi_usage,
                     sigma2d_usage = sigma2d_usage,
                     R_usage = R_usage)

    while gx1 < logy:
      x1 = np.random.uniform(L, R)
      x1_D[i,j] = x1
      gx1 = log_krnl_D(par = x1_D,
                       S_obs = S_obs,
                       M_X = M_X,
                       M_Y = M_Y,
                       Sigma_usage = Sigma_usage,
                       Beta_usage = Beta_usage,
                       V_usage = V_usage,
                       phi_usage = phi_usage,
                       sigma2d_usage = sigma2d_usage,
                       R_usage = R_usage)
      if x1 > x0:
        R = x1
      else:
        L = x1

    D_prev = x1_D.copy()

  return x1_D
```

Here we adapted to Python an R code for univariate slice sampling written by Neal (2008).

B.1.3 First simulation study – Generating data and specifying hyperparameters

```
# Sites in the unit square
S = np.vstack([[0.0, 0.0], [1.0, 1.0],
               [0.0, 1/3], [0.0, 2/3], [0.0, 1.0],
               [1/3, 0.0], [1/3, 1/3], [1/3, 2/3], [1/3, 1.0],
               [1/2, 1/2],
               [2/3, 0.0], [2/3, 1/3], [2/3, 2/3], [2/3, 1.0],
               [1.0, 0.0], [1.0, 1/3], [1.0, 2/3]]).T

# The number of sites and the number of replications (or times)
N = S.shape[1]
T = 1000 # Run with 10, 100 and 1000
p, q = 2, 2
seed_v = 1000

D = np.zeros((2, N))
D[:,0:2] = S[:,0:2]

psi = -2*log(0.05)/np.max(distance.cdist(S.T, S.T, metric = 'sqeuclidean'))
R_d = nb_BR(matrix = S,
            need_transp = True,
            squared = True,
            scalar = psi)
R_12 = R_d[0:2,0:2]
R_3N = R_d[2:18,2:18]
R_star = R_d[2:18,0:2]
S_cond = S[:,2:(N + 1)] + np.matmul(D[:,0:2] - S[:,0:2],
                                    np.linalg.inv(R_12) @ R_star.T)
R_cond = R_3N - np.matmul(R_star, np.linalg.inv(R_12) @ R_star.T)

sigma2d = np.array([[0.500, 0.000],
                    [0.000, 0.500]])

D[:,2:(N + 1)] = rMVN(n = 1,
                      avg = S_cond,
                      left = sigma2d,
                      right = R_cond,
                      n_seed = seed_v)

V = 0.4
```

```
a_V, b_V = 0.001, 0.001

Sigma = np.array([[1.0, 0.7],
                  [0.7, 1.0]])
a_Sigma, b_Sigma = 0.001, 0.001*np.identity(q)

phi = 0.1
a_phi, b_phi = 0.001, 0.001

# Using the true deformations to specify B
B = nb_BR(matrix = D, need_transp = True, scalar = phi)

M0, C0 = np.zeros((p, q)), 1.0*np.identity(p)
W = 1.0*np.identity(p)
Beta = np.zeros((T + 1, p, q))

G = np.zeros((T + 1, p, p))
for t in range(0, T + 1):
  if t == 0:
    Beta[t] = rMVN(n = 1,
                   avg = M0,
                   left = V*C0,
                   right = Sigma,
                   n_seed = seed_v + t)
  else:
    G[t] = np.identity(p)
    Beta[t] = rMVN(avg = np.matmul(G[t], Beta[t-1]),
                   left = V*W,
                   right = Sigma,
                   n = 1,
                   n_seed = seed_v + t)

# Likelihood function
Y, X = np.zeros((T + 1, N, q)), np.zeros((T + 1, N, p))
for t in range(1, T + 1):
  X[t] = np.vstack([np.ones(N),
                    uniform.rvs(loc = 0,
                                scale = 1,
                                size = N,
```

```
                                    random_state = seed_v + t)]).T
  Y[t] = rMVN(n = 1,
              avg = np.matmul(X[t], Beta[t]),
              left = V*B,
              right = Sigma,
              n_seed = seed_v + t)

# R correlation matrix
def comp_R(S_obs, psi_obs):
  if isfinite(psi_obs):
    R_N = nb_BR(matrix = S_obs,
                need_transp = True,
                squared = True,
                scalar = psi_obs)
  else:
    N_obs = S_obs.shape[1]
    R_N = np.identity(N_obs)
  return R_N

# Configuration of the distortion level
psi_usage = 5.0
R_usg = comp_R(S_obs = S, psi_obs = psi_usage)
sigma2d_usg = np.cov(S)
```

B.1.4 Second simulation study – Generating data and specifying hyperparameters

```
# Sites in unit square
S = np.vstack([[1/5, 1/5], [4/5, 4/5], [1/5, 4/5], [4/5, 1/5],
                           [1/5, 2/5], [1/5, 3/5],
               [2/5, 1/5], [2/5, 2/5], [2/5, 3/5], [2/5, 4/5],
               [3/5, 1/5], [3/5, 2/5], [3/5, 3/5], [3/5, 4/5],
                           [4/5, 2/5], [4/5, 3/5]]).T
S_i = np.array([[0.7, 0.5],
                [0.5, 0.7],
                [0.3, 0.3]]).T
S_tot = np.hstack([S, S_i])

# The number of sites and the number of replications (or times)
```

```
N, N_i, N_tot = S.shape[1], S_i.shape[1], S_tot.shape[1]
T, T_p = 100, 10
T_tot = T + T_p
p, q = 2, 2
seed_v = 500

V = 1.0
a_V, b_V = 0.001, 0.001

Sigma = np.array([[1.00, 0.80],
                  [0.80, 1.00]])
a_Sigma, b_Sigma = 0.001, 0.001*np.identity(q)

phi = 0.5
a_phi, b_phi = 0.001, 0.001

### B and B_i are specified by an anisotropic structure
xi_x = 1
xi_y = 3
Lambda1 = np.diag([xi_x, xi_y])
Sh_x = 0.05
Sh_y = 0.00
Lambda2 = np.array([[1,    Sh_x],
                    [Sh_y,    1]])
eta = -pi/6
Lambda3 = np.array([[np.cos(eta), -np.sin(eta)],
                    [np.sin(eta),  np.cos(eta)]])
Lambda = Lambda1 @ Lambda2 @ Lambda3
A = Lambda.T @ Lambda
u = np.zeros(2)
D = np.zeros((2, N))
D_i = np.zeros((2, N_i))
for n in range(1, N_tot + 1):
  if n <= 2:
    D[:,n - 1] = S[:,n - 1]
  elif n >= 3 and n <= N:
    D[:,n - 1] = Lambda @ S[:,n - 1] + u
  else:
    D_i[:,N_tot - n - 1] = Lambda @ S_i[:,N_tot - n - 1] + u
```

```
B_aug = nb_BR(matrix = np.hstack([D, D_i]),
              need_transp = True, scalar = phi)

M0, C0 = np.zeros((p, q)), 1.0*np.identity(p)
W = 1.0*np.identity(p)
Beta_tot = np.zeros((T_tot + 1, p, q))
Beta, Beta_p = np.zeros((T + 1, p, q)), np.zeros((T_p, p, q))

G = np.zeros((T_tot + 1, p, p))
for t in range(0, T_tot + 1):
  if t == 0:
    G[t] = np.identity(p)
    Beta_tot[t] = rMVN(n = 1,
                       avg = M0,
                       left = V*C0,
                       right = Sigma,
                       n_seed = seed_v + t)
  else:
    G[t] = np.identity(p)
    Beta_tot[t] = rMVN(avg = np.matmul(G[t], Beta_tot[t-1]),
                       left = V*W,
                       right = Sigma,
                       n = 1,
                       n_seed = seed_v + t)

for t in range(0, T_tot + 1):
  if t <= T:
    Beta[t] = Beta_tot[t]
  else:
    Beta_p[t - T - 1] = Beta_tot[t]

# X
X, X_i = np.zeros((T_tot + 1, N, p)), np.zeros((T_tot + 1, N_i, p))
for t in range(1, T_tot + 1):
  X[t] = np.vstack([np.ones(N),
                    uniform.rvs(loc = 0,
                                scale = 1,
                                size = N,
```

```
                                    random_state = seed_v + t)]).T
  X_i[t] = np.vstack([np.ones(N_i),
                      uniform.rvs(loc = 0,
                                  scale = 1,
                                  size = N_i,
                                  random_state = seed_v + t)]).T

# Response matrices
Y_aug = np.zeros((T_tot + 1, N + N_i, q))
for t in range(1, T_tot + 1):
  Y_aug[t] = rMVN(n = 1,
                  avg = np.matmul(np.vstack([X[t], X_i[t]]), Beta_tot[t]),
                  left = V*B_aug,
                  right = Sigma,
                  n_seed = seed_v + t)

Y, Y_i = np.zeros((T+1,N,q)), np.zeros((T+1,N_i,q))
Y_p, Y_p_i = np.zeros((T_p,N,q)), np.zeros((T_p,N_i,q))
for t in range(1, T_tot + 1):
  if t <= T:
    Y[t] = Y_aug[t][0:N,:]
    Y_i[t] = Y_aug[t][N:N_tot,:]
  else:
    Y_p[t - T - 1] = Y_aug[t][0:N,:]
    Y_p_i[t - T - 1] = Y_aug[t][N:N_tot,:]

# R correlation matrix
def comp_R(S_obs, psi_obs):
  if isfinite(psi_obs):
    R_N = nb_BR(matrix = S_obs,
                need_transp = True,
                squared = True,
                scalar = psi_obs)
  else:
    N_obs = S_obs.shape[1]
    R_N = np.identity(N_obs)
  return R_N

# Configuration of the distortion level
```

```
psi_usage = 15.0
R_usg = comp_R(S_obs = S, psi_obs = psi_usage)
sigma2d_usg = np.cov(S)
```

B.1.5 Hybrid MCMC algorithm

```
############################ Creating objects to store the MCMC samples
burn_in = 5000
num_iter = 15000 + burn_in
thin = 15
smp_size = int((num_iter - burn_in)/thin)

e_phi = [0 for k in range(0, smp_size)]
e_V = [0 for k in range(0, smp_size)]
e_Beta = [np.zeros((T + 1, p, q)) for k in range(0, smp_size)]
e_Sigma = [np.zeros((q, q)) for k in range(0, smp_size)]
e_D = [np.zeros((2, N)) for k in range(0, smp_size)]

############################ Initial values for MCMC estimation
phi_prev = 2.0
V_prev = 2.0
Sigma_prev = 2.0*np.identity(q)
Beta_prev = np.zeros((T + 1, p, q))
D_prev = np.copy(S)
delta_phi = 65.0

############################ Metropolis-within-Gibbs algorithm
# ind will vary between 0 and smp_size - 1 (i.e. k = 1, ..., K)
ind = 0
acc_phi, acc_D = 0, np.zeros((2, N))

# Algorithm and processing time in seconds
start = timer()
for j in range(1, num_iter + 1):
  Beta_curr = updt_Beta(it = seed_v + j,
                        init_M0 = M0,
                        init_C0 = C0,
                        M_Y = Y,
                        M_X = X,
```

```
                        M_G = G,
                        D_usage = D_prev,
                        W_usage = W,
                        phi_usage = phi_prev,
                        V_usage = V_prev,
                        Sigma_usage = Sigma_prev)
V_curr = updt_V(it = seed_v + j,
                init_a_V = a_V,
                init_b_V = b_V,
                init_M0 = M0,
                init_C0 = C0,
                M_Y = Y,
                M_X = X,
                M_G = G,
                D_usage = D_prev,
                Beta_usage = Beta_curr,
                phi_usage = phi_prev,
                Sigma_usage = Sigma_prev,
                W_usage = W)
phi_curr = updt_phi(it = seed_v + j,
                    init_a_phi = a_phi,
                    init_b_phi = b_phi,
                    D_usage = D_prev,
                    M_Y = Y,
                    M_X = X,
                    Beta_usage = Beta_curr,
                    V_usage = V_curr,
                    phi_usage = phi_prev,
                    Sigma_usage = Sigma_prev,
                    delta2 = delta_phi)
Sigma_curr = updt_Sigma(it = seed_v + j,
                        init_a_Sigma = a_Sigma,
                        init_b_Sigma = b_Sigma,
                        init_M0 = M0,
                        init_C0 = C0,
                        M_Y = Y,
                        M_X = X,
                        M_G = G,
                        Beta_usage = Beta_curr,
```

```
                          W_usage = W,
                          phi_usage = phi_curr,
                          V_usage = V_curr,
                          D_usage = D_prev)
D_curr = updt_D(it = seed_v + j,
                S_obs = S,
                M_X = X,
                M_Y = Y,
                Sigma_usage = Sigma_curr,
                Beta_usage = Beta_curr,
                V_usage = V_curr,
                phi_usage = phi_curr,
                sigma2d_usage = sigma2d_usg,
                R_usage = R_usg,
                D_usage = D_prev)

if abs(phi_curr - phi_prev) != 0:
  acc_phi += 1
else:
  pass

for r in range(0, 2):
  for n in range(0, N):
    if abs(D_curr[r,n] - D_prev[r,n]) != 0:
      acc_D[r,n] += 1
    else:
      pass

if j > burn_in and j % thin == 0:
  e_V[ind] = V_curr
  e_Beta[ind] = Beta_curr
  e_phi[ind] = phi_curr
  e_Sigma[ind] = Sigma_curr
  e_D[ind] = D_curr
  ind += 1
else:
  pass

V_prev = V_curr
```

```
  phi_prev = phi_curr
  Beta_prev = Beta_curr
  Sigma_prev = Sigma_curr
  D_prev = D_curr

  print(j)
end = timer()
print(end - start)
```

The above algorithm uses the slice sampler to sample from $\mathbf{D}$, performing the function described in Appendix B.1.2.5.2. To use the Metropolis-Hastings algorithm to sample from $\mathbf{D}$, use the function given in Appendix B.1.2.5.1 and let `D_curr` receive the code below. To use the isotropic model $\mathcal{M}_{\text{I}}$, just let `D_curr = np.copy(S)`.

```
  D_curr, delta_D = updt_D(it = seed_v + j,
                           S_obs = S,
                           M_X = X,
                           M_Y = Y,
                           Sigma_usage = Sigma_curr,
                           Beta_usage = Beta_curr,
                           V_usage = V_curr,
                           phi_usage = phi_curr,
                           sigma2d_usage = sigma2d_usg,
                           R_usage = R_usg,
                           D_usage = D_prev,
                           delta_usage = delta_D,
                           warm_up = burn_in,
                           acc_now = acc_D)
```

B.1.6 Forecasting and interpolation

B.1.6.1 Forecasting

```
e_Y = [np.zeros((T + 1, N, q)) for j in range(0, smp_size)]
for k in range(0, int(smp_size)):
  for t in range(1, T + 1):
    e_Y[k][t] = Y[t]

e_Beta_p = [np.zeros((T_p, p, q)) for k in range(0, smp_size)]
```

```
for k in range(0, int(smp_size)):
  for t in range(1, T_p + 1):
    aux_t = G[T + t]
    cov_t = W.copy()
    if t > 1:
      for t2 in reversed(range(2, t + 1)):
        aux_t = np.matmul(aux_t, G[T + t2])
        cov_t += np.matmul(aux_t.T, W @ aux_t)
    else:
      pass
    e_Beta_p[k][t - 1] = rMVN(n = 1,
                              avg = np.matmul(aux_t, e_Beta[k][T]),
                              left = e_V[k] * cov_t,
                              right = e_Sigma[k],
                              n_seed = seed_v + k*t)

e_Y_p = [np.zeros((T_p, N, q)) for k in range(0, smp_size)]
for k in range(0, int(smp_size)):
  for t in range(1, T_p + 1):
    e_Y_p[k][t - 1] = rMVN(n = 1,
                           avg = np.matmul(X[T + t], e_Beta_p[k][t - 1]),
                           left = e_V[k] * nb_BR(matrix = e_D[k],
                                                 need_transp = True,
                                                 scalar = e_phi[k]),
                           right = e_Sigma[k],
                           n_seed = seed_v + k*t)
```

B.1.6.2 Interpolation

Below we present an implementation for Algorithms 8 and 9.

```
if isfinite(psi_usage) is False:
  R_tot = np.identity(N_tot)
else:
  R_tot = nb_BR(matrix = S_tot,
                need_transp = True,
                squared = True,
                scalar = psi_usage)
R_gu = R_tot[0:N,N:N_tot]
R_1N = R_tot[0:N,0:N]
```

```
inv_R_1N = np.linalg.inv(R_1N)
R_star = R_tot[N:N_tot,N:N_tot]
D_cov = R_star - np.matmul(R_gu.T, inv_R_1N @ R_gu)

e_D_i = [np.zeros((2, N_i)) for k in range(0, smp_size)]
for k in range(0, int(smp_size)):
  D_mean_k = S_i + np.matmul(e_D[k] - S, inv_R_1N @ R_gu)
  e_D_i[k] = rMVN(n = 1,
                  avg = D_mean_k,
                  left = sigma2d_usg,
                  right = D_cov,
                  n_seed = k + seed_v)

e_Y_i = [np.zeros((T + 1, N_i, q)) for j in range(0, smp_size)]
e_Y_p_i = [np.zeros((T_p, N_i, q)) for j in range(0, smp_size)]
for k in range(0, int(smp_size)):
  B_k = nb_BR(matrix = e_D[k],
              need_transp = True,
              scalar = e_phi[k])
  inv_B_k = np.linalg.inv(B_k)
  dist_oi_k = np.zeros((N, N_i))
  for n in range(0, N):
    for m in range(0, N_i):
      dist_oi_k[n, m] = dist(e_D[k][:,n], e_D_i[k][:,m])
  B_oi_k = np.exp(-e_phi[k] * dist_oi_k)
  B_io_k = B_oi_k.T
  B_i_k = nb_BR(matrix = e_D_i[k],
                need_transp = True,
                scalar = e_phi[k])
  aux_B_k = np.matmul(B_io_k, inv_B_k)
  cov_B_k = B_i_k - np.matmul(aux_B_k, B_oi_k)
  for t in range(1, T + 1):
    avg_t_k = (X_i[t] @ e_Beta[k][t]) + (aux_B_k @ (Y[t]-(X[t] @ e_Beta[k][t])))
    e_Y_i[k][t] = rMVN(n = 1,
                       avg = avg_t_k,
                       left = e_V[k] * cov_B_k,
                       right = e_Sigma[k],
                       n_seed = seed_v + k*t)
  for t in range(T + 1, T_tot + 1):
```

```
    avg1 = (X_i[t] @ e_Beta_p[k][t - T - 1])
    avg2 = aux_B_k @ (e_Y_p[k][t - T - 1]-(X[t] @ e_Beta_p[k][t - T - 1]))
    avg_sum = avg1 + avg2
    e_Y_p_i[k][t - T - 1] = rMVN(n = 1,
                                 avg = avg_sum,
                                 left = e_V[k] * cov_B_k,
                                 right = e_Sigma[k],
                                 n_seed = seed_v + k*t)
```

B.1.7 Additional codes (PMSE and DIC)

```
def PMSE(M_Y_i: np.ndarray, e_Y_i = e_Y_i) -> np.float64:
  K = np.shape(e_Y_i)[0]
  T = np.shape(e_Y_i)[1] - 1
  N_i = np.shape(e_Y_i)[2]
  q = np.shape(e_Y_i)[3]
  square_dif_sum = 0
  for n in np.arange(0, N_i):
    for i in np.arange(0, q):
      for t in np.arange(1, T + 1):
        aux_sum = 0
        for k in np.arange(0, K):
          aux_sum += e_Y_i[k][t][n,i]
        aux_mean = aux_sum / K
        square_dif_sum += pow(M_Y_i[t][n,i] - aux_mean, 2)
  square_dif_mean = square_dif_sum / (N_i*q*T)
  return square_dif_mean
PMSE_value = PMSE(M_Y_i = Y_i)

def DIC(M_Y: np.ndarray, M_X: np.ndarray, e_Beta = e_Beta, e_V = e_V,
        e_phi = e_phi, e_D = e_D, e_Sigma = e_Sigma, df = 1) -> np.float64:
  K = np.shape(e_Beta)[0]
  T = np.shape(e_Beta)[1] - 1
  p = np.shape(e_Beta)[2]
  q = np.shape(e_Beta)[3]
  N = np.shape(e_D)[2]
  L = np.zeros((K, T))
  for k in range(0, K):
    leftcov_k = e_V[k] * nb_BR(matrix = e_D[k],
```

```
                                need_transp = True,
                                scalar = e_phi[k])
    for t in range(1, T + 1):
      L[k,t-1] = matrix_normal.logpdf(X = M_Y[t],
                                      mean = np.matmul(M_X[t],
                                                       e_Beta[k][t]),
                                      rowcov = leftcov_k,
                                      colcov = e_Sigma[k])
  Beta_sum = np.zeros((T + 1, p, q))
  V_sum = 0
  phi_sum = 0
  D_sum = np.zeros((2, N))
  Sigma_sum = np.zeros((q, q))
  for k in range(0, K):
    V_sum += e_V[k]
    phi_sum += e_phi[k]
    Sigma_sum += e_Sigma[k]
    D_sum += e_D[k]
    for t in range(0, T + 1):
      Beta_sum[t] += e_Beta[k][t]
  Beta_bar = Beta_sum / K
  V_bar = V_sum / K
  phi_bar = phi_sum / K
  D_bar = D_sum / K
  Sigma_bar = Sigma_sum / K
  leftcov_bar = V_bar * nb_BR(matrix = D_bar,
                              need_transp = True,
                              scalar = phi_bar)
  F = np.zeros(T)
  for t in range(1, T + 1):
    F[t-1] += matrix_normal.logpdf(X = M_Y[t],
                                   mean = np.matmul(M_X[t],
                                                    Beta_bar[t]),
                                   rowcov = leftcov_bar,
                                   colcov = Sigma_bar)
  return -(4/K)*np.sum(L) + 2*np.sum(F)
DIC_value = DIC(M_Y = Y, M_X = X)
```

B.2 Implementation of the pseudocodes of Chapter 3

B.2.1 Libraries and auxiliary functions

```
# Obtaining the matrices B or R
@njit(parallel = True)
def nb_BR(matrix: float64[:,:], need_transp: boolean, scalar: float64,
          squared = False) -> float64[:,:]:
  if need_transp is True:
    X = matrix.T
  else:
    X = matrix
  n, m = X.shape
  M = np.ones((n, n))
  for i in prange(0, n - 1):
    for j in prange(i + 1, n):
      dist_ij = 0.0
      for k in prange(0, m):
        dist_ij += pow(X[i, k] - X[j, k], 2)
      if squared is False:
        M[i, j] = exp(-scalar * sqrt(dist_ij)) # This is for B
      else:
        M[i, j] = exp(-scalar * dist_ij) # This is for R
      M[j, i] = M[i, j]

  return M

# Obtaining Y_obs_t, Y_mis_t, N_obs_t, N_mis_t, P_t, L_obs_t and L_mis_t
@njit
def Permut(vector: float64[:]):
  N = len(vector)
  Positions = [0]*N
  Identify = [0]*N

  i = 0
  for n in range(0, N):
    if np.isnan(vector[n]):
      pass
    else:
      i += 1
```

```
        Positions[n] = i
        Identify[n] = 1

    N_o = i
    N_m = N - N_o

    j = i
    for n in range(0, N):
      if np.isnan(vector[n]):
        j += 1
        Positions[n] = j
      else:
        pass

    Seq = np.array(Positions, dtype = np.int64) - 1
    P_t = np.eye(N)[:,Seq]
    L_o_t = np.hstack((np.eye(N_o), np.zeros((N_o, N_m))))
    L_m_t = np.hstack((np.zeros((N_m, N_o)), np.eye(N_m)))
    Subseq_o = np.array(Identify, dtype = bool_)
    Subseq_m = np.logical_not(Subseq_o)
    Y_o_t = vector[Subseq_o].reshape(N_o, 1)
    Y_m_t = vector[Subseq_m].reshape(N_m, 1)

    return Y_o_t, Y_m_t, N_o, N_m, P_t, L_o_t, L_m_t, Subseq_o

# Kronecker product
@njit
def nb_kp(X: float64[:,:], Y: float64[:,:]) -> float64[:,:]:
    return np.kron(X, Y)

# Sampling from multivariate normal distribution (fast, but less robust)
@njit("float64[:,:](int64,␣float64[:,:],␣float64[:,:])")
def chol_rmvN(n_seed: int64, avg: float64[:,:],
                cov: float64[:,:]) -> float64[:,:]:
    p = cov.shape[0]
    np.random.seed(n_seed)
    gen = np.linalg.cholesky(cov) @ np.random.standard_normal(p).reshape(p, 1)
    sample = avg + gen
    return sample
```

```
# Sampling from matrix-variate normal distribution (slow, but more robust)
def rMVN(n: np.uint8, avg: np.array, left: np.array,
         right: np.array, n_seed: np.int64) -> np.array:
  p, q = avg.shape
  vec_avg = np.matrix.flatten(avg, 'F')
  kron_cov = nb_kp(right, left)

  np.random.seed(seed = n_seed)
  vec_sample = np.random.multivariate_normal(mean = vec_avg,
                                             cov = kron_cov,
                                             size = n,
                                             check_valid = 'ignore')

  return np.reshape(vec_sample, (p, q), 'F')

# Sampling from matrix-variate normal distribution (fast, but less robust)
@njit
def chol_rMVN(n_seed: int64, avg: float64[:,:],
              left: float64[:,:], right: float64[:,:]) -> float64[:,:]:
  p, q = avg.shape
  vec_avg = (avg.T).flatten()
  kron_cov = np.kron(right, left)
  np.random.seed(n_seed)
  gen = np.linalg.cholesky(kron_cov) @ np.random.standard_normal(vec_avg.size)
  vec_sample = vec_avg + gen
  return vec_sample.reshape((q, p)).T

# Sampling from the inverse Wishart distribution
@njit
def chol_rIW(nu: float64, M: float64[:,:], n_seed: uint8) -> float64[:,:]:
  np.random.seed(n_seed)
  dim = M.shape[0]
  inv_M = np.linalg.inv(M)
  chol = np.linalg.cholesky(inv_M)
  df = nu + dim - 1
  foo = np.zeros((dim, dim))
  for i in range(dim):
    for j in range(i + 1):
```

```
      if i == j:
        foo[i, j] = sqrt(np.random.chisquare(df - (i + 1) + 1))
      else:
        foo[i, j] = np.random.standard_normal()
  return np.linalg.inv(chol @ (foo @ (foo.T @ chol.T)))

# Sampling from the Inverse-Normal Distribution
@njit
def rIN(n_seed: int64, mu: float64, sigma2: float64) -> float64:
  np.random.seed(n_seed)
  return np.random.wald(mu, sigma2)

# Sampling from the uniform distribution and applying the natural logarithm
@njit
def log_rUnif(n_seed: int64) -> float64:
  np.random.seed(n_seed)
  return log(np.random.rand())

# Logarithm of the PDF of the Inverse-Normal distribution
@njit
def logpdf_IN(x: float64, mean: float64, sigma2: float64) -> float64:
  term1 = 0.5*(log(sigma2) - log(2*pi*pow(x, 3)))
  term2 = -0.5*sigma2*pow(x - mean, 2) / (pow(mean, 2) * x)
  return term1 + term2
```

B.2.2 Full conditional distributions

B.2.2.1 Sampling from the full conditional distribution of V

```
@njit(parallel = True)
def updt_V(it: int64, M_x: float64[:,:,:], M_g: float64[:,:,:],
           W_usage: float64[:,:], init_a_V: float64, init_b_V: float64,
           init_m0: float64[:,:], init_C0: float64[:,:], D_usage: float64[:,:],
           phi_usage: float64, beta_usage: float64[:,:,:],
           Sigma_usage: float64[:,:], M_y: float64[:,:,:]) -> float64:
  T_obs = beta_usage.shape[0] - 1
  q_obs = Sigma_usage.shape[0]
  p_obs = W_usage.shape[0]
  N_obs = D_usage.shape[1]
```

```
  B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)
  Sig_B = np.kron(Sigma_usage, B_usage)
  Sig_C0 = np.kron(Sigma_usage, init_C0)
  Sig_W = np.kron(Sigma_usage, W_usage)
  inv_Sig_B = np.linalg.inv(Sig_B)
  inv_Sig_C0 = np.linalg.inv(Sig_C0)
  inv_Sig_W = np.linalg.inv(Sig_W)

  df = init_a_V + 0.5*(p_obs*q_obs + T_obs*p_obs*q_obs + T_obs*N_obs*q_obs)
  aux, par = 0.0, 0.0
  for t in prange(1, T_obs + 1):
    sub_t = beta_usage[t] - (M_g[t] @ beta_usage[t - 1])
    aux += ((sub_t.T @ inv_Sig_W) @ sub_t).item()
    mu_obs_t = M_x[t] @ beta_usage[t]
    y_obs_t = M_y[t]
    dif_obs_t = y_obs_t - mu_obs_t
    par += ((dif_obs_t.T @ inv_Sig_B) @ dif_obs_t).item()
  sub_0 = beta_usage[0] - init_m0
  aux_0 = ((sub_0.T @ inv_Sig_C0) @ sub_0).item()

  quantity = init_b_V + 0.5*(aux_0 + aux + par)

  np.random.seed(it)
  inv_sample = np.random.gamma(shape = df, scale = 1/quantity)

  return 1/inv_sample
```

B.2.2.2 Sampling from the full conditional distribution of Σ

```
@njit(parallel = True)
def updt_Sigma(it: int64, beta_usage: float64[:,:,:], M_y: float64[:,:,:],
               M_G: float64[:,:,:], M_X: float64[:,:,:], W_usage: float64[:,:],
               init_M0: float64[:,:], init_C0: float64[:,:], V_usage: float64,
               init_a_Sigma: float64, init_b_Sigma: float64[:,:],
               D_usage: float64[:,:], phi_usage: float64) -> float64[:,:]:

  T_obs = M_y.shape[0] - 1
  N_obs = D_usage.shape[1]
```

```
  q_obs = init_b_Sigma.shape[1]
  p_obs = M_X.shape[2]

  B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)

  M_Y = np.zeros((T_obs + 1, N_obs, q_obs))
  Beta_usage = np.zeros((T_obs + 1, p_obs, q_obs))
  M_Y[0] = (M_y[0].reshape(q_obs, N_obs)).T
  Beta_usage[0] = (beta_usage[0].reshape(q_obs, p_obs)).T

  inv_V_usage = 1 / V_usage
  inv_W_usage = np.linalg.inv(W_usage)
  inv_VW_usage = inv_V_usage * inv_W_usage
  inv_B_usage = np.linalg.inv(B_usage)
  inv_VB_usage = inv_V_usage * inv_B_usage
  inv_init_C0 = np.linalg.inv(init_C0)
  dif_0 = Beta_usage[0] - init_M0
  par_init = init_b_Sigma + ((dif_0.T @ (inv_V_usage * inv_init_C0)) @ dif_0)
  par_beta = np.zeros((q_obs, q_obs))
  par_Y = np.zeros((q_obs, q_obs))
  for t in prange(1, T_obs + 1):
    M_Y[t] = (M_y[t].reshape(q_obs, N_obs)).T
    Beta_usage[t] = (beta_usage[t].reshape(q_obs, p_obs)).T
    dif_beta_t = Beta_usage[t] - (M_G[t] @ Beta_usage[t - 1])
    dif_Y_t = M_Y[t] - (M_X[t] @ Beta_usage[t])
    par_beta += ((dif_beta_t.T @ inv_VW_usage) @ dif_beta_t)
    par_Y += ((dif_Y_t.T @ inv_VB_usage) @ dif_Y_t)
  df = init_a_Sigma + p_obs + T_obs*p_obs + N_obs*T_obs

  sample = chol_rIW(M = par_init + par_beta + par_Y,
                    nu = df,
                    n_seed = it)

  return sample
```

B.2.2.3 Sampling from the full conditional distribution of $\operatorname{vec}\boldsymbol{\beta}_0, \operatorname{vec}\boldsymbol{\beta}_1, \ldots, \operatorname{vec}\boldsymbol{\beta}_T$

```
@njit
def FF(init_m0: float64[:,:], init_C0: float64[:,:], M_y: float64[:,:,:],
```

```
        M_x: float64[:,:,:], M_g: float64[:,:,:], V_usage: float64,
        phi_usage: float64, D_usage: float64[:,:], Sigma_usage: float64[:,:],
        W_usage: float64[:,:]) -> (float64[:,:,:], float64[:,:,:]):

  T_obs = M_y.shape[0] - 1
  q_obs = Sigma_usage.shape[0]
  p_obs = W_usage.shape[0]
  N_obs = D_usage.shape[1]

  B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)
  VSig_W = V_usage * np.kron(Sigma_usage, W_usage)
  VSig_B = V_usage * np.kron(Sigma_usage, B_usage)

  cov_usage = V_usage * B_usage
  M_m = np.zeros((T_obs + 1, p_obs*q_obs, 1))
  M_C = np.zeros((T_obs + 1, p_obs*q_obs, p_obs*q_obs))
  M_m[0] = init_m0
  M_C[0] = V_usage * np.kron(Sigma_usage, init_C0)
  for t in range(1, T_obs + 1):
    E_t = VSig_W + (M_g[t] @ M_C[t-1]) @ M_g[t].T
    Q_t = VSig_B + (M_x[t] @ E_t) @ M_x[t].T
    inv_Q_t = np.linalg.inv(Q_t)
    a_t = M_g[t] @ M_m[t-1]
    aux_prod_t = (E_t @ M_x[t].T) @ inv_Q_t
    aux_dif_t = M_y[t] - (M_x[t] @ a_t)
    M_m[t] = a_t + (aux_prod_t @ aux_dif_t)
    M_C[t] = E_t - (aux_prod_t @ M_x[t]) @ E_t
  return M_m, M_C

@njit
def BS_t(C_now: float64[:,:], m_now: float64[:,:], g_now: float64[:,:],
         g_next: float64[:,:], beta_next: float64[:,:], Sig_now: float64[:,:],
         V: float64, W: float64[:,:]) -> (float64[:,:], float64[:,:]):
  inv_VSig_W = (1 / V) * np.linalg.inv(np.kron(Sig_now, W))
  inv_C_now = np.linalg.inv(C_now)
  inv_H_now = inv_C_now + ((g_now.T @ inv_VSig_W) @ g_now)
  H_now = np.linalg.inv(inv_H_now)
  h_now = (inv_C_now @ m_now) + ((g_next @ inv_VSig_W) @ beta_next)
  avg_now = H_now @ h_now
```

```
  return avg_now, H_now

@njit
def updt_beta(it: int64, init_m0: float64[:,:], init_C0: float64[:,:],
              M_y: float64[:,:,:], M_x: float64[:,:,:], M_g: float64[:,:,:],
              D_usage: float64[:,:], W_usage: float64[:,:], V_usage: float64,
              phi_usage: float64, Sigma_usage: float64[:,:]) -> float64[:,:,:]:

  T_obs = M_y.shape[0] - 1
  q_obs = Sigma_usage.shape[0]
  p_obs = W_usage.shape[0]

  M_m, M_C = FF(init_m0 = init_m0,
                init_C0 = init_C0,
                M_y = M_y,
                M_x = M_x,
                M_g = M_g,
                V_usage = V_usage,
                W_usage = W_usage,
                D_usage = D_usage,
                phi_usage = phi_usage,
                Sigma_usage = Sigma_usage)

  sample = np.zeros((T_obs + 1, p_obs*q_obs, 1))
  for x in range(0, T_obs + 1):
    t = T_obs - x
    if t == T_obs:
      sample[t] = chol_rmvN(avg = M_m[t],
                            cov = M_C[t],
                            n_seed = it)
    else:
      aux_BS_t = BS_t(C_now = M_C[t],
                      m_now = M_m[t],
                      g_now = M_g[t],
                      g_next = M_g[t + 1],
                      Sig_now = Sigma_usage,
                      beta_next = sample[t + 1],
                      W = W_usage,
                      V = V_usage)
```

```
      sample[t] = chol_rmvN(avg = aux_BS_t[0],
                            cov = aux_BS_t[1],
                            n_seed = it)

  return sample
```

B.2.2.4 Sampling from the full conditional distribution of ϕ

```
@njit(parallel = True)
def log_krnl_phi(par: float64, init_a_phi: float64, init_b_phi: float64,
                 M_y: float64[:,:,:], M_x: float64[:,:,:], V_usage: float64,
                 D_usage: float64[:,:], beta_usage: float64[:,:,:],
                 Sigma_usage: float64[:,:]) -> float64:
  T_obs = beta_usage.shape[0] - 1
  q_obs = Sigma_usage.shape[0]
  N_obs = D_usage.shape[1]
  log_lh = 0.0
  if par > 0:
    log_prior = (init_a_phi - 1)*log(par) - init_b_phi*par
    B_usage = nb_BR(matrix = D_usage, need_transp = True, scalar = par)
    ldet_B_usage = log(np.linalg.det(B_usage))
    Sig_B = np.kron(Sigma_usage, B_usage)
    inv_Sig_B = np.linalg.inv(Sig_B)
    for t in prange(1, T_obs + 1):
      mu_obs_t = M_x[t] @ beta_usage[t]
      y_obs_t = M_y[t]
      dif_obs_t = y_obs_t - mu_obs_t
      log_lh += ((dif_obs_t.T @ inv_Sig_B) @ dif_obs_t).item()
    return log_prior - 0.5*(1 / V_usage)*log_lh - 0.5*T_obs*q_obs*ldet_B_usage
  else:
    return -inf

@njit
def updt_phi(it: int64, init_a_phi: float64, init_b_phi: float64,
             D_usage: float64[:,:], M_y: float64[:,:,:], M_x: float64[:,:,:],
             beta_usage: float64[:,:,:], V_usage: float64, phi_usage: float64,
             Sigma_usage: float64[:,:], delta2: float64) -> float64:

  # Proposing a new value from the inverse-normal distribution
```

```
#  np.random.seed(it)
#  phi_prop = exp(np.random.normal(loc = log(phi_usage), scale = delta2))
  phi_prop = rIN(n_seed = it, mu = phi_usage, sigma2 = delta2)

  # Generating a random number between zero and one in log scale
  log_u = log_rUnif(it)

  # Computing the log of the Metropolis ratio
  log_num = +logpdf_IN(mean = phi_prop,
                       sigma2 = delta2,
                       x = phi_usage) + log_krnl_phi(par = phi_prop,
                                                     init_a_phi = init_a_phi,
                                                     init_b_phi = init_b_phi,
                                                     D_usage = D_usage,
                                                     M_y = M_y,
                                                     M_x = M_x,
                                                     beta_usage = beta_usage,
                                                     V_usage = V_usage,
                                                     Sigma_usage = Sigma_usage)
  log_den = +logpdf_IN(mean = phi_usage,
                       sigma2 = delta2,
                       x = phi_prop) + log_krnl_phi(par = phi_usage,
                                                    init_a_phi = init_a_phi,
                                                    init_b_phi = init_b_phi,
                                                    D_usage = D_usage,
                                                    M_y = M_y,
                                                    M_x = M_x,
                                                    beta_usage = beta_usage,
                                                    V_usage = V_usage,
                                                    Sigma_usage = Sigma_usage)
  log_rho = log_num - log_den
  log_alpha = min([0, log_rho])

  if isfinite(log_alpha) and log_u <= log_alpha:
    return phi_prop
  else:
    return phi_usage
```

B.2.2.5 Sampling from the full conditional distribution of D

```
@njit(parallel = True)
def log_krnl_D(par: float64[:,:], S_obs: float64[:,:], M_x: float64[:,:,:],
               M_y: float64[:,:,:], R_usage: float64[:,:],
               sigma2d_usage: float64[:,:], Sigma_usage: float64[:,:],
               V_usage: float64, beta_usage: float64[:,:,:],
               phi_usage: float64) -> float64:
  T_obs = beta_usage.shape[0] - 1
  q_obs = Sigma_usage.shape[0]
  N_obs = par.shape[1]
  log_lh = 0.0
  dif = par - S_obs
  inv_R_usage = np.linalg.inv(R_usage)
  inv_sigma2d_usage = np.linalg.inv(sigma2d_usage)
  krnl_prior = (dif.T @ inv_sigma2d_usage) @ (dif @ inv_R_usage)
  log_prior = np.trace(krnl_prior)

  B_usage = nb_BR(matrix = par, need_transp = True, scalar = phi_usage)
  Sig_B = np.kron(Sigma_usage, B_usage)
  ldet_B = log(np.linalg.det(B_usage))
  inv_Sig_B = np.linalg.inv(Sig_B)
  for t in prange(1, T_obs + 1):
    mu_obs_t = M_x[t] @ beta_usage[t]
    y_obs_t = M_y[t]
    dif_obs_t = y_obs_t - mu_obs_t
    log_lh += ((dif_obs_t.T @ inv_Sig_B) @ dif_obs_t).item()

  return -0.5*(log_prior + T_obs*q_obs*ldet_B + (1 / V_usage)*log_lh)

@njit
def updt_D(it: int64, S_obs: float64[:,:], M_x: float64[:,:,:],
           M_y: float64[:,:,:], beta_usage: float64[:,:,:], V_usage: float64,
           phi_usage: float64, sigma2d_usage: float64[:,:],
           D_usage: float64[:,:], R_usage: float64[:,:],
           Sigma_usage: float64[:,:]) -> float64[:,:]:
  N_obs = D_usage.shape[1]
  D_prev = D_usage.copy()
  w = 1
  m = +inf
```

```
lower = -inf
upper = +inf
for i in range(0, 2):
  for j in range(2, N_obs):
    np.random.seed(it + i + j)
    D_prop = D_prev.copy()

    x0 = D_prop[i,j]
    x0_D = D_prop.copy()

    # Find the log density at the initial point, if not already known.
    gx0 = log_krnl_D(par = x0_D,
                     S_obs = S_obs,
                     M_x = M_x,
                     M_y = M_y,
                     Sigma_usage = Sigma_usage,
                     beta_usage = beta_usage,
                     V_usage = V_usage,
                     phi_usage = phi_usage,
                     sigma2d_usage = sigma2d_usage,
                     R_usage = R_usage)
    # Determine the slice level, in log terms.
    logy = gx0 - np.random.exponential(scale = 1.0,
                                       size = 1).item()

    # Find the initial interval to sample from.
    u = np.random.uniform(0, w)
    L = x0 - u
    R = x0 + (w - u) # should guarantee that x0 is in [L,R], even with roundoff

    L_D = D_prop.copy()
    L_D[i,j] = L
    R_D = D_prop.copy()
    R_D[i,j] = R

    # Expand the interval until its ends are outside the slice, or until
    # the limit on steps is reached.
    if isfinite(m) is False:
      while L > lower and logy < log_krnl_D(par = L_D,
```

```
                                        S_obs = S_obs,
                                        M_x = M_x,
                                        M_y = M_y,
                                        Sigma_usage = Sigma_usage,
                                        beta_usage = beta_usage,
                                        V_usage = V_usage,
                                        phi_usage = phi_usage,
                                        sigma2d_usage = sigma2d_usage,
                                        R_usage = R_usage):
      L = L - w
      L_D[i,j] = L
    while R < upper and logy < log_krnl_D(par = R_D,
                                        S_obs = S_obs,
                                        M_x = M_x,
                                        M_y = M_y,
                                        Sigma_usage = Sigma_usage,
                                        beta_usage = beta_usage,
                                        V_usage = V_usage,
                                        phi_usage = phi_usage,
                                        sigma2d_usage = sigma2d_usage,
                                        R_usage = R_usage):
      R = R + w
      R_D[i,j] = R
  else:
    pass

  # Shrink interval to lower and upper bounds.
  if L < lower:
    L = lower
    L_D[i,j] = L
  else:
    pass
  if R > upper:
    R = upper
    R_D[i,j] = R
  else:
    pass

  # Sample from the interval, shrinking it on each rejection.
```

```
      x1_D = D_prop.copy()
      x1 = np.random.uniform(L, R)
      x1_D[i,j] = x1
      gx1 = log_krnl_D(par = x1_D,
                       S_obs = S_obs,
                       M_x = M_x,
                       M_y = M_y,
                       Sigma_usage = Sigma_usage,
                       beta_usage = beta_usage,
                       V_usage = V_usage,
                       phi_usage = phi_usage,
                       sigma2d_usage = sigma2d_usage,
                       R_usage = R_usage)

      while gx1 < logy:
        x1 = np.random.uniform(L, R)
        x1_D[i,j] = x1
        gx1 = log_krnl_D(par = x1_D,
                         S_obs = S_obs,
                         M_x = M_x,
                         M_y = M_y,
                         Sigma_usage = Sigma_usage,
                         beta_usage = beta_usage,
                         V_usage = V_usage,
                         phi_usage = phi_usage,
                         sigma2d_usage = sigma2d_usage,
                         R_usage = R_usage)
        if x1 > x0:
          R = x1
        else:
          L = x1

      D_prev = x1_D.copy()

  return x1_D
```

B.2.3 Missing data imputation

```
@njit(parallel = True)
```

```
def updt_y(it: int64, m_y: float64[:,:,:], M_x: float64[:,:,:],
           D_usage: float64[:,:], phi_usage: float64, V_usage: float64,
           beta_usage: float64[:,:,:], Sigma_usage: float64[:,:],
           num_obs: int64[:], num_mis: int64[:], M_P: float64[:,:,:],
           M_inv_P: float64[:,:,:], Id_obs: bool_[:,:]) -> float64[:,:,:]:
  T_obs = beta_usage.shape[0] - 1
  q_obs = Sigma_usage.shape[0]
  p_obs = int(beta_usage.shape[1] / q_obs)
  N_obs = D_usage.shape[1]
  B_usg = nb_BR(matrix = D_usage, need_transp = True, scalar = phi_usage)
  Sig_B_usg = np.kron(Sigma_usage, B_usg)
  vec_y = np.zeros((T_obs + 1, N_obs*q_obs, 1))
  M_y = np.zeros((T_obs + 1, N_obs, q_obs))
  for t in prange(1, T_obs + 1):
    if num_mis[t] == 0:
      vec_y[t] = m_y[t]
    elif num_mis[t] < N_obs*q_obs:
      vec_beta_t = beta_usage[t].reshape(p_obs*q_obs, 1)
      mu_t = M_P[t] @ (M_x[t] @ vec_beta_t)
      mu_o_t = mu_t[0:num_obs[t]]
      mu_m_t = mu_t[num_obs[t]:(N_obs*q_obs)]
      Delta_t = (M_P[t] @ Sig_B_usg) @ M_P[t].T
      Delta_oo_t = Delta_t[0:num_obs[t],0:num_obs[t]]
      Delta_om_t = Delta_t[0:num_obs[t],num_obs[t]:(N_obs*q_obs)]
      Delta_mo_t = Delta_om_t.T
      Delta_mm_t = Delta_t[num_obs[t]:(N_obs*q_obs),
                           num_obs[t]:(N_obs*q_obs)]
      inv_Delta_oo_t = np.linalg.inv(Delta_oo_t)
      y_obs_t = m_y[t][Id_obs[t]]
      dif_o_t = y_obs_t - mu_o_t
      aux_mult = Delta_mo_t @ inv_Delta_oo_t
      mu_cond_t = mu_m_t + (aux_mult @ dif_o_t)
      Delta_cond_t = Delta_mm_t - (aux_mult @ Delta_om_t)
      vec_m_j_t = chol_rmvN(n_seed = it + t,
                            avg = mu_cond_t,
                            cov = V_usage * Delta_cond_t)
      vec_y[t] = M_inv_P[t] @ np.vstack((y_obs_t, vec_m_j_t))
    else:
      vec_beta_t = beta_usage[t].reshape(p_obs*q_obs, 1)
```

```
      vec_y[t] = chol_rmvN(n_seed = it + t,
                           avg = M_x[t] @ vec_beta_t,
                           cov = V_usage * Sig_B_usg)
  return vec_y
```

B.2.4 First simulation study – Generating data and specifying hyperparameters

```
# Sites in the unit square
S = np.vstack([[1/5, 1/5], [4/5, 4/5], [1/5, 4/5], [4/5, 1/5],
                           [1/5, 2/5], [1/5, 3/5],
               [2/5, 1/5], [2/5, 2/5], [2/5, 3/5], [2/5, 4/5],
               [3/5, 1/5], [3/5, 2/5], [3/5, 3/5], [3/5, 4/5],
                           [4/5, 2/5], [4/5, 3/5]]).T

# The number of sites and the number of replications (or times)
N = S.shape[1]
T = 200
p, q = 2, 2
seed_v = 500

V = 0.1
a_V, b_V = 0.001, 0.001

Sigma = np.array([[1.00, 0.75],
                  [0.75, 1.00]])
a_Sigma, b_Sigma = 0.001, 0.001*np.identity(q)

phi = 0.3
a_phi, b_phi = 0.001, 0.001

### B and B_i are specified by an anisotropic structure
D = np.zeros((2, N))
D[:,0:2] = S[:,0:2]

psi = 1/(2*np.max(distance.cdist(S.T, S.T, metric = 'sqeuclidean')))
R_d = nb_BR(matrix = S,
            need_transp = True,
            squared = True,
            scalar = psi)
```

```
R_12 = R_d[0:2,0:2]
R_3N = R_d[2:18,2:18]
R_star = R_d[2:18,0:2]
S_cond = S[:,2:(N + 1)] + np.matmul(D[:,0:2] - S[:,0:2],
                                    np.linalg.inv(R_12) @ R_star.T)
R_cond = R_3N - np.matmul(R_star, np.linalg.inv(R_12) @ R_star.T)

sigma2d = np.array([[0.200, 0.000],
                    [0.000, 0.200]])

D[:,2:(N + 1)] = rMVN(n = 1,
                      avg = S_cond,
                      left = sigma2d,
                      right = R_cond,
                      n_seed = seed_v)

B = nb_BR(matrix = D, need_transp = True, scalar = phi)

M0, C0 = np.zeros((p, q)), 1.0*np.identity(p)
m0 = (M0.T).reshape((p*q, 1))
W = 1.0*np.identity(p)
Beta, beta = np.zeros((T + 1, p, q)), np.zeros((T + 1, p*q, 1))
Beta = np.zeros((T + 1, p, q))

G = np.zeros((T + 1, p, p))
for t in range(0, T + 1):
  if t == 0:
    Beta[t] = rMVN(n = 1,
                   avg = M0,
                   left = V*C0,
                   right = Sigma,
                   n_seed = seed_v + t)
  else:
    G[t] = np.identity(p)
    Beta[t] = rMVN(avg = np.matmul(G[t], Beta[t-1]),
                   left = V*W,
                   right = Sigma,
                   n = 1,
                   n_seed = seed_v + t)
```

```
    beta[t] = (Beta[t].T).reshape((p*q, 1))

# Likelihood function
Y, X = np.zeros((T + 1, N, q)), np.zeros((T + 1, N, p))
for t in range(1, T + 1):
  X[t] = np.vstack([np.ones(N),
                    uniform.rvs(loc = 0,
                                scale = 1,
                                size = N,
                                random_state = seed_v + t)]).T
  Y[t] = rMVN(n = 1,
              avg = np.matmul(X[t], Beta[t]),
              left = V*B,
              right = Sigma,
              n_seed = seed_v + t)

# x e g
x, g = np.zeros((T+1,N*q,p*q)), np.zeros((T+1,p*q,p*q))
for t in range(0, T + 1):
  g[t] = nb_kp(X = np.eye(q), Y = G[t])
  x[t] = nb_kp(X = np.eye(q), Y = X[t])

# Inserting two some missing values in each column and vectorizing
copy_Y = np.copy(Y)
for t in range(1, T + 1):
  for i in range(0, q):
    np.random.seed(t + i)
    aux = np.random.choice(N, 4, replace = False)
    copy_Y[t][aux, i] = np.array([np.nan, np.nan, np.nan, np.nan])

y = np.zeros((T + 1, N*q, 1))
for t in range(1, T + 1):
  y[t] = (copy_Y[t].T).reshape((N*q, 1))

# Computing N_obs_t, P_t and Id_obs_t
N_o = np.zeros(T + 1, dtype = np.int32)
N_m = np.zeros(T + 1, dtype = np.int32)
P = np.zeros((T + 1, N*q, N*q))
inv_P = np.zeros((T + 1, N*q, N*q))
```

```
Id_o = np.zeros((T + 1, N*q), dtype = np.bool_)
for t in range(1, T + 1):
  aux_t = Permut(y[t])
  N_o[t] = aux_t[2]
  N_m[t] = aux_t[3]
  P[t] = aux_t[4]
  inv_P[t] = np.linalg.inv(P[t])
  Id_o[t] = aux_t[7]

# R correlation matrix
def comp_R(S_obs, psi_obs):
  if isfinite(psi_obs):
    R_N = nb_BR(matrix = S_obs,
                need_transp = True,
                squared = True,
                scalar = psi_obs)
  else:
    N_obs = S_obs.shape[1]
    R_N = np.identity(N_obs)
  return R_N

# Configuration of the distortion level
psi_usage = 9.85
R_usg = comp_R(S_obs = S, psi_obs = psi_usage)
sigma2d_usg = np.cov(S)
```

B.2.5 Second simulation study – Generating data and specifying hyperparameters

```
# Sites in the unit square
S = np.vstack([[1/5, 1/5], [4/5, 4/5], [1/5, 4/5], [4/5, 1/5],
                           [1/5, 2/5], [1/5, 3/5],
               [2/5, 1/5], [2/5, 2/5], [2/5, 3/5], [2/5, 4/5],
               [3/5, 1/5], [3/5, 2/5], [3/5, 3/5], [3/5, 4/5],
                           [4/5, 2/5], [4/5, 3/5]]).T
S_i = np.array([[0.5434049, 0.2783694],
                [0.2500000, 0.3000000],
                [0.2500000, 0.7000000]]).T
S_tot = np.hstack([S, S_i])
```

```
# The number of sites and the number of replications (or times)
N, N_i, N_tot = S.shape[1], S_i.shape[1], S_tot.shape[1]
T, T_p = 100, 10
T_tot = T + T_p
p, q = 2, 2
seed_v = 500

V = 0.6
a_V, b_V = 0.001, 0.001

Sigma = np.array([[1.00, 0.85],
                  [0.85, 1.00]])
a_Sigma, b_Sigma = 0.001, 0.001*np.identity(q)

phi = 0.4
a_phi, b_phi = 0.001, 0.001

### B and B_i are specified by an anisotropic structure
A = 9*np.eye(2) - 4*np.ones((2,2))
Lambda = cholesky(A)
u = np.zeros(2)
D = np.zeros((2, N))
D_i = np.zeros((2, N_i))
for n in range(1, N_tot + 1):
  if n <= 2:
    D[:,n - 1] = S[:,n - 1]
  elif n >= 3 and n <= N:
    D[:,n - 1] = Lambda @ S[:,n - 1] + u
  else:
    D_i[:,N_tot - n - 1] = Lambda @ S_i[:,N_tot - n - 1] + u

B_aug = nb_BR(matrix = np.hstack([D, D_i]),
              need_transp = True, scalar = phi)

M0, C0 = np.zeros((p, q)), 1.0*np.identity(p)
m0 = (M0.T).reshape((p*q, 1))
W = 1.0*np.identity(p)
Beta_tot, beta_tot = np.zeros((T_tot + 1, p, q)), np.zeros((T_tot + 1, p*q, 1))
```

```
Beta, beta = np.zeros((T + 1, p, q)), np.zeros((T + 1, p*q, 1))
Beta_p, beta_p = np.zeros((T_p, p, q)), np.zeros((T_p, p*q, 1))

G = np.zeros((T_tot + 1, p, p))
for t in range(0, T_tot + 1):
  if t == 0:
    G[t] = np.identity(p)
    Beta_tot[t] = rMVN(n = 1,
                       avg = M0,
                       left = V*C0,
                       right = Sigma,
                       n_seed = seed_v + t)
    beta_tot[t] = (Beta_tot[t].T).reshape((p*q, 1))
  else:
    G[t] = np.identity(p)
    Beta_tot[t] = rMVN(avg = np.matmul(G[t], Beta_tot[t-1]),
                       left = V*W,
                       right = Sigma,
                       n = 1,
                       n_seed = seed_v + t)
    beta_tot[t] = (Beta_tot[t].T).reshape((p*q, 1))

for t in range(0, T_tot + 1):
  if t <= T:
    Beta[t] = Beta_tot[t]
    beta[t] = (Beta[t].T).reshape((p*q, 1))
  else:
    Beta_p[t - T - 1] = Beta_tot[t]
    beta_p[t - T - 1] = (Beta_p[t - T - 1].T).reshape((p*q, 1))

# X
X, X_i = np.zeros((T_tot + 1, N, p)), np.zeros((T_tot + 1, N_i, p))
for t in range(1, T_tot + 1):
  X[t] = np.vstack([np.ones(N),
                    uniform.rvs(loc = 0,
                                scale = 1,
                                size = N,
                                random_state = seed_v + t)]).T
  X_i[t] = np.vstack([np.ones(N_i),
```

```
                        uniform.rvs(loc = 0,
                                    scale = 1,
                                    size = N_i,
                                    random_state = seed_v + t)]).T

# x, x_i and g
x, g = np.zeros((T+1,N*q,p*q)), np.zeros((T+1,p*q,p*q))
x_i = np.zeros((T_tot + 1, N_i*q, p*q))
for t in range(0, T + 1):
  g[t] = nb_kp(X = np.eye(q), Y = G[t])
  x[t] = nb_kp(X = np.eye(q), Y = X[t])
  x_i[t] = nb_kp(X = np.eye(q), Y = X_i[t])

# Response matrices
Y_aug = np.zeros((T_tot + 1, N + N_i, q))
for t in range(1, T_tot + 1):
  Y_aug[t] = rMVN(n = 1,
                  avg = np.matmul(np.vstack([X[t], X_i[t]]), Beta_tot[t]),
                  left = V*B_aug,
                  right = Sigma,
                  n_seed = seed_v + t)

Y, Y_i = np.zeros((T+1,N,q)), np.zeros((T+1,N_i,q))
Y_p, Y_p_i = np.zeros((T_p,N,q)), np.zeros((T_p,N_i,q))
for t in range(1, T_tot + 1):
  if t <= T:
    Y[t] = Y_aug[t][0:N,:]
    Y_i[t] = Y_aug[t][N:N_tot,:]
  else:
    Y_p[t - T - 1] = Y_aug[t][0:N,:]
    Y_p_i[t - T - 1] = Y_aug[t][N:N_tot,:]

# Inserting two some missing values in each column and vectorizing
copy_Y = np.copy(Y)
copy_Y_i = np.copy(Y_i)
copy_Y_p = np.copy(Y_p)
copy_Y_p_i = np.copy(Y_p_i)
for t in range(1, T_tot + 1):
  for i in range(0, q):
```

```
    if t <= T:
      copy_Y[t][np.random.choice(N, 2,
                                 replace = False), i] = np.array([np.nan,
                                                                  np.nan])
      copy_Y_i[t][np.random.choice(N_i, 2,
                                   replace = False), i] = np.array([np.nan,
                                                                    np.nan])
    else:
      copy_Y_p[t-T-1][np.random.choice(N, 2, replace = False),
                                       i] = np.array([np.nan,
                                                      np.nan])
      copy_Y_p_i[t-T-1][np.random.choice(N_i, 2, replace = False),
                                         i] = np.array([np.nan,
                                                        np.nan])


y = np.zeros((T + 1, N*q, 1))
y_i = np.zeros((T + 1, N_i*q, 1))
y_p = np.zeros((T_p, N*q, 1))
y_p_i = np.zeros((T_p, N_i*q, 1))
for t in range(1, T_tot + 1):
  if t <= T:
    y[t] = (copy_Y[t].T).reshape((N*q, 1))
    y_i[t] = (copy_Y_i[t].T).reshape((N_i*q, 1))
  else:
    y_p[t-T-1] = (copy_Y_p[t-T-1].T).reshape((N*q, 1))
    y_p_i[t-T-1] = (copy_Y_p_i[t-T-1].T).reshape((N_i*q, 1))

# Computing N_obs_t, P_t and Id_obs_t
N_o = np.zeros(T + 1, dtype = np.int32)
N_m = np.zeros(T + 1, dtype = np.int32)
P = np.zeros((T + 1, N*q, N*q))
inv_P = np.zeros((T + 1, N*q, N*q))
Id_o = np.zeros((T + 1, N*q), dtype = np.bool_)
for t in range(1, T + 1):
  aux_t = Permut(y[t])
  N_o[t] = aux_t[2]
  N_m[t] = aux_t[3]
  P[t] = aux_t[4]
```

```
  inv_P[t] = np.linalg.inv(P[t])
  Id_o[t] = aux_t[7]

# R correlation matrix
def comp_R(S_obs, psi_obs):
  if isfinite(psi_obs):
    R_N = nb_BR(matrix = S_obs,
                need_transp = True,
                squared = True,
                scalar = psi_obs)
  else:
    N_obs = S_obs.shape[1]
    R_N = np.identity(N_obs)
  return R_N

# Configuration of the distortion level
psi_usage = 9.85
R_usg = comp_R(S_obs = S, psi_obs = psi_usage)
sigma2d_usg = np.cov(S)
```

B.2.6 Hybrid MCMC algorithm

```
############################ Creating objects to store the MCMC samples
burn_in = 2000
num_iter = 10000 + burn_in
thin = 10
smp_size = int((num_iter - burn_in)/thin)

e_y = [np.zeros((T + 1, N, q)) for k in range(0, smp_size)]
e_phi = [0 for k in range(0, smp_size)]
e_V = [0 for k in range(0, smp_size)]
e_beta = [np.zeros((T + 1, p*q, 1)) for k in range(0, smp_size)]
e_Sigma = [np.zeros((q, q)) for k in range(0, smp_size)]
e_D = [np.zeros((2, N)) for k in range(0, smp_size)]

############################ Initial values for MCMC estimation
phi_prev = 2.0
V_prev = 2.0
Sigma_prev = 2.0*np.identity(q)
```

```
beta_prev = np.zeros((T + 1, p*q, 1))
D_prev = np.copy(S)
delta_phi = 21.0

############################ Metropolis-within-Gibbs algorithm
# ind will vary between 0 and smp_size - 1 (i.e. k = 1, ..., K)
ind = 0
acc_phi, acc_D = 0, np.zeros((2, N))

# Algorithm and processing time in seconds
start = timer()
for j in range(1, num_iter + 1):
  y_curr = updt_y(it = seed_v + j,
                  m_y = y,
                  M_x = x,
                  D_usage = D_prev,
                  phi_usage = phi_prev,
                  V_usage = V_prev,
                  beta_usage = beta_prev,
                  Sigma_usage = Sigma_prev,
                  M_P = P,
                  M_inv_P = inv_P,
                  Id_obs = Id_o,
                  num_obs = N_o,
                  num_mis = N_m)
  beta_curr = updt_beta(it = seed_v + j,
                        init_m0 = m0,
                        init_C0 = C0,
                        M_y = y_curr,
                        M_x = x,
                        M_g = g,
                        D_usage = D_prev,
                        W_usage = W,
                        phi_usage = phi_prev,
                        V_usage = V_prev,
                        Sigma_usage = Sigma_prev)
  V_curr = updt_V(it = seed_v + j,
                  init_a_V = a_V,
                  init_b_V = b_V,
```

```
                  init_m0 = m0,
                  init_C0 = C0,
                  M_y = y_curr,
                  M_x = x,
                  M_g = g,
                  D_usage = D_prev,
                  beta_usage = beta_curr,
                  phi_usage = phi_prev,
                  Sigma_usage = Sigma_prev,
                  W_usage = W)
phi_curr = updt_phi(it = seed_v + j,
                    init_a_phi = a_phi,
                    init_b_phi = b_phi,
                    D_usage = D_prev,
                    M_y = y_curr,
                    M_x = x,
                    beta_usage = beta_curr,
                    V_usage = V_curr,
                    phi_usage = phi_prev,
                    Sigma_usage = Sigma_prev,
                    delta2 = delta_phi)
Sigma_curr = updt_Sigma(it = seed_v + j,
                        init_a_Sigma = a_Sigma,
                        init_b_Sigma = b_Sigma,
                        init_M0 = M0,
                        init_C0 = C0,
                        M_y = y_curr,
                        M_X = X,
                        M_G = G,
                        beta_usage = beta_curr,
                        W_usage = W,
                        phi_usage = phi_curr,
                        V_usage = V_curr,
                        D_usage = D_prev)
D_curr = updt_D(it = seed_v + j,
                S_obs = S,
                M_x = x,
                M_y = y_curr,
                Sigma_usage = Sigma_curr,
```

```
                    beta_usage = beta_curr,
                    V_usage = V_curr,
                    phi_usage = phi_curr,
                    sigma2d_usage = sigma2d_usg,
                    R_usage = R_usg,
                    D_usage = D_prev)

  if abs(phi_curr - phi_prev) != 0:
    acc_phi += 1
  else:
    pass

  for r in range(0, 2):
    for n in range(0, N):
      if abs(D_curr[r,n] - D_prev[r,n]) != 0:
        acc_D[r,n] += 1
      else:
        pass

  if j > burn_in and j % thin == 0:
    e_y[ind] = y_curr
    e_V[ind] = V_curr
    e_beta[ind] = beta_curr
    e_phi[ind] = phi_curr
    e_Sigma[ind] = Sigma_curr
    e_D[ind] = D_curr
    ind += 1
  else:
    pass

  y_prev = y_curr
  V_prev = V_curr
  phi_prev = phi_curr
  beta_prev = beta_curr
  Sigma_prev = Sigma_curr
  D_prev = D_curr

  print(j)
end = timer()
```

```
print(end - start)
```

To use the isotropic model $\mathcal{M}_{\text{I}}$ with $B_{n,n'} = \exp\{-\phi\|\underset{\sim}{s}_n - \underset{\sim}{s}_{n'}\|\}$, just let `D_curr = np.copy(S)`.

B.2.7 Forecasting and interpolation

B.2.7.1 Forecasting

```
e_Beta = [np.zeros((T + 1, p, q)) for k in range(0, smp_size)]
aux_kron = nb_kp(((np.eye(q).T).reshape((q*q, 1))).T, np.eye(p))
for k in range(0, int(smp_size)):
  for t in range(1, T + 1):
    e_Beta[k][t] = np.matmul(aux_kron, nb_kp(np.eye(q), e_beta[k][t]))

e_Y = [np.zeros((T + 1, N, q)) for k in range(0, smp_size)]
for k in range(0, int(smp_size)):
  for t in range(1, T + 1):
    e_Y[k][t] = (e_y[k][t].reshape(q, N)).T

e_Beta_p = [np.zeros((T_p, p, q)) for k in range(0, smp_size)]
for k in range(0, int(smp_size)):
  for t in range(1, T_p + 1):
    aux_t = G[T + t]
    cov_t = W.copy()
    if t > 1:
      for t2 in reversed(range(2, t + 1)):
        aux_t = np.matmul(aux_t, G[T + t2])
        cov_t += np.matmul(aux_t.T, W @ aux_t)
    else:
      pass
    e_Beta_p[k][t - 1] = rMVN(n = 1,
                              avg = np.matmul(aux_t, e_Beta[k][T]),
                              left = e_V[k] * cov_t,
                              right = e_Sigma[k],
                              n_seed = seed_v + k*t)

e_Y_p = [np.zeros((T_p, N, q)) for k in range(0, smp_size)]
for k in range(0, int(smp_size)):
  for t in range(1, T_p + 1):
    e_Y_p[k][t - 1] = rMVN(n = 1,
```

```
                        avg = np.matmul(X[T + t], e_Beta_p[k][t - 1]),
                        left = e_V[k] * nb_BR(matrix = e_D[k],
                                              need_transp = True,
                                              scalar = e_phi[k]),
                        right = e_Sigma[k],
                        n_seed = seed_v + k*t)
```

B.2.7.2 Interpolation

Below we present an implementation for Algorithms 13 and 14.

```
if isfinite(psi_usage) is False:
  R_tot = np.identity(N_tot)
else:
  R_tot = nb_BR(matrix = S_tot,
                need_transp = True,
                squared = True,
                scalar = psi_usage)
R_gu = R_tot[0:N,N:N_tot]
R_1N = R_tot[0:N,0:N]
inv_R_1N = np.linalg.inv(R_1N)
R_star = R_tot[N:N_tot,N:N_tot]
D_cov = R_star - np.matmul(R_gu.T, inv_R_1N @ R_gu)

e_D_i = [np.zeros((2, N_i)) for k in range(0, smp_size)]
for k in range(0, int(smp_size)):
  D_mean_k = S_i + np.matmul(e_D[k] - S, inv_R_1N @ R_gu)
  e_D_i[k] = rMVN(n = 1,
                  avg = D_mean_k,
                  left = sigma2d_usg,
                  right = D_cov,
                  n_seed = k + seed_v)

e_Y_i = [np.zeros((T + 1, N_i, q)) for j in range(0, smp_size)]
e_Y_p_i = [np.zeros((T_p, N_i, q)) for j in range(0, smp_size)]
for k in range(0, int(smp_size)):
  B_k = nb_BR(matrix = e_D[k],
              need_transp = True,
              scalar = e_phi[k])
  inv_B_k = np.linalg.inv(B_k)
```

```
  dist_oi_k = np.zeros((N, N_i))
  for n in range(0, N):
    for m in range(0, N_i):
      dist_oi_k[n, m] = dist(e_D[k][:,n], e_D_i[k][:,m])
  B_oi_k = np.exp(-e_phi[k] * dist_oi_k)
  B_io_k = B_oi_k.T
  B_i_k = nb_BR(matrix = e_D_i[k],
                need_transp = True,
                scalar = e_phi[k])
  aux_B_k = np.matmul(B_io_k, inv_B_k)
  cov_B_k = B_i_k - np.matmul(aux_B_k, B_oi_k)
  for t in range(1, T + 1):
    avg1 = (X_i[t] @ e_Beta[k][t])
    avg2 = (aux_B_k @ (e_Y[k][t]-(X[t] @ e_Beta[k][t])))
    avg_t_k = avg1 + avg2
    e_Y_i[k][t] = rMVN(n = 1,
                       avg = avg_t_k,
                       left = e_V[k] * cov_B_k,
                       right = e_Sigma[k],
                       n_seed = seed_v + k*t)
  for t in range(T + 1, T_tot + 1):
    avg1 = (X_i[t] @ e_Beta_p[k][t - T - 1])
    avg2 = aux_B_k @ (e_Y_p[k][t - T - 1]-(X[t] @ e_Beta_p[k][t - T - 1]))
    avg_sum = avg1 + avg2
    e_Y_p_i[k][t - T - 1] = rMVN(n = 1,
                                 avg = avg_sum,
                                 left = e_V[k] * cov_B_k,
                                 right = e_Sigma[k],
                                 n_seed = seed_v + k*t)
```

B.2.8 Additional codes (PMSE and DIC)

```
def PMSE(M_Y_i: np.ndarray, e_Y_i = e_Y_i) -> np.float64:
  K = np.shape(e_Y_i)[0]
  T = np.shape(e_Y_i)[1] - 1
  N_i = np.shape(e_Y_i)[2]
  q = np.shape(e_Y_i)[3]
  aux_tot, square_dif_sum = 0, 0
  for n in np.arange(0, N_i):
```

```
    for i in np.arange(0, q):
      for t in np.arange(1, T + 1):
        if np.isnan(Y_i[t][n,i]):
          pass
        else:
          aux_tot += 1
          aux_sum = 0
          for k in np.arange(0, K):
            aux_sum += e_Y_i[k][t][n,i]
          aux_mean = aux_sum / K
          square_dif_sum += pow(M_Y_i[t][n,i] - aux_mean, 2)
  square_dif_mean = square_dif_sum / aux_tot
  return square_dif_mean
PMSE_value = PMSE(M_Y_i = Y_i)

def DIC(M_y: np.ndarray, M_x: np.ndarray, e_beta = e_beta, e_V = e_V,
        e_phi = e_phi, e_D = e_D, e_Sigma = e_Sigma, e_Beta = e_Beta,
        num_obs = N_o, M_P = P, Id_obs = Id_o, e_y = e_y) -> np.float64:
  K, N = np.shape(e_Beta)[0], np.shape(e_D)[2]
  T = np.shape(e_Beta)[1] - 1
  p, q = np.shape(e_Beta)[2], np.shape(e_Beta)[3]
  L = np.zeros((K, T))
  for k in range(0, K):
    cov_k = e_V[k] * nb_kp(e_Sigma[k],
                           nb_BR(matrix = e_D[k],
                                 need_transp = True,
                                 scalar = e_phi[k]))
    for t in range(1, T + 1):
      if num_obs[t] == N*q:
        mu_t = np.matmul(M_x[t], e_beta[k][t]).reshape(N*q)
        L[k,t-1] = multivariate_normal.logpdf(x = M_y[t].reshape(N*q),
                                              mean = mu_t,
                                              cov = cov_k)
      elif num_obs[t] > 0 and num_obs[t] < N*q:
        mu_t = M_P[t] @ (M_x[t] @ e_beta[k][t])
        mu_o_t = mu_t[0:num_obs[t]]
        mu_m_t = mu_t[num_obs[t]:(N*q)]
        Delta_t = (M_P[t] @ cov_k) @ M_P[t].T
        Delta_oo_t = Delta_t[0:num_obs[t],0:num_obs[t]]
```

```
        Delta_om_t = Delta_t[0:num_obs[t],num_obs[t]:(N*q)]
        Delta_mo_t = Delta_om_t.T
        Delta_mm_t = Delta_t[num_obs[t]:(N*q),
                             num_obs[t]:(N*q)]
        inv_Delta_mm_t = np.linalg.inv(Delta_mm_t)
        y_obs_t = (M_y[t][Id_obs[t]]).reshape(num_obs[t])
        y_mis_t = e_y[k][t][np.logical_not(Id_obs[t])]
        dif_m_t = y_mis_t - mu_m_t
        aux_mult = Delta_om_t @ inv_Delta_mm_t
        mu_cond_t = (mu_o_t + (aux_mult @ dif_m_t)).reshape(num_obs[t])
        Delta_cond_t = Delta_oo_t - (aux_mult @ Delta_mo_t)
        L[k,t-1] = multivariate_normal.logpdf(x = y_obs_t,
                                              mean = mu_cond_t,
                                              cov = Delta_cond_t)
    else:
      pass
beta_sum = np.zeros((T + 1, p*q, 1))
V_sum = 0
phi_sum = 0
D_sum = np.zeros((2, N))
Sigma_sum = np.zeros((q, q))
y_sum = np.zeros((T + 1, N*q, 1))
for k in range(0, K):
  V_sum += e_V[k]
  phi_sum += e_phi[k]
  Sigma_sum += e_Sigma[k]
  D_sum += e_D[k]
  for t in range(0, T + 1):
    beta_sum[t] += e_beta[k][t]
    y_sum[t] += e_y[k][t]
y_bar = y_sum / K
beta_bar = beta_sum / K
V_bar = V_sum / K
phi_bar = phi_sum / K
D_bar = D_sum / K
Sigma_bar = Sigma_sum / K
cov_bar = V_bar * nb_kp(Sigma_bar,
                        nb_BR(matrix = D_bar,
                              need_transp = True,
```

```
                                    scalar = phi_bar))
  F = np.zeros(T)
  for t in range(1, T + 1):
    if num_obs[t] == N*q:
      mu_bar = (np.matmul(M_x[t], beta_bar[t])).reshape(N*q)
      F[t-1] = multivariate_normal.logpdf(x = M_y[t].reshape(N*q),
                                          mean = mu_bar,
                                          cov = cov_bar)
    elif num_obs[t] > 0 and num_obs[t] < N*q:
      mu_t = M_P[t] @ (M_x[t] @ beta_bar[t])
      mu_o_t = mu_t[0:num_obs[t]]
      mu_m_t = mu_t[num_obs[t]:(N*q)]
      Delta_t = (M_P[t] @ cov_bar) @ M_P[t].T
      Delta_oo_t = Delta_t[0:num_obs[t],0:num_obs[t]]
      Delta_om_t = Delta_t[0:num_obs[t],num_obs[t]:(N*q)]
      Delta_mo_t = Delta_om_t.T
      Delta_mm_t = Delta_t[num_obs[t]:(N*q),
                           num_obs[t]:(N*q)]
      inv_Delta_mm_t = np.linalg.inv(Delta_mm_t)
      y_obs_t = (M_y[t][Id_obs[t]]).reshape(num_obs[t])
      y_mis_t = y_bar[t][np.logical_not(Id_obs[t])]
      dif_m_t = y_mis_t - mu_m_t
      aux_mult = Delta_om_t @ inv_Delta_mm_t
      mu_cond_t = (mu_o_t + (aux_mult @ dif_m_t)).reshape(num_obs[t])
      Delta_cond_t = Delta_oo_t - (aux_mult @ Delta_mo_t)
      F[t-1] = multivariate_normal.logpdf(x = y_obs_t,
                                          mean = mu_cond_t,
                                          cov = Delta_cond_t)
    else:
      pass
  return -(4/K)*np.sum(L) + 2*np.sum(F)
DIC_value = DIC(M_y = y, M_x = x)
```

Appendix C

Supplementary material

We export the observed values and the MCMC outputs that we obtained after running a hybrid MCMC algorithm (Appendix B.1.5 or Appendix B.2.6) to R (R Core Team, 2023). We generate some figures using the following R packages:

- `Morpho` (Schlager, 2017), to plot the spatial deformations.
- `ggplot2` (Wickham, 2016), to plot maps, histograms, line charts, boxplots, trace plots, etc.
- `ggpubr` (Kassambara, 2020), to plot sub-figures in a figure with a common legend.
- `ggmap` (Kahle and Wickham, 2013), to plot maps highlighting the sites or monitoring stations.
- `ggrepel` (Slowikowski, 2021), to label the sites or monitoring stations.
- `coda` (Plummer et al., 2006), to represent the autocorrelation of the sampled values via the MCMC algorithm for some parameter.

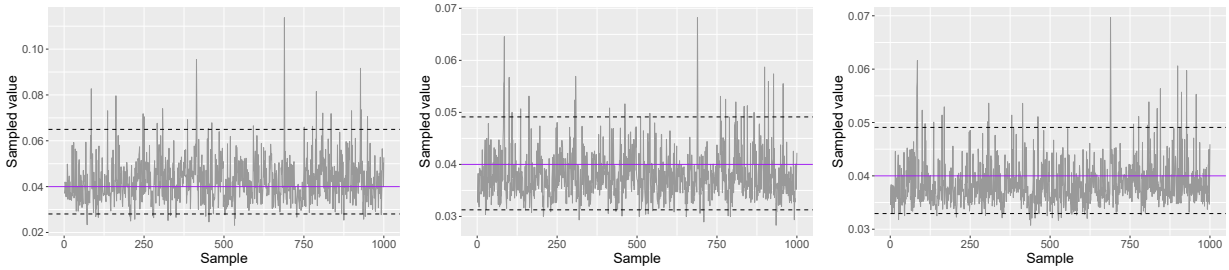
C.1 Supplemental material of Chapter 2

C.1.1 First simulation study

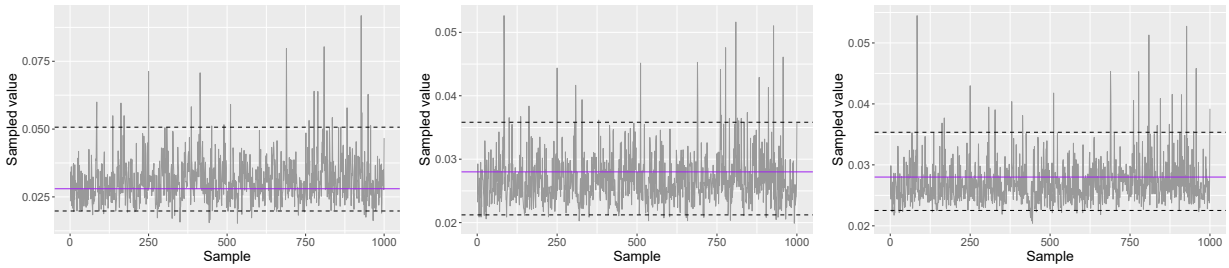
Here we present some trace plots of the MCMC samples of the parameters involved in the first simulation study of Chapter 2:

- Figure C.1 shows the trace plots of the parameters $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ for the sample sizes $T \in \{10, 100, 1000\}$.

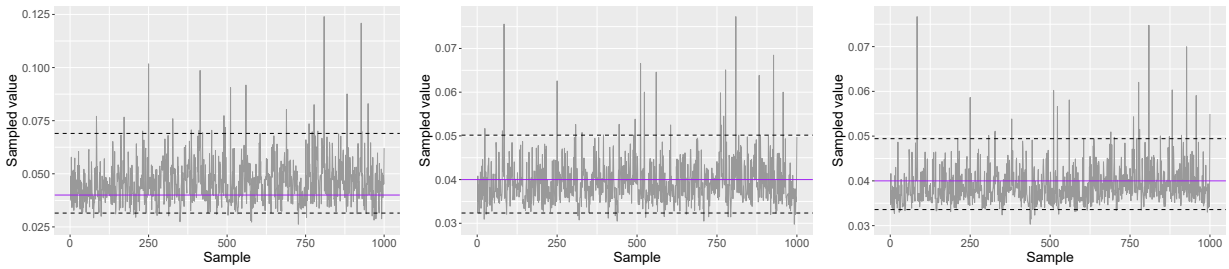
- Figures C.2, C.3, C.4, C.5, C.6 and C.7 show the trace plots of the parameters $D_{m,n}$ for the sample sizes $T \in \{10, 100, 1000\}$.



(a) $\phi \cdot V \cdot \Sigma_{1,1}$ with $T = 10$. (b) $\phi \cdot V \cdot \Sigma_{1,1}$ with $T = 100$. (c) $\phi \cdot V \cdot \Sigma_{1,1}$ with $T = 1000$.

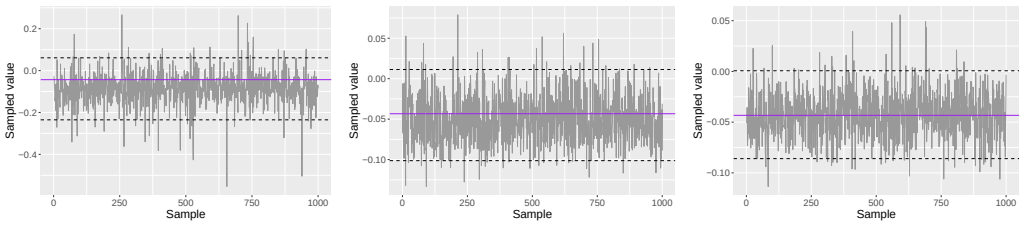


(d) $\phi \cdot V \cdot \Sigma_{1,2}$ with $T = 10$. (e) $\phi \cdot V \cdot \Sigma_{1,2}$ with $T = 100$. (f) $\phi \cdot V \cdot \Sigma_{1,2}$ with $T = 1000$.

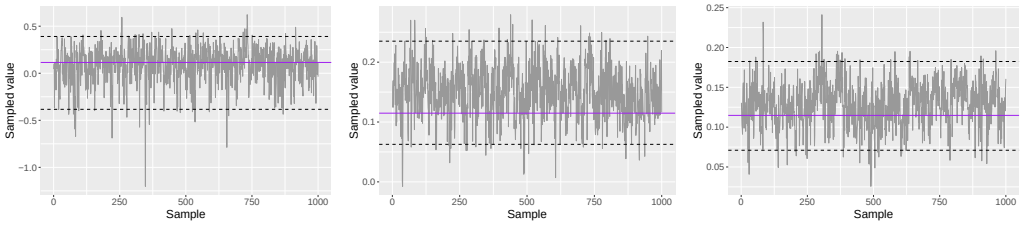


(g) $\phi \cdot V \cdot \Sigma_{2,2}$ with $T = 10$. (h) $\phi \cdot V \cdot \Sigma_{2,2}$ with $T = 100$. (i) $\phi \cdot V \cdot \Sigma_{2,2}$ with $T = 1000$.

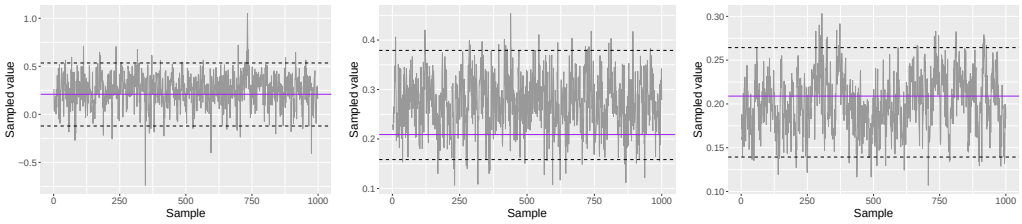
Figure C.1: Trace plots of the posterior distribution of the parameters $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



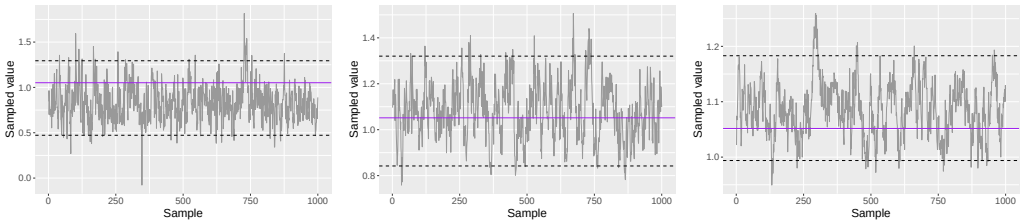
(a) $D_{1,3}$ with $T = 10$. (b) $D_{1,3}$ with $T = 100$. (c) $D_{1,3}$ with $T = 1000$.



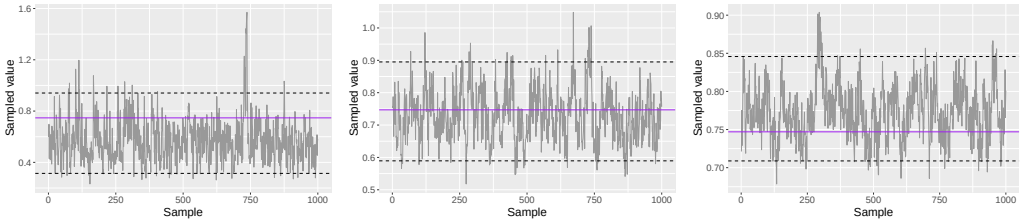
(d) $D_{1,4}$ with $T = 10$. (e) $D_{1,4}$ with $T = 100$. (f) $D_{1,4}$ with $T = 1000$.



(g) $D_{1,5}$ with $T = 10$. (h) $D_{1,5}$ with $T = 100$. (i) $D_{1,5}$ with $T = 1000$.

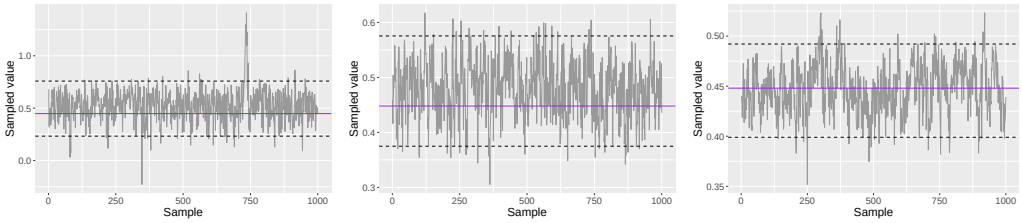


(j) $D_{1,6}$ with $T = 10$. (k) $D_{1,6}$ with $T = 100$. (l) $D_{1,6}$ with $T = 1000$.

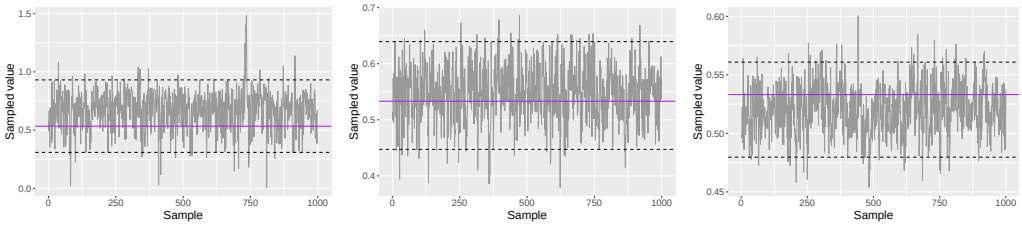


(m) $D_{1,7}$ with $T = 10$. (n) $D_{1,7}$ with $T = 100$. (o) $D_{1,7}$ with $T = 1000$.

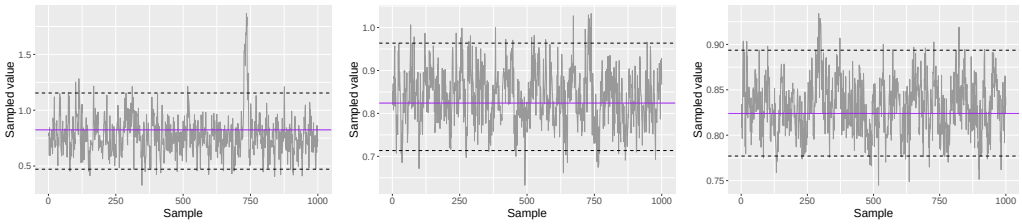
Figure C.2: Trace plots of the posterior distribution of $D_{1,n}$, $3 \leq n \leq 7$, for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



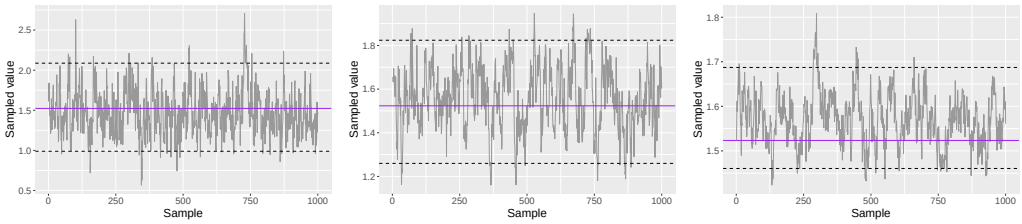
(a) $D_{1,8}$ with $T = 10$. (b) $D_{1,8}$ with $T = 100$. (c) $D_{1,8}$ with $T = 1000$.



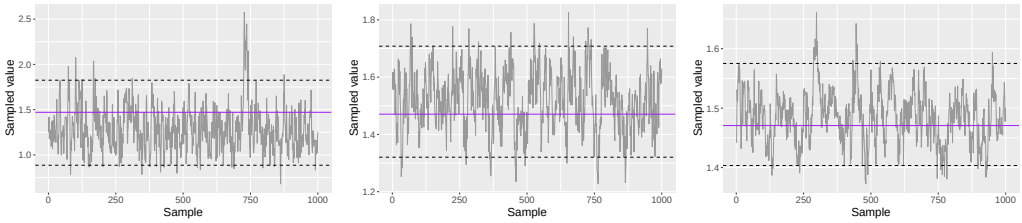
(d) $D_{1,9}$ with $T = 10$. (e) $D_{1,9}$ with $T = 100$. (f) $D_{1,9}$ with $T = 1000$.



(g) $D_{1,10}$ with $T = 10$. (h) $D_{1,10}$ with $T = 100$. (i) $D_{1,10}$ with $T = 1000$.

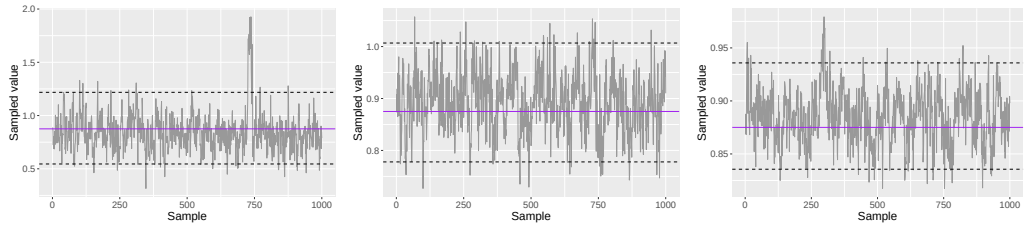


(j) $D_{1,11}$ with $T = 10$. (k) $D_{1,11}$ with $T = 100$. (l) $D_{1,11}$ with $T = 1000$.

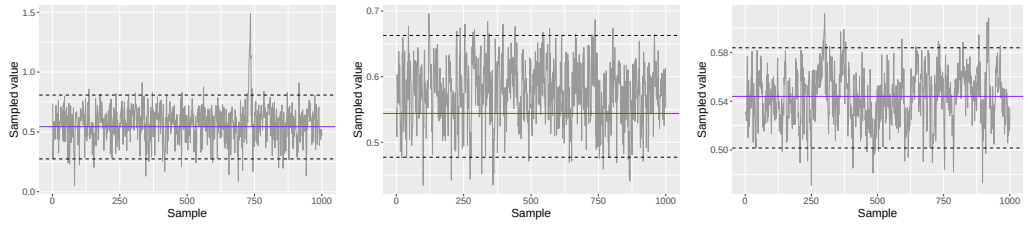


(m) $D_{1,12}$ with $T = 10$. (n) $D_{1,12}$ with $T = 100$. (o) $D_{1,12}$ with $T = 1000$.

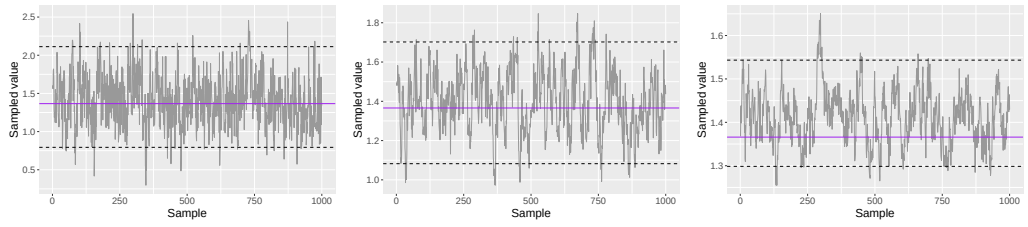
Figure C.3: Trace plots of the posterior distribution of $D_{1,n}$, $8 \leq n \leq 12$, for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



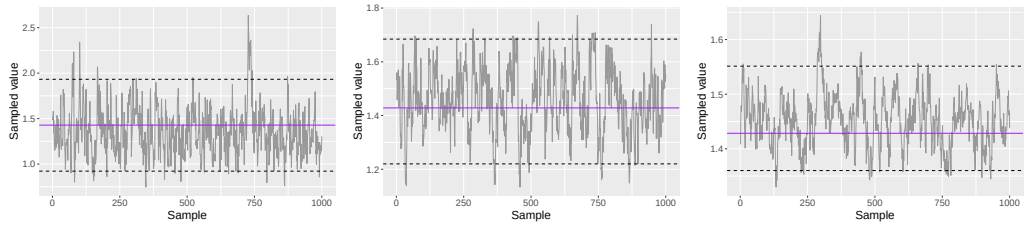
(a) $D_{1,13}$ with $T = 10$. (b) $D_{1,13}$ with $T = 100$. (c) $D_{1,13}$ with $T = 1000$.



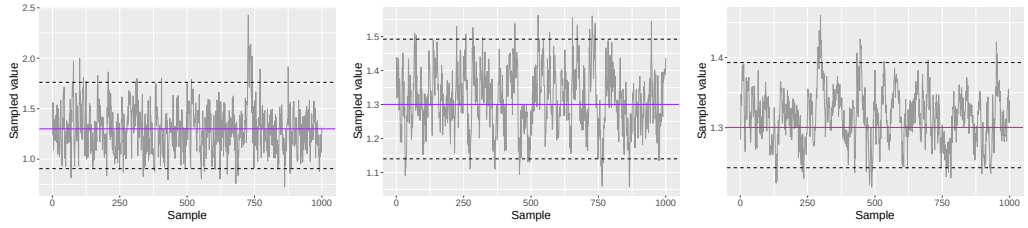
(d) $D_{1,14}$ with $T = 10$. (e) $D_{1,14}$ with $T = 100$. (f) $D_{1,14}$ with $T = 1000$.



(g) $D_{1,15}$ with $T = 10$. (h) $D_{1,15}$ with $T = 100$. (i) $D_{1,15}$ with $T = 1000$.

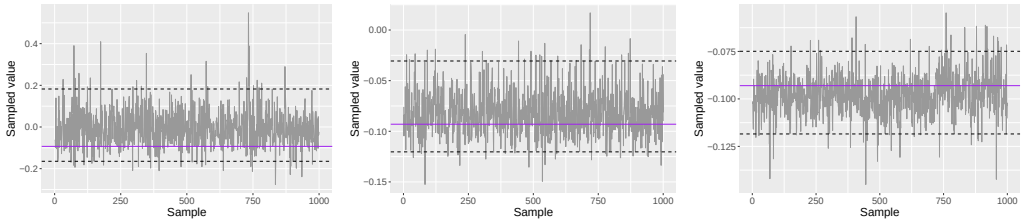


(j) $D_{1,16}$ with $T = 10$. (k) $D_{1,16}$ with $T = 100$. (l) $D_{1,16}$ with $T = 1000$.

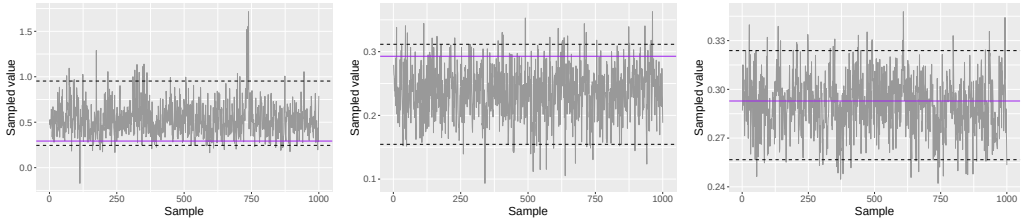


(m) $D_{1,17}$ with $T = 10$. (n) $D_{1,17}$ with $T = 100$. (o) $D_{1,17}$ with $T = 1000$.

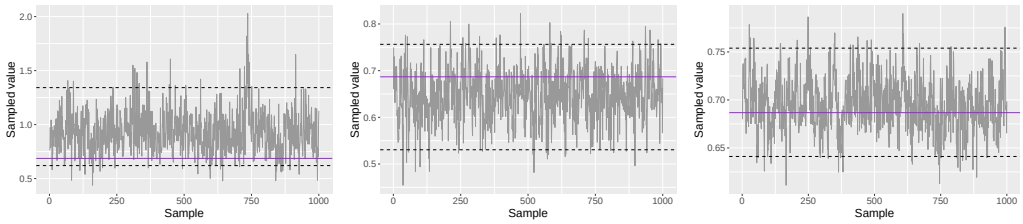
Figure C.4: Trace plots of the posterior distribution of $D_{1,n}$, $13 \leq n \leq 17$, for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



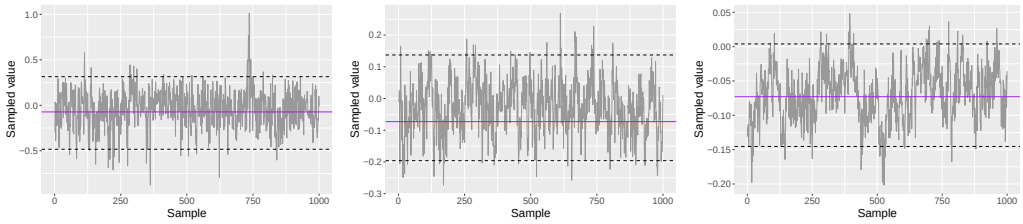
(a) $D_{2,3}$ with $T = 10$. (b) $D_{2,3}$ with $T = 100$. (c) $D_{2,3}$ with $T = 1000$.



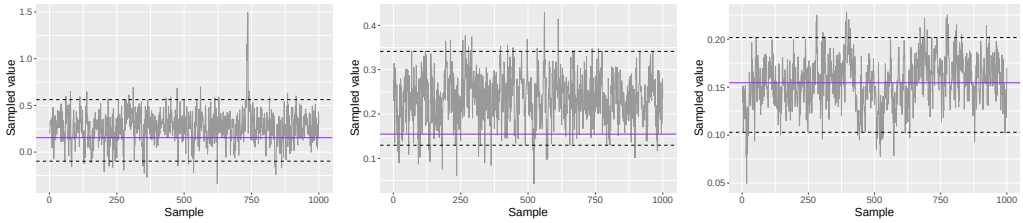
(d) $D_{2,4}$ with $T = 10$. (e) $D_{2,4}$ with $T = 100$. (f) $D_{2,4}$ with $T = 1000$.



(g) $D_{2,5}$ with $T = 10$. (h) $D_{2,5}$ with $T = 100$. (i) $D_{2,5}$ with $T = 1000$.

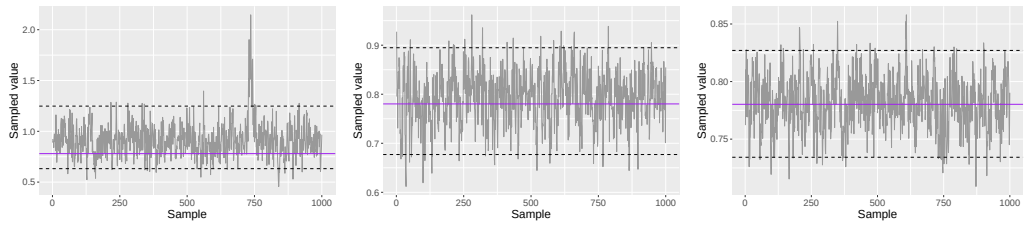


(j) $D_{2,6}$ with $T = 10$. (k) $D_{2,6}$ with $T = 100$. (l) $D_{2,6}$ with $T = 1000$.

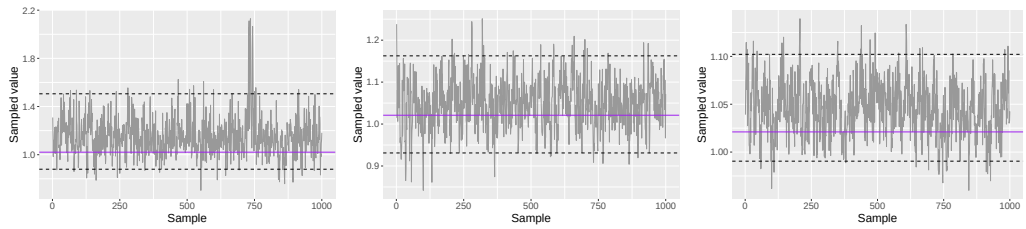


(m) $D_{2,7}$ with $T = 10$. (n) $D_{2,7}$ with $T = 100$. (o) $D_{2,7}$ with $T = 1000$.

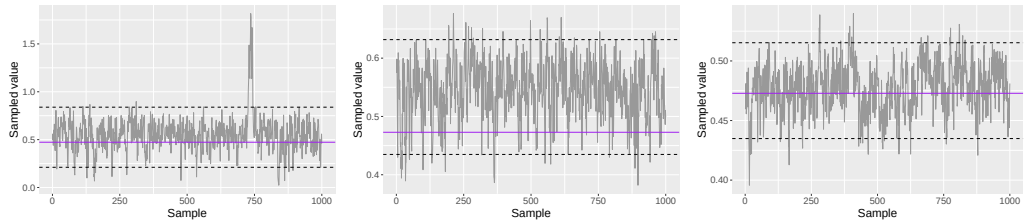
Figure C.5: Trace plots of the posterior distribution of $D_{2,n}$, $3 \leq n \leq 7$, for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



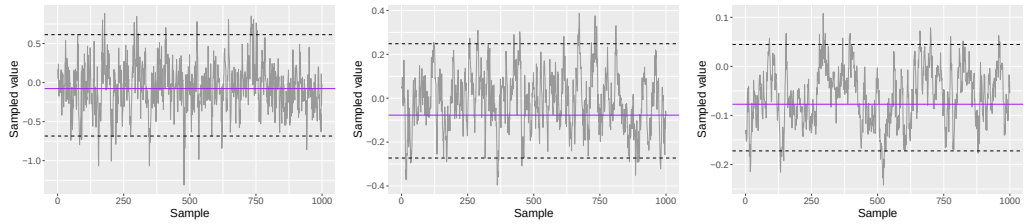
(a) $D_{2,8}$ with $T = 10$. (b) $D_{2,8}$ with $T = 100$. (c) $D_{2,8}$ with $T = 1000$.



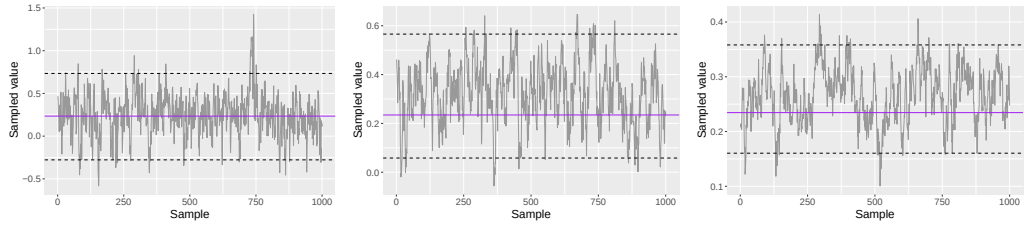
(d) $D_{2,9}$ with $T = 10$. (e) $D_{2,9}$ with $T = 100$. (f) $D_{2,9}$ with $T = 1000$.



(g) $D_{2,10}$ with $T = 10$. (h) $D_{2,10}$ with $T = 100$. (i) $D_{2,10}$ with $T = 1000$.

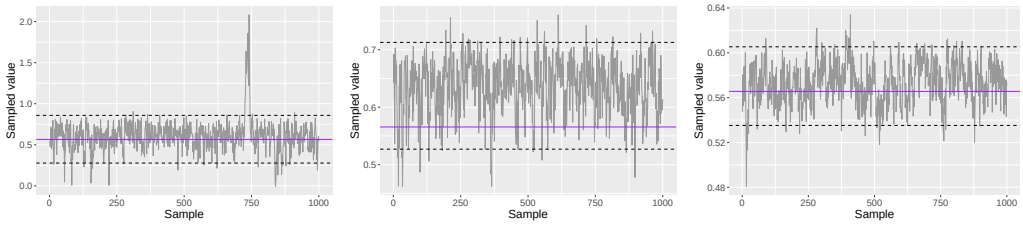


(j) $D_{2,11}$ with $T = 10$. (k) $D_{2,11}$ with $T = 100$. (l) $D_{2,11}$ with $T = 1000$.

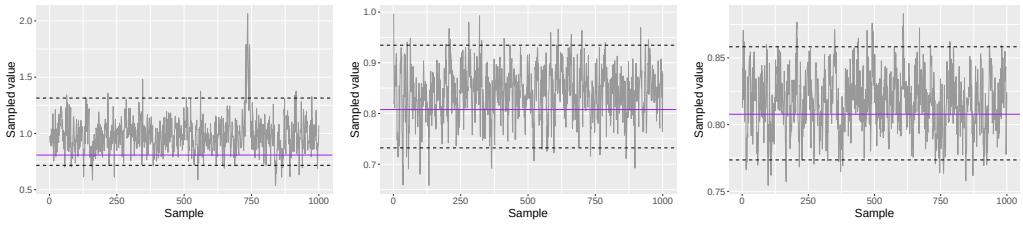


(m) $D_{2,12}$ with $T = 10$. (n) $D_{2,12}$ with $T = 100$. (o) $D_{2,12}$ with $T = 1000$.

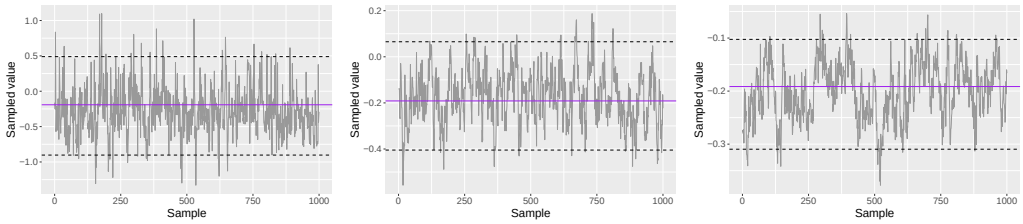
Figure C.6: Trace plots of the posterior distribution of $D_{2,n}$, $8 \leq n \leq 12\}$, for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



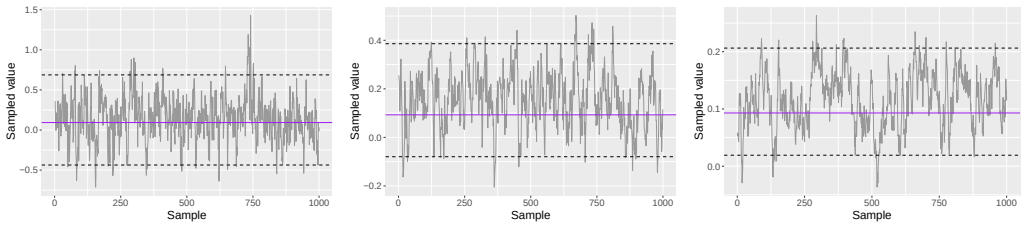
(a) $D_{2,13}$ with $T = 10$. (b) $D_{2,13}$ with $T = 100$. (c) $D_{2,13}$ with $T = 1000$.



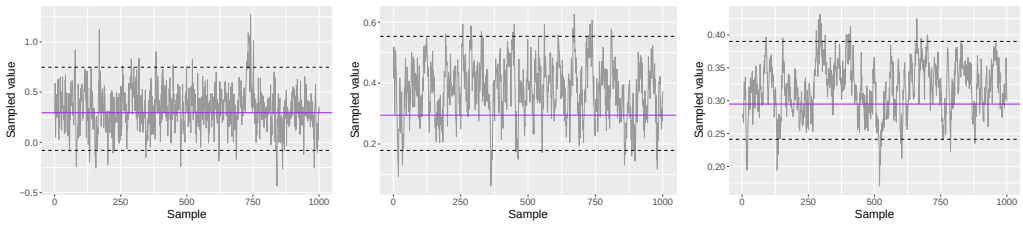
(d) $D_{2,14}$ with $T = 10$. (e) $D_{2,14}$ with $T = 100$. (f) $D_{2,14}$ with $T = 1000$.



(g) $D_{2,15}$ with $T = 10$. (h) $D_{2,15}$ with $T = 100$. (i) $D_{2,15}$ with $T = 1000$.



(j) $D_{2,16}$ with $T = 10$. (k) $D_{2,16}$ with $T = 100$. (l) $D_{2,16}$ with $T = 1000$.



(m) $D_{2,17}$ with $T = 10$. (n) $D_{2,17}$ with $T = 100$. (o) $D_{2,17}$ with $T = 1000$.

Figure C.7: Trace plots of the posterior distribution of $D_{2,n}$, $13 \leq n \leq 17$, for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.

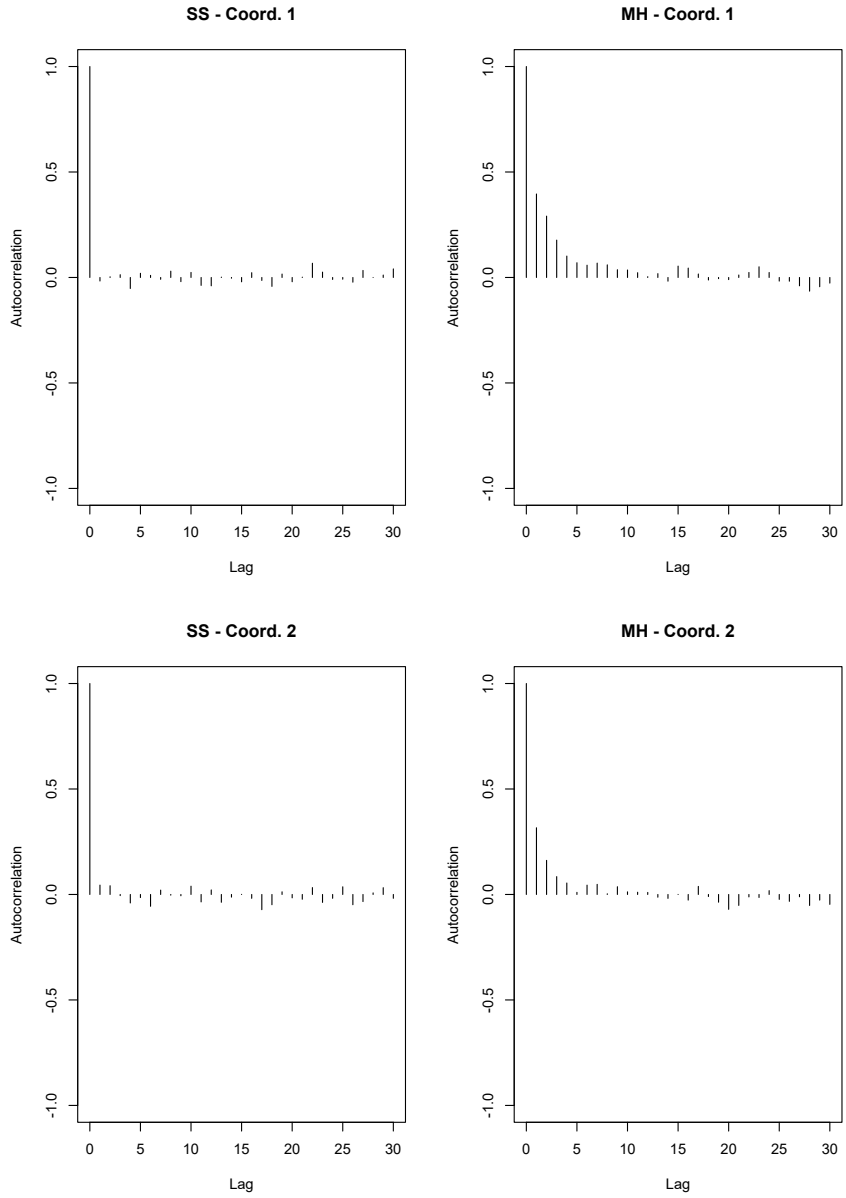


Figure C.8: Autocorrelation of sampled values for the pair of parameters $D_{1,3}$ (1^{st} coordinate) and $D_{2,3}$ (2^{nd} coordinate), obtained through the slice sampling (SS) and Metropolis-Hastings (MH) algorithms with $T = 100$, resulting from the first simulation study in Chapter 2.

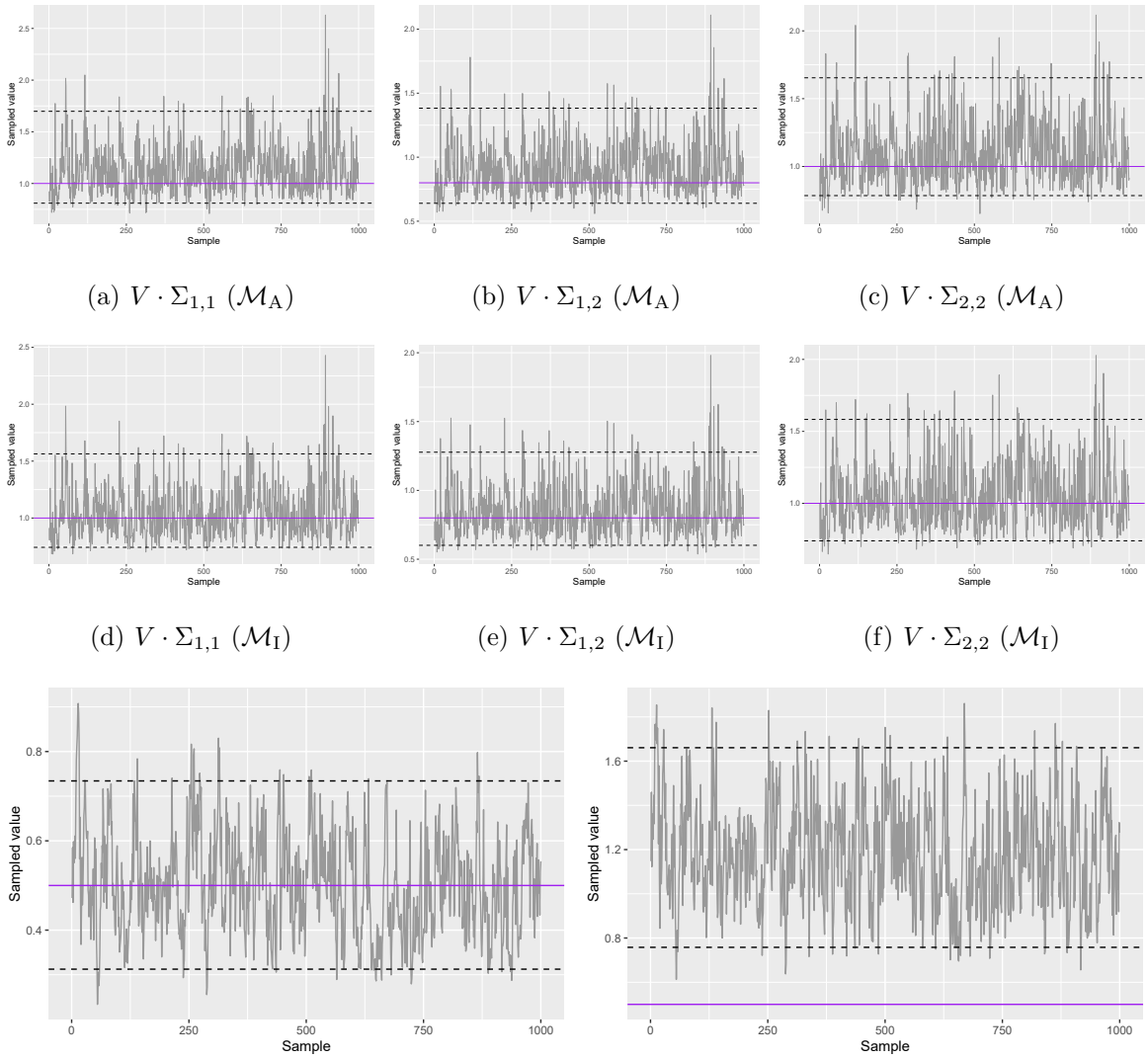
C.1.2 Second simulation study

Here we present some trace plots of the MCMC samples of the parameters involved in the second simulation study of Chapter 2:

- Figure C.9 shows the trace plots of the parameters ϕ, $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$ and $V \cdot \Sigma_{2,2}$ for models

$\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$.

- Figures C.10 and C.11 show the trace plots of the parameters $D_{m,n}$.

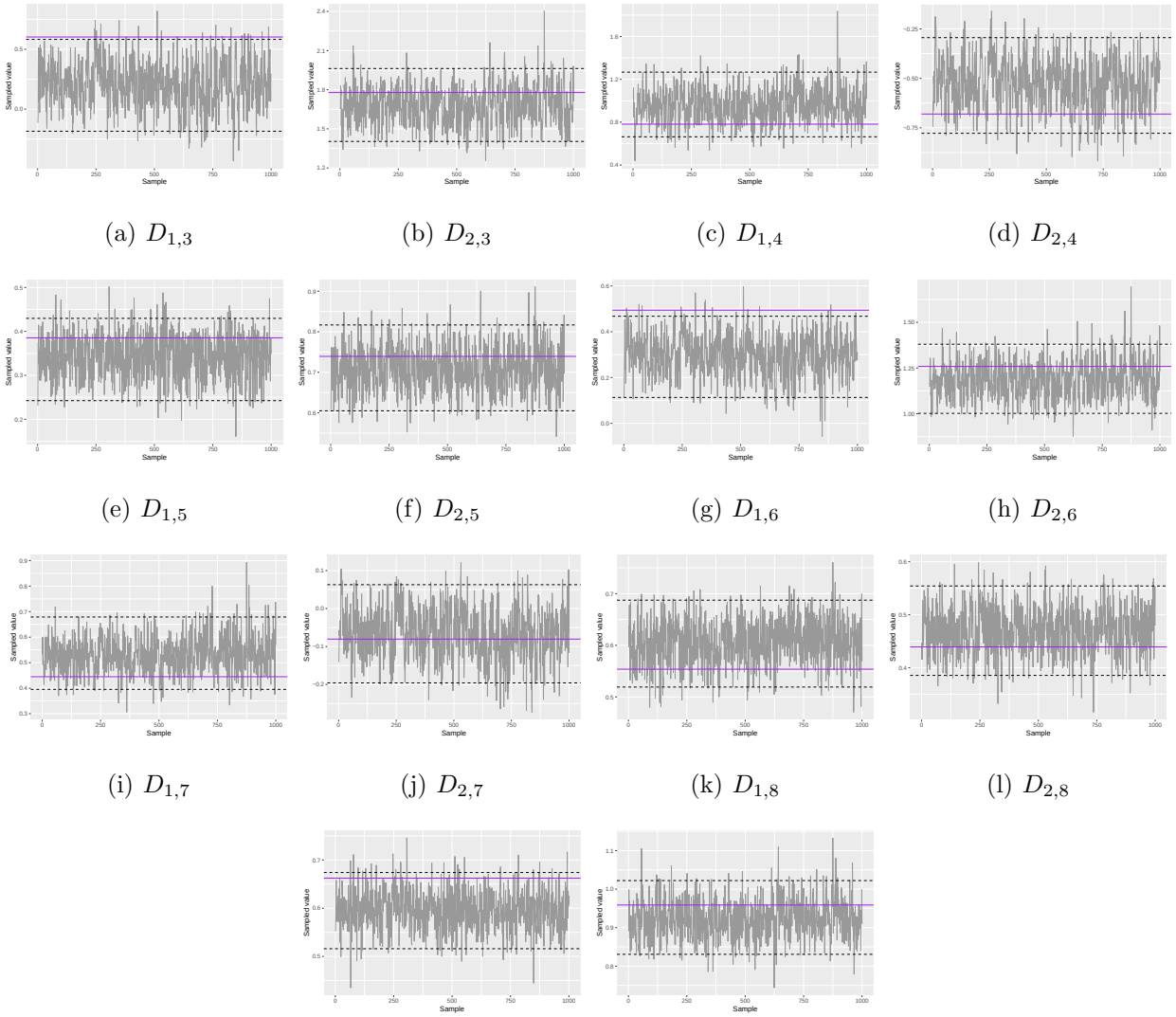


(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\text{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\text{A}}$) (c) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\text{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\text{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\text{I}}$) (f) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\text{I}}$)

(g) ϕ ($\mathcal{M}_{\text{A}}$) (h) ϕ ($\mathcal{M}_{\text{I}}$)

Figure C.9: Trace plots of the posterior distribution of the parameters $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$, $V \cdot \Sigma_{2,2}$ and ϕ by model, resulting from the second simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



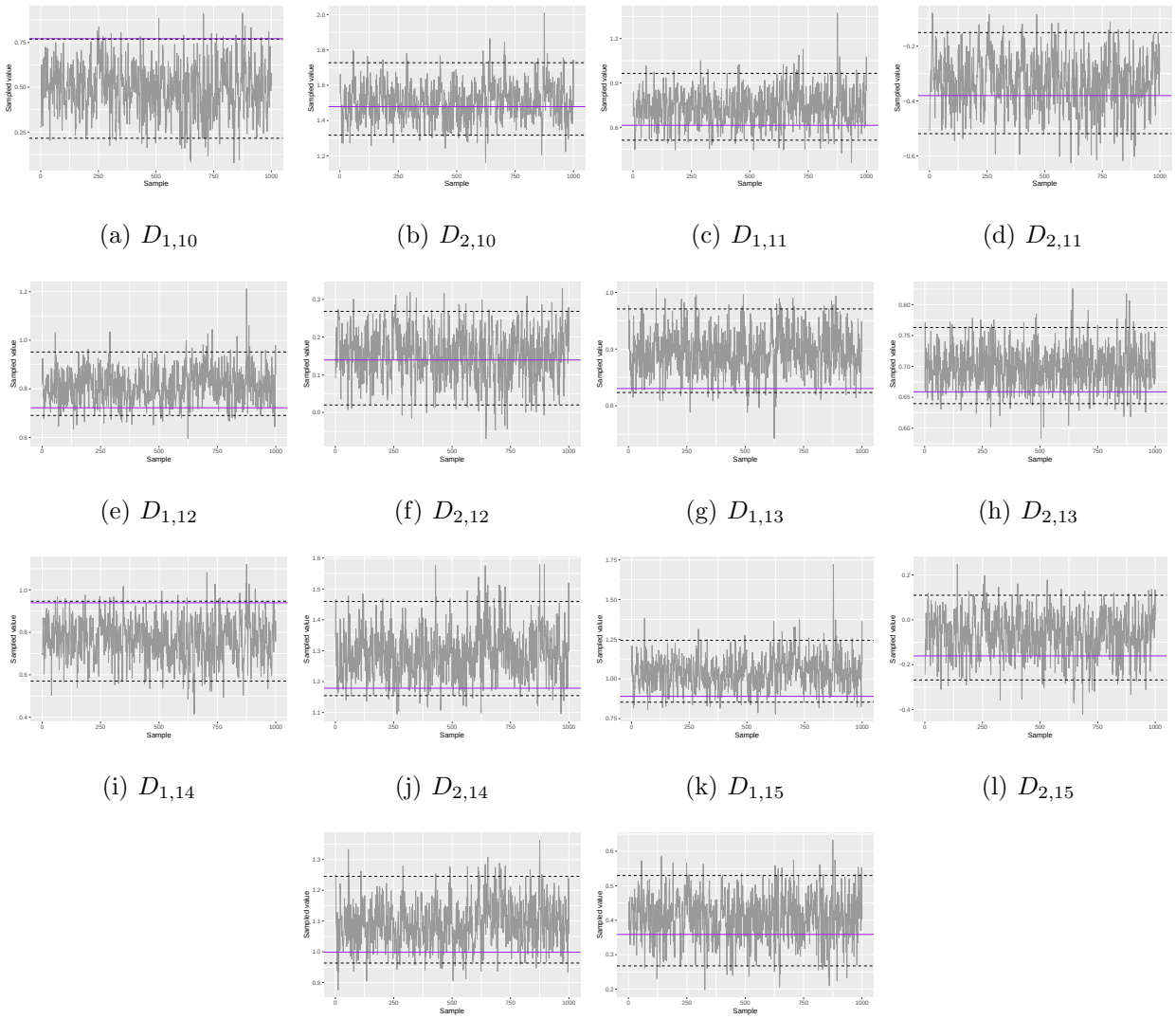
(a) $D_{1,3}$ (b) $D_{2,3}$ (c) $D_{1,4}$ (d) $D_{2,4}$

(e) $D_{1,5}$ (f) $D_{2,5}$ (g) $D_{1,6}$ (h) $D_{2,6}$

(i) $D_{1,7}$ (j) $D_{2,7}$ (k) $D_{1,8}$ (l) $D_{2,8}$

(m) $D_{1,9}$ (n) $D_{2,9}$

Figure C.10: Trace plots of the posterior distributions of $D_{1,n}$ and $D_{2,n}$, where $n \in \{3, 4, \ldots, 9\}$, resulting from the second simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



(a) $D_{1,10}$ (b) $D_{2,10}$ (c) $D_{1,11}$ (d) $D_{2,11}$

(e) $D_{1,12}$ (f) $D_{2,12}$ (g) $D_{1,13}$ (h) $D_{2,13}$

(i) $D_{1,14}$ (j) $D_{2,14}$ (k) $D_{1,15}$ (l) $D_{2,15}$

(m) $D_{1,16}$ (n) $D_{2,16}$

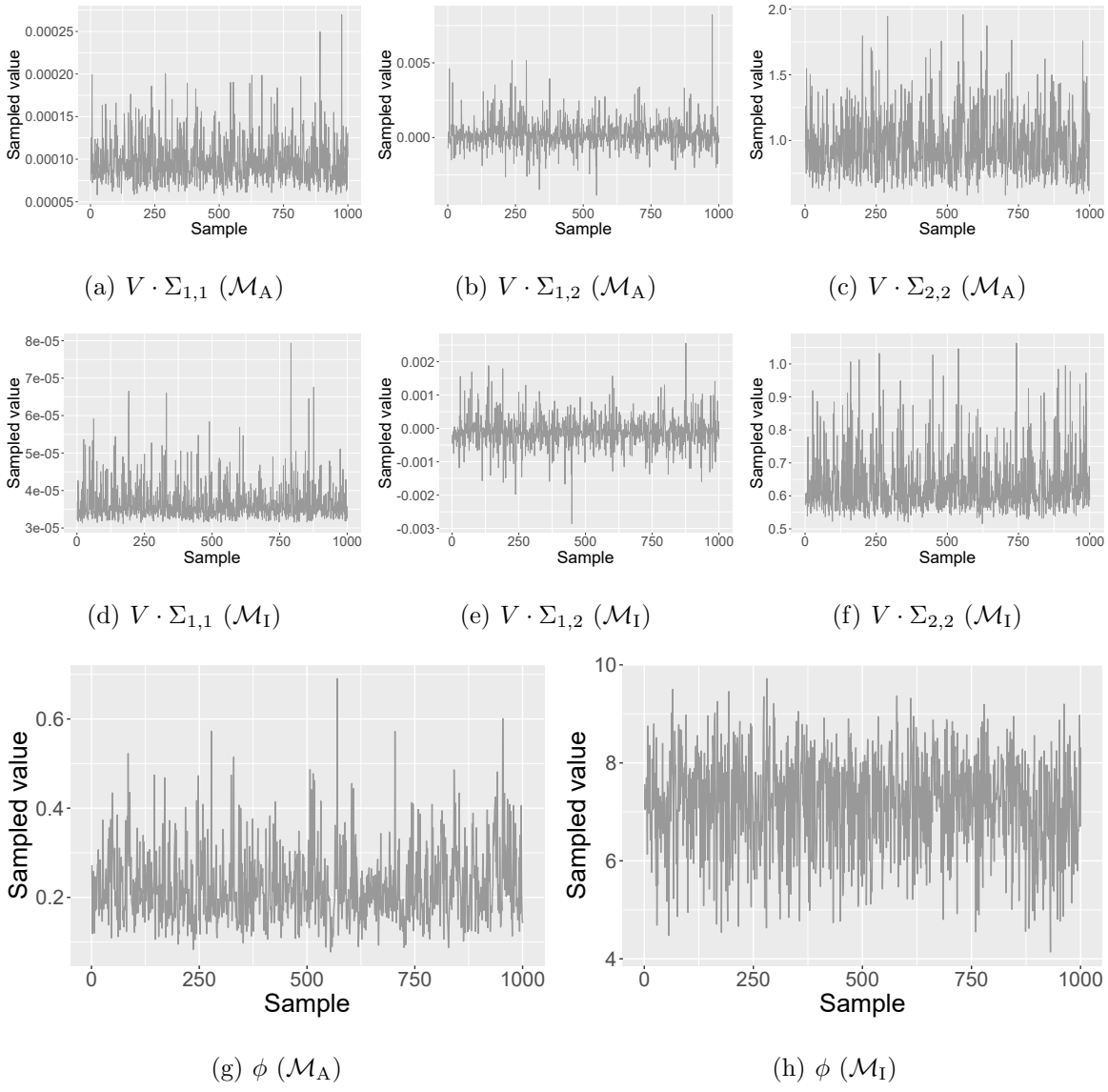
Figure C.11: Trace plots of the posterior distributions of $D_{1,n}$ and $D_{2,n}$, where $n \in \{10, 11, \ldots, 16\}$, resulting from the second simulation study in Chapter 2. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.

C.1.3 Illustrative example

Here we present some trace plots of the MCMC samples of the parameters involved in the illustrative example of Chapter 2:

- Figure C.12 shows the trace plots of the parameters ϕ, $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$ and $V \cdot \Sigma_{2,2}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$.

- Figure C.13 shows the trace plots of the parameters $D_{m,n}$.

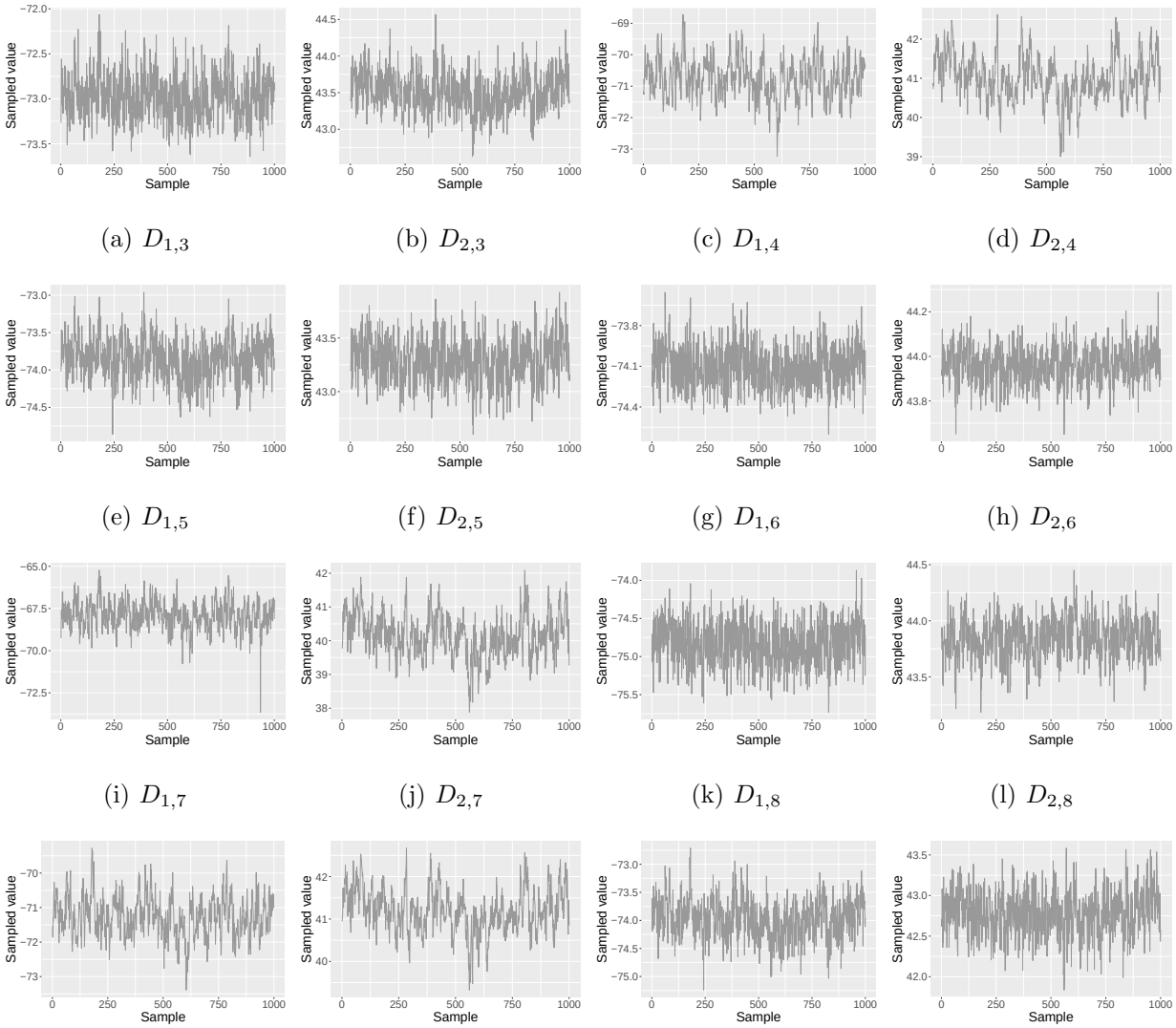


(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$)

(g) ϕ ($\mathcal{M}_{\mathrm{A}}$) (h) ϕ ($\mathcal{M}_{\mathrm{I}}$)

Figure C.12: Trace plots of the posterior distribution of the parameters $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$, $V \cdot \Sigma_{2,2}$ and ϕ by model, resulting from the illustrative example in Chapter 2.



(a) $D_{1,3}$ (b) $D_{2,3}$ (c) $D_{1,4}$ (d) $D_{2,4}$

(e) $D_{1,5}$ (f) $D_{2,5}$ (g) $D_{1,6}$ (h) $D_{2,6}$

(i) $D_{1,7}$ (j) $D_{2,7}$ (k) $D_{1,8}$ (l) $D_{2,8}$

(m) $D_{1,9}$ (n) $D_{2,9}$ (o) $D_{1,10}$ (p) $D_{2,10}$

Figure C.13: Trace plots of the posterior distributions of $D_{1,n}$ and $D_{2,n}$, where $n \in \{3, 4, \ldots, 10\}$, resulting from the illustrative example in Chapter 2.

C.2 Supplemental material of Chapter 3

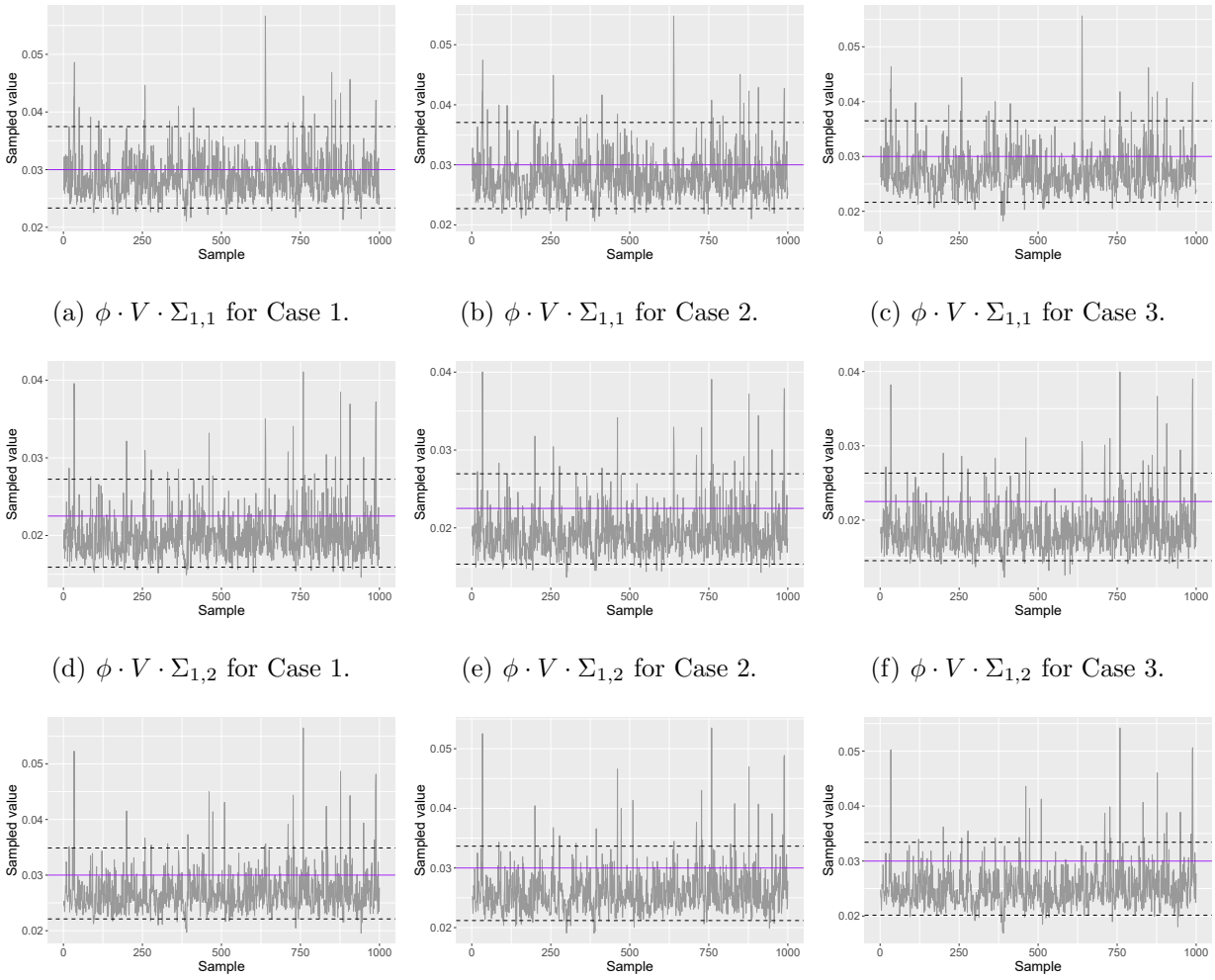
C.2.1 First simulation study

Here we present some trace plots of the MCMC samples of the parameters involved in the first simulation study of Chapter 3:

- Figure C.14 shows the trace plots of the parameters $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ for

Cases 1, 2 and 3.

- Figures C.15, C.16, C.17, C.18, C.19 and C.20 show the trace plots of the parameters $D_{m,n}$ for Cases 1, 2 and 3.

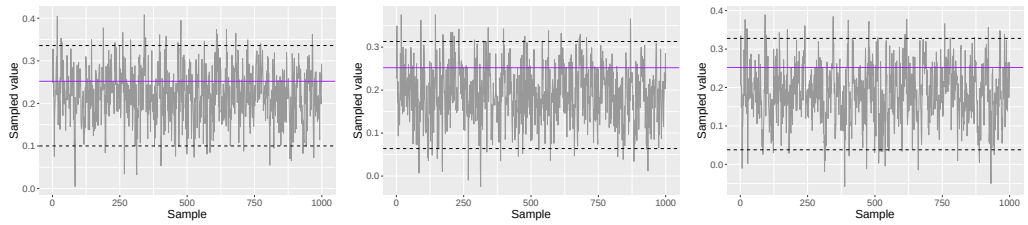


(a) $\phi \cdot V \cdot \Sigma_{1,1}$ for Case 1. (b) $\phi \cdot V \cdot \Sigma_{1,1}$ for Case 2. (c) $\phi \cdot V \cdot \Sigma_{1,1}$ for Case 3.

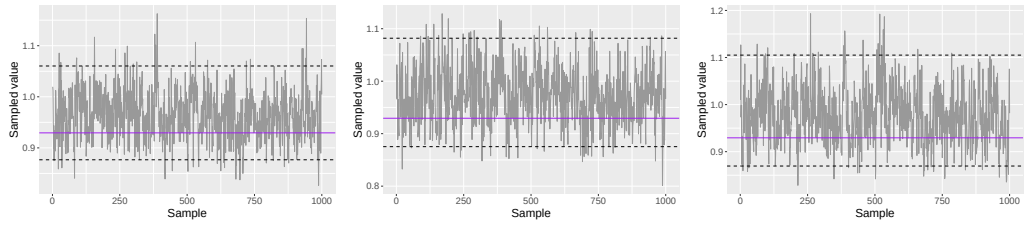
(d) $\phi \cdot V \cdot \Sigma_{1,2}$ for Case 1. (e) $\phi \cdot V \cdot \Sigma_{1,2}$ for Case 2. (f) $\phi \cdot V \cdot \Sigma_{1,2}$ for Case 3.

(g) $\phi \cdot V \cdot \Sigma_{2,2}$ for Case 1. (h) $\phi \cdot V \cdot \Sigma_{2,2}$ for Case 2. (i) $\phi \cdot V \cdot \Sigma_{2,2}$ for Case 3.

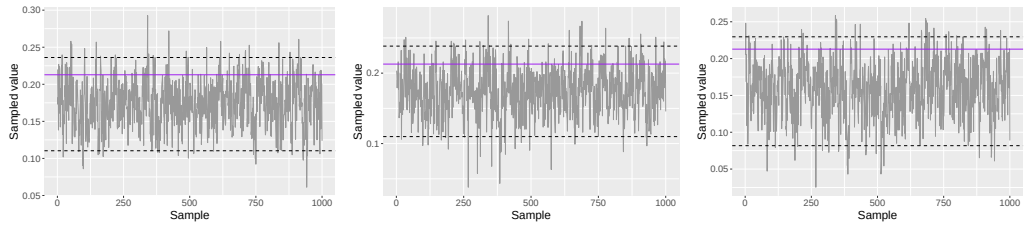
Figure C.14: Trace plots of the posterior distribution of the parameters $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ for $T \in \{10, 100, 1000\}$, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



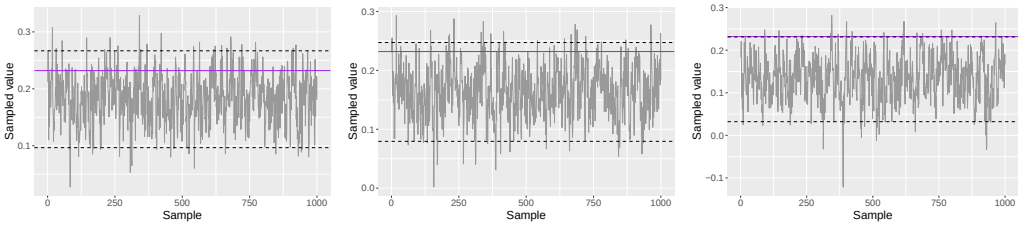
(a) $D_{1,3}$ for Case 1. (b) $D_{1,3}$ for Case 2. (c) $D_{1,3}$ for Case 3.



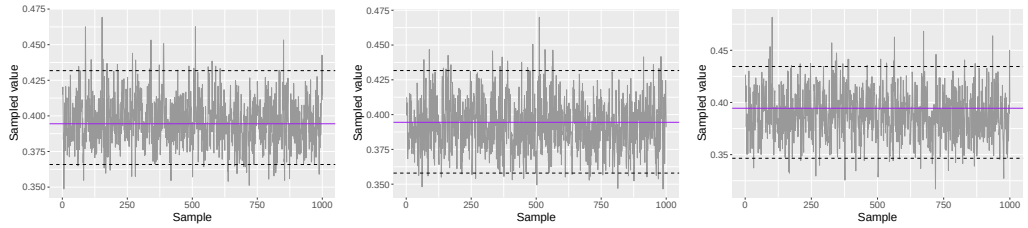
(d) $D_{1,4}$ for Case 1. (e) $D_{1,4}$ for Case 2. (f) $D_{1,4}$ for Case 3.



(g) $D_{1,5}$ for Case 1. (h) $D_{1,5}$ for Case 2. (i) $D_{1,5}$ for Case 3.

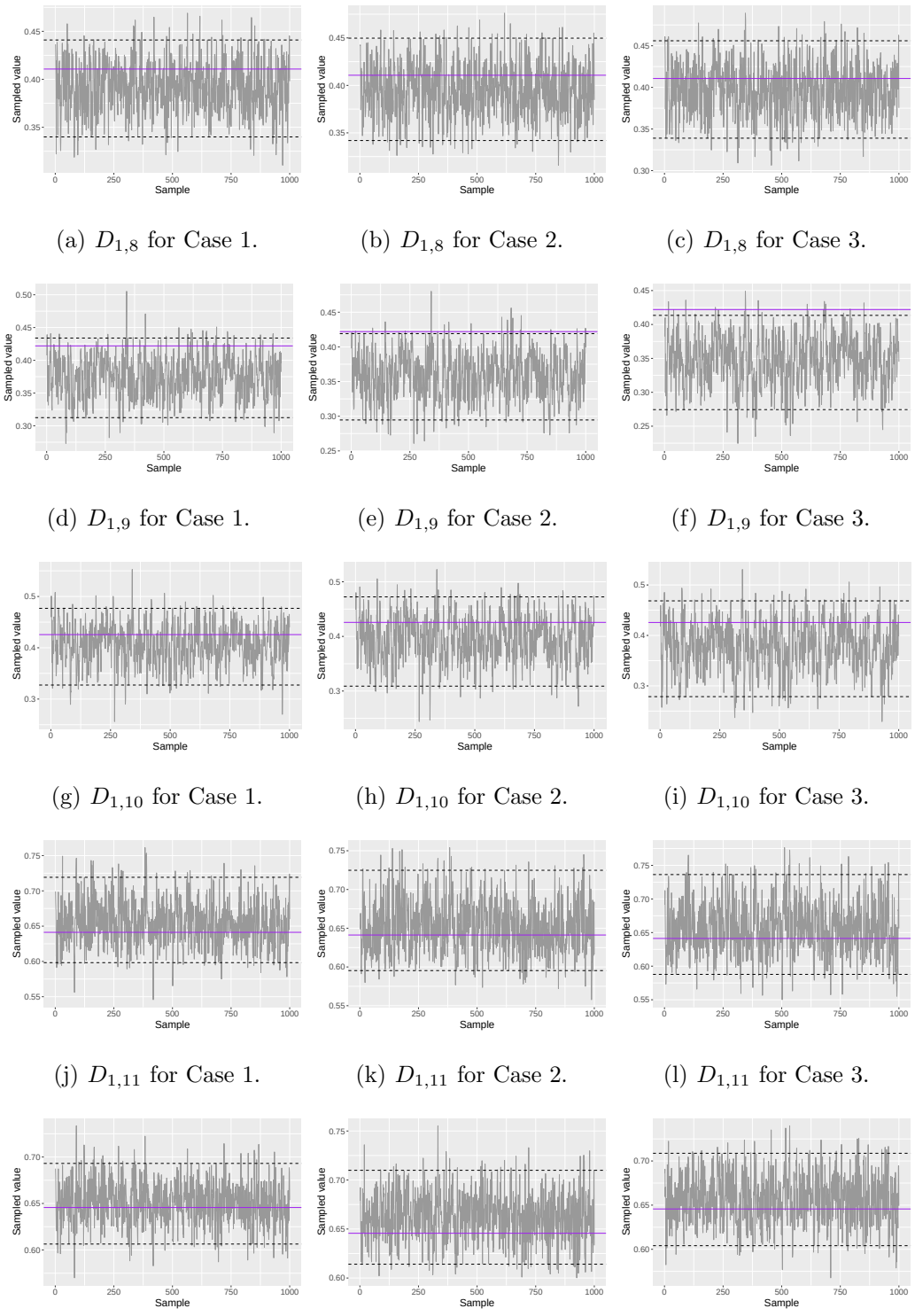


(j) $D_{1,6}$ for Case 1. (k) $D_{1,6}$ for Case 2. (l) $D_{1,6}$ for Case 3.



(m) $D_{1,7}$ for Case 1. (n) $D_{1,7}$ for Case 2. (o) $D_{1,7}$ for Case 3.

Figure C.15: Trace plots of the posterior distribution of $D_{1,n}$, $3 \leq n \leq 7$, for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



(a) $D_{1,8}$ for Case 1. (b) $D_{1,8}$ for Case 2. (c) $D_{1,8}$ for Case 3.

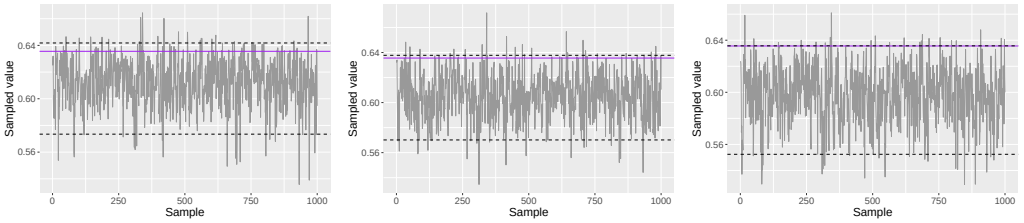
(d) $D_{1,9}$ for Case 1. (e) $D_{1,9}$ for Case 2. (f) $D_{1,9}$ for Case 3.

(g) $D_{1,10}$ for Case 1. (h) $D_{1,10}$ for Case 2. (i) $D_{1,10}$ for Case 3.

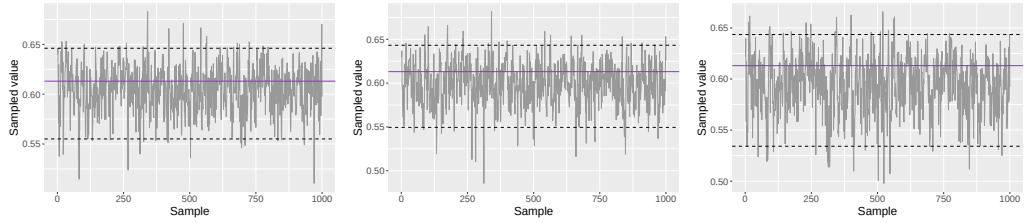
(j) $D_{1,11}$ for Case 1. (k) $D_{1,11}$ for Case 2. (l) $D_{1,11}$ for Case 3.

(m) $D_{1,12}$ for Case 1. (n) $D_{1,12}$ for Case 2. (o) $D_{1,12}$ for Case 3.

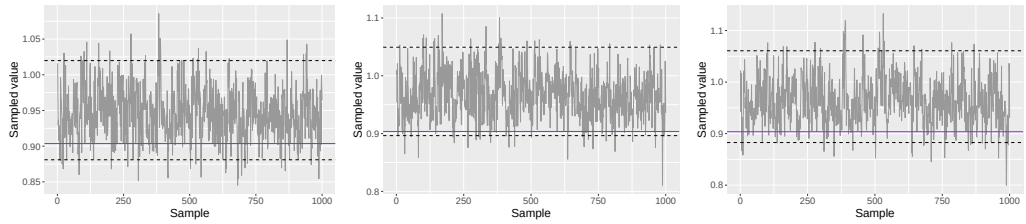
Figure C.16: Trace plots of the posterior distribution of $D_{1,n}$, $8 \leq n \leq 12$, for Cases 1, 2, and 3, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



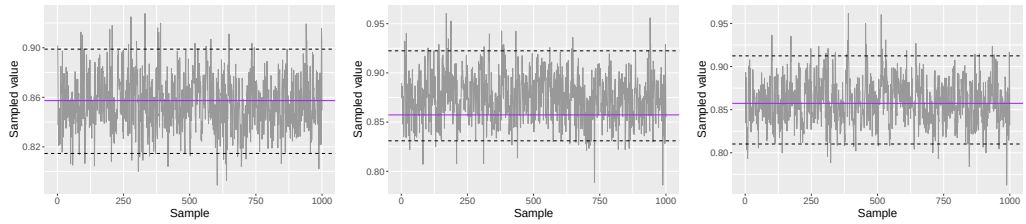
(a) $D_{1,13}$ for Case 1. (b) $D_{1,13}$ for Case 2. (c) $D_{1,13}$ for Case 3.



(d) $D_{1,14}$ for Case 1. (e) $D_{1,14}$ for Case 2. (f) $D_{1,14}$ for Case 3.

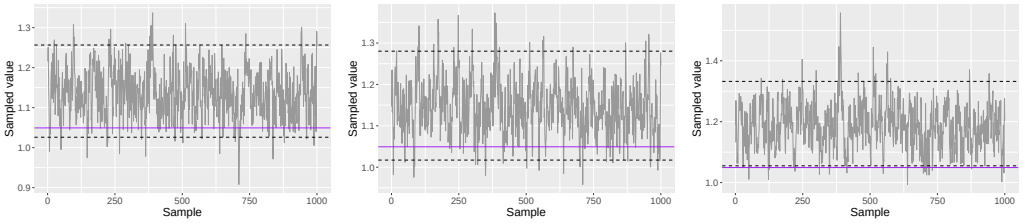


(g) $D_{1,15}$ for Case 1. (h) $D_{1,15}$ for Case 2. (i) $D_{1,15}$ for Case 3.

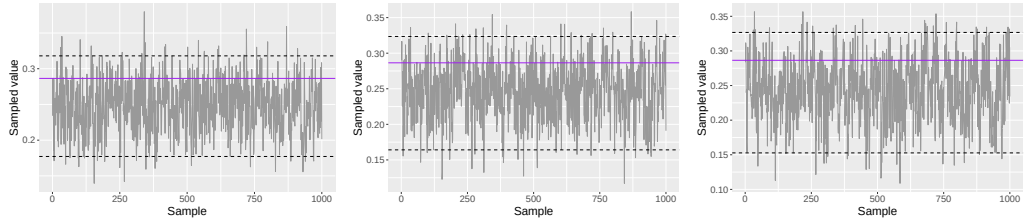


(j) $D_{1,16}$ for Case 1. (k) $D_{1,16}$ for Case 2. (l) $D_{1,16}$ for Case 3.

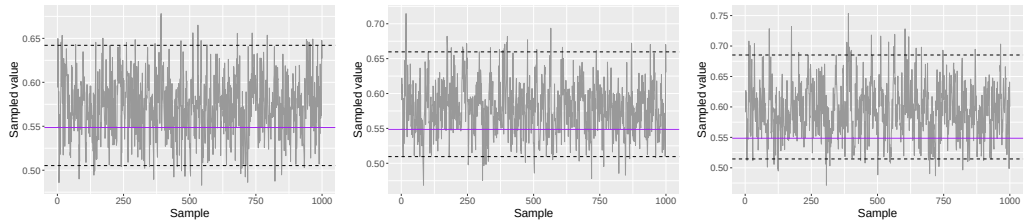
Figure C.17: Trace plots of the posterior distribution of $D_{1,n}$, $13 \leq n \leq 16$, for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



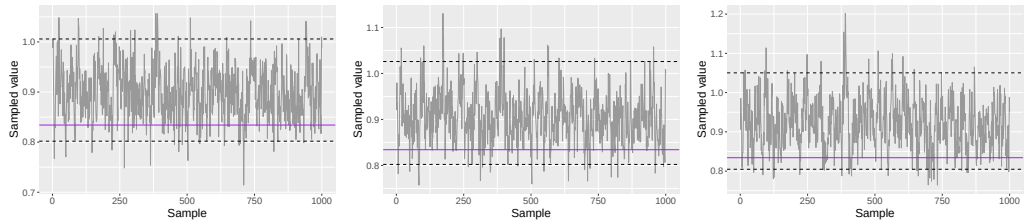
(a) $D_{2,3}$ for Case 1. (b) $D_{2,3}$ for Case 2. (c) $D_{2,3}$ for Case 3.



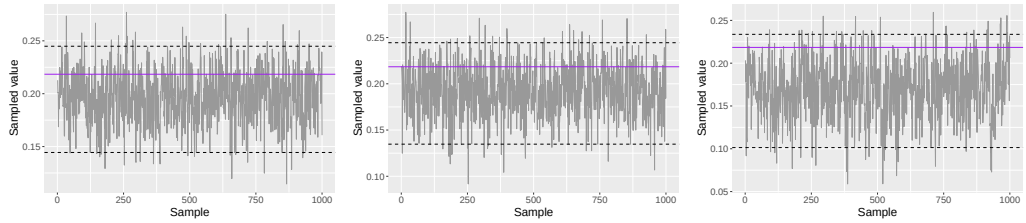
(d) $D_{2,4}$ for Case 1. (e) $D_{2,4}$ for Case 2. (f) $D_{2,4}$ for Case 3.



(g) $D_{2,5}$ for Case 1. (h) $D_{2,5}$ for Case 2. (i) $D_{2,5}$ for Case 3.

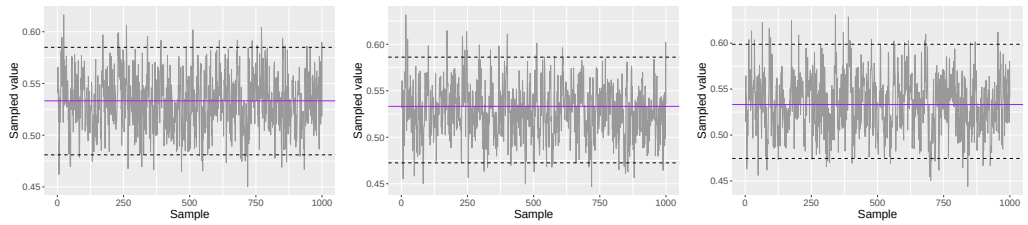


(j) $D_{2,6}$ for Case 1. (k) $D_{2,6}$ for Case 2. (l) $D_{2,6}$ for Case 3.

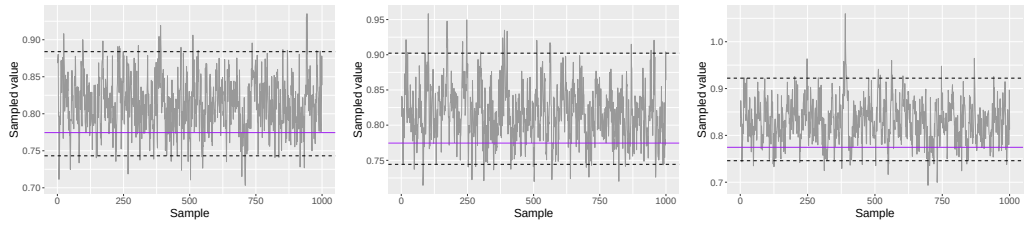


(m) $D_{2,7}$ for Case 1. (n) $D_{2,7}$ for Case 2. (o) $D_{2,7}$ for Case 3.

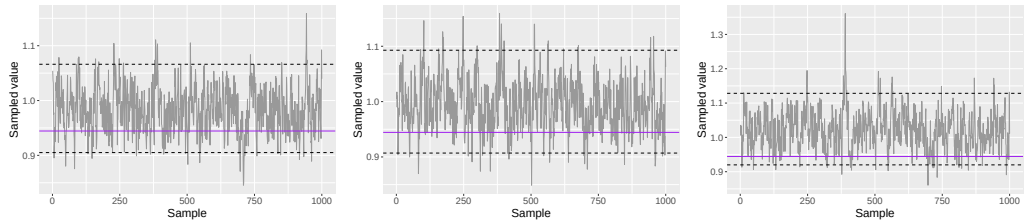
Figure C.18: Trace plots of the posterior distribution of $D_{2,n}$, $3 \leq n \leq 7$, for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



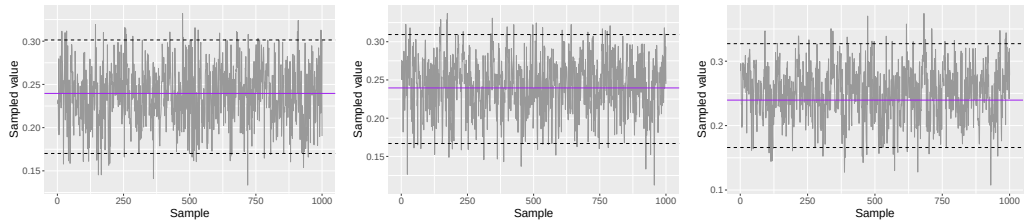
(a) $D_{2,8}$ for Case 1. (b) $D_{2,8}$ for Case 2. (c) $D_{2,8}$ for Case 3.



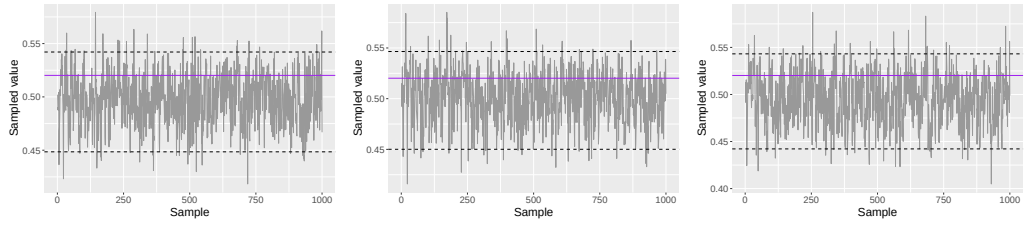
(d) $D_{2,9}$ for Case 1. (e) $D_{2,9}$ for Case 2. (f) $D_{2,9}$ for Case 3.



(g) $D_{2,10}$ for Case 1. (h) $D_{2,10}$ for Case 2. (i) $D_{2,10}$ for Case 3.

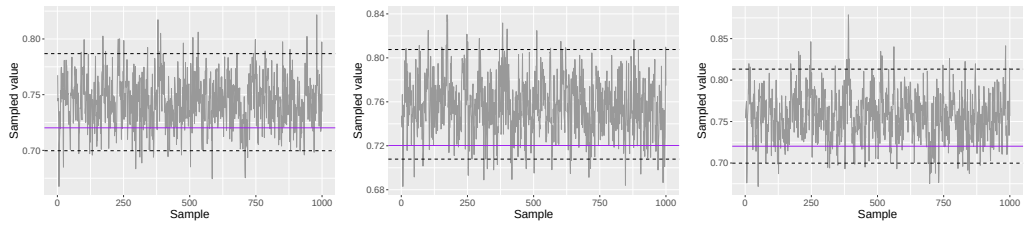


(j) $D_{2,11}$ for Case 1. (k) $D_{2,11}$ for Case 2. (l) $D_{2,11}$ for Case 3.

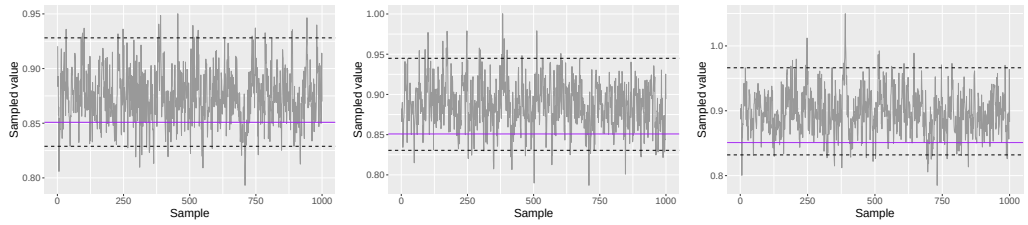


(m) $D_{2,12}$ for Case 1. (n) $D_{2,12}$ for Case 2. (o) $D_{2,12}$ for Case 3.

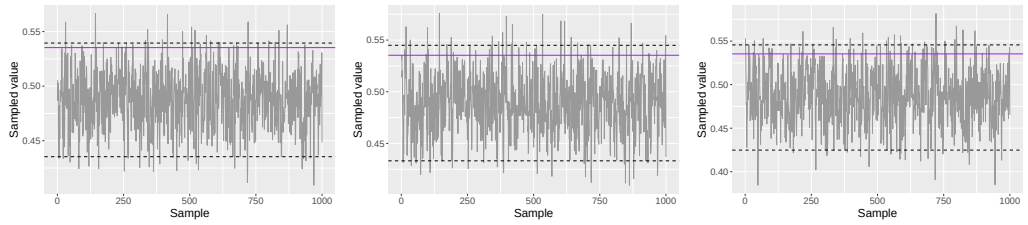
Figure C.19: Trace plots of the posterior distribution of $D_{2,n}$, $8 \leq n \leq 12$, for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



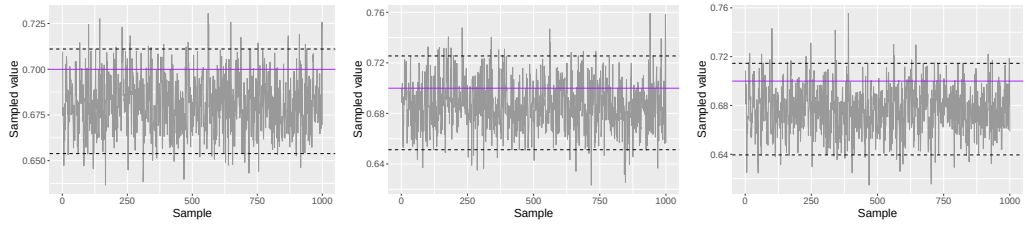
(a) $D_{2,13}$ for Case 1. (b) $D_{2,13}$ for Case 2. (c) $D_{2,13}$ for Case 3.



(d) $D_{2,14}$ for Case 1. (e) $D_{2,14}$ for Case 2. (f) $D_{2,14}$ for Case 3.



(g) $D_{2,15}$ for Case 1. (h) $D_{2,15}$ for Case 2. (i) $D_{2,15}$ for Case 3.



(j) $D_{2,16}$ for Case 1. (k) $D_{2,16}$ for Case 2. (l) $D_{2,16}$ for Case 3.

Figure C.20: Trace plots of the posterior distribution of $D_{2,n}$, $13 \leq n \leq 17$, for Cases 1, 2 and 3, resulting from the first simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.

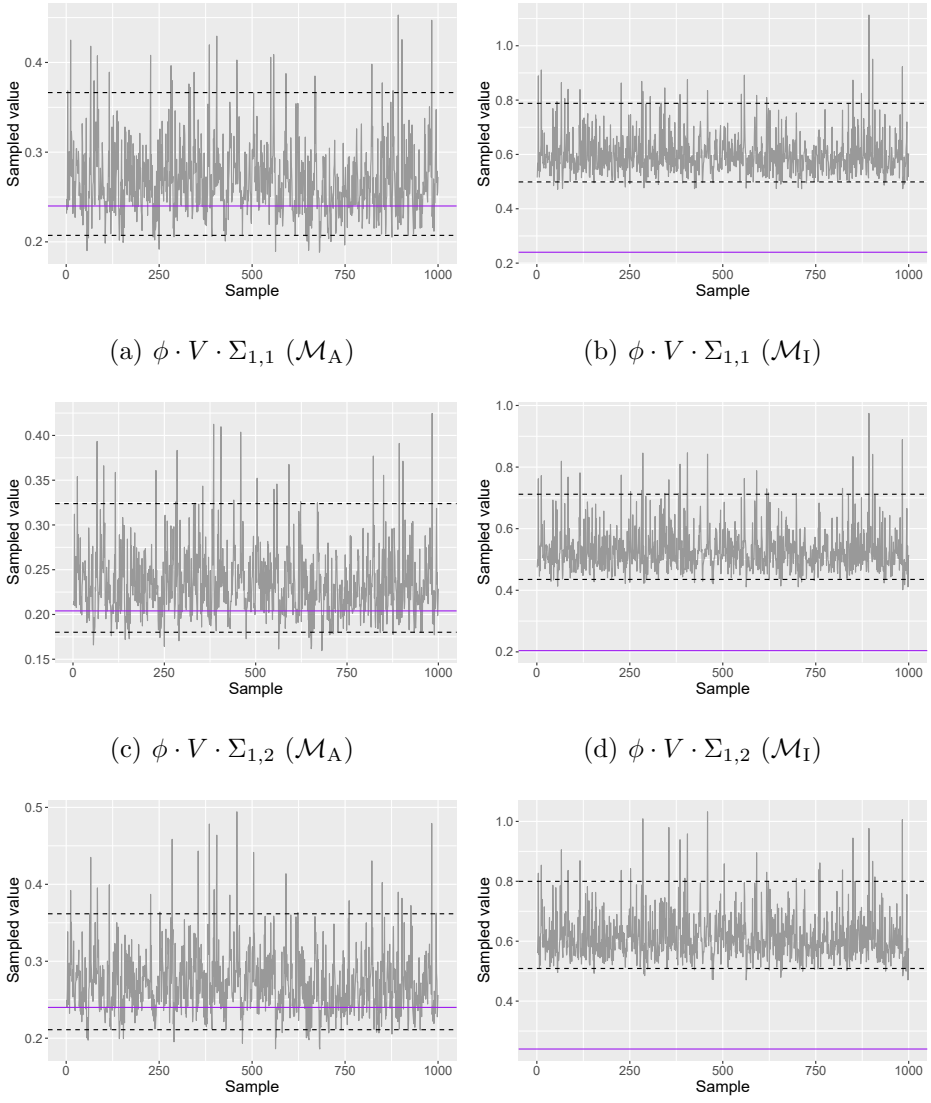
C.2.2 Second simulation study

Here we present some trace plots of the MCMC samples of the parameters involved in the second simulation study of Chapter 3:

- Figure C.21 shows the trace plots of the parameters $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ for

models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$.

- Figures C.22 and C.23 show the trace plots of the parameters $D_{m,n}$.



(a) $\phi \cdot V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\text{A}}$)

(b) $\phi \cdot V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\text{I}}$)

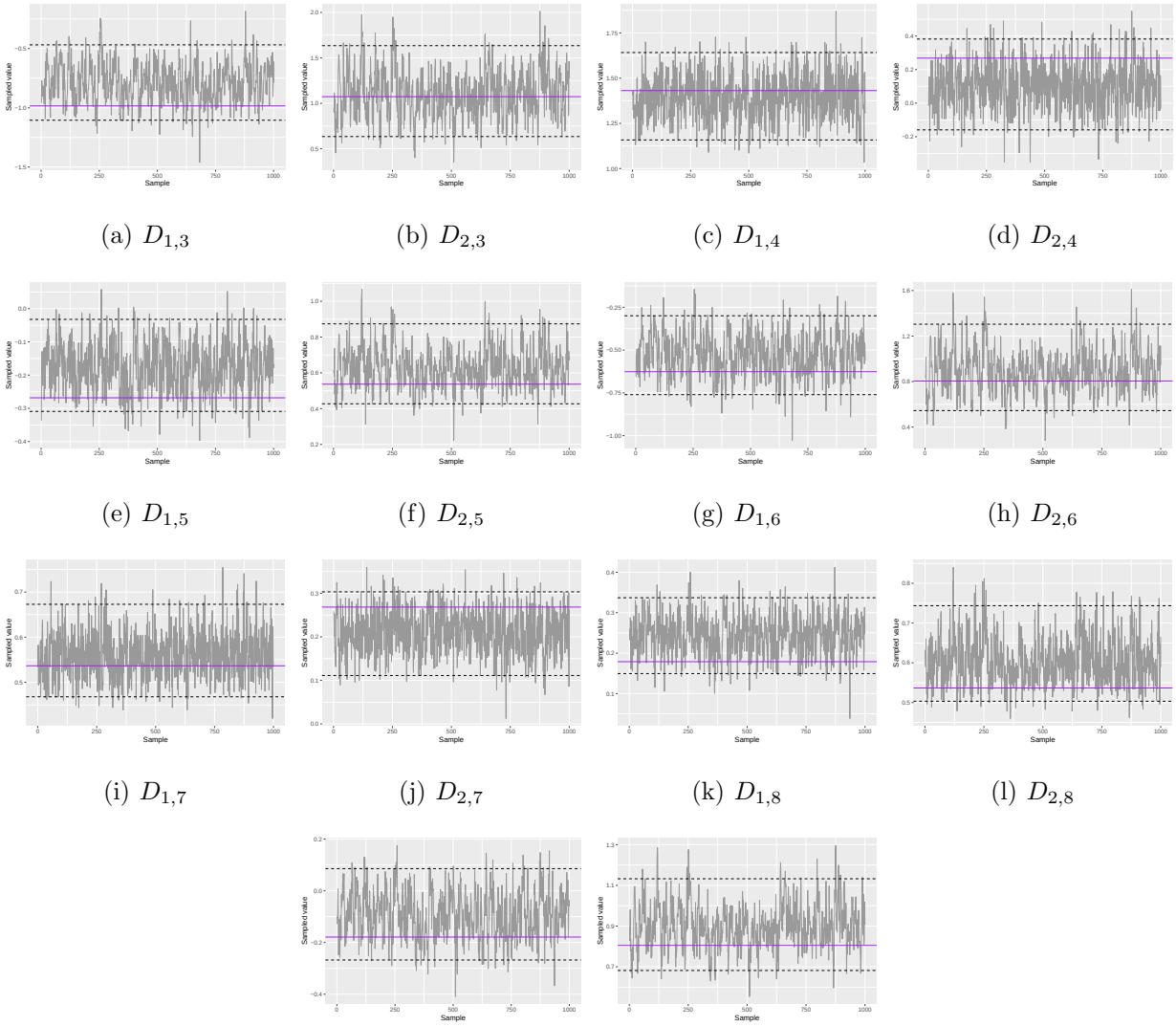
(c) $\phi \cdot V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\text{A}}$)

(d) $\phi \cdot V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\text{I}}$)

(e) $\phi \cdot V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\text{A}}$)

(f) $\phi \cdot V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\text{I}}$)

Figure C.21: Trace plots of the posterior distribution of the parameters $\phi \cdot V \cdot \Sigma_{1,1}$, $\phi \cdot V \cdot \Sigma_{1,2}$ and $\phi \cdot V \cdot \Sigma_{2,2}$ by model, resulting from the second simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



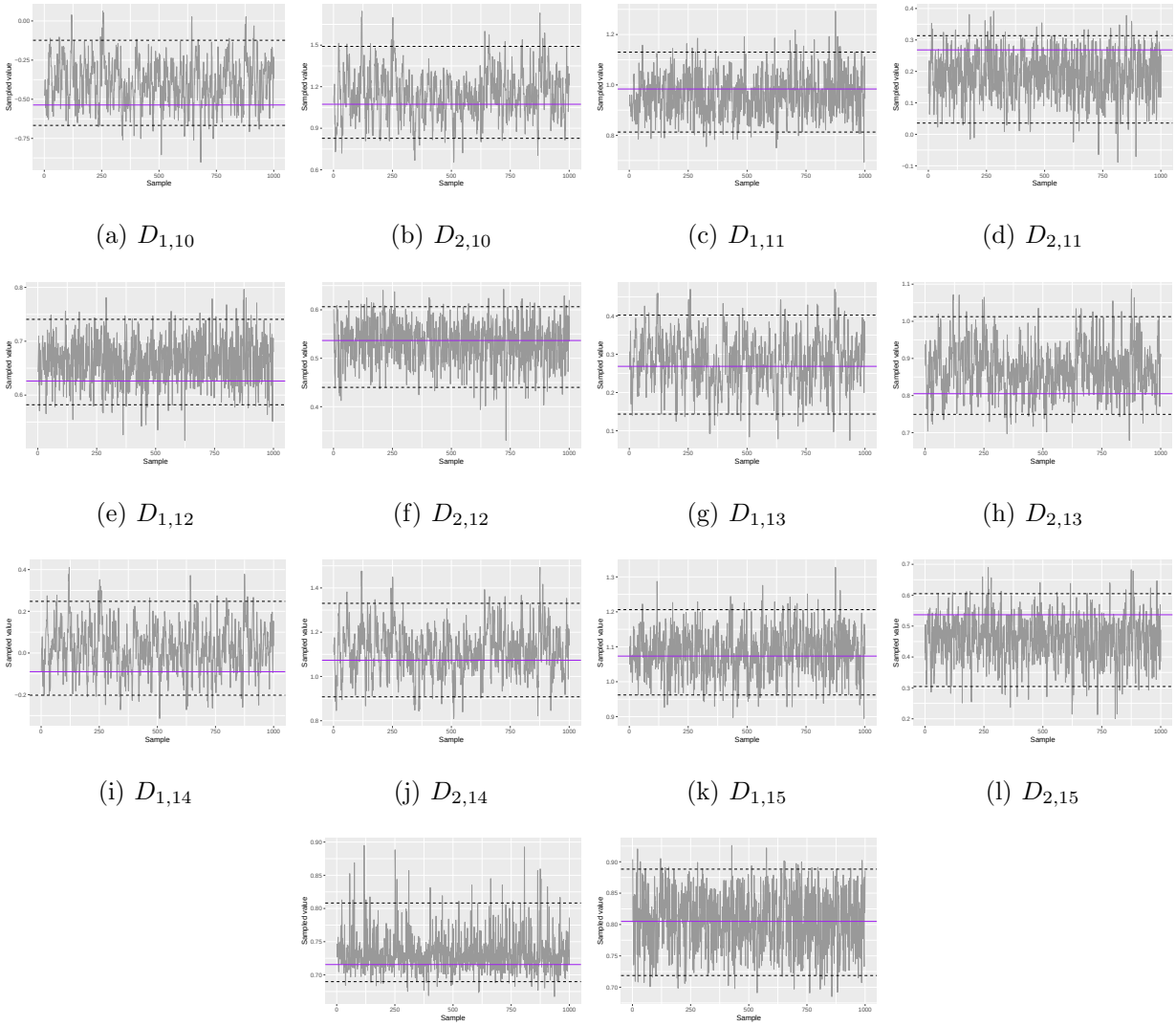
(a) $D_{1,3}$ (b) $D_{2,3}$ (c) $D_{1,4}$ (d) $D_{2,4}$

(e) $D_{1,5}$ (f) $D_{2,5}$ (g) $D_{1,6}$ (h) $D_{2,6}$

(i) $D_{1,7}$ (j) $D_{2,7}$ (k) $D_{1,8}$ (l) $D_{2,8}$

(m) $D_{1,9}$ (n) $D_{2,9}$

Figure C.22: Trace plots of the posterior distributions of $D_{1,n}$ and $D_{2,n}$, where $n \in \{3, 4, \ldots, 9\}$, resulting from the second simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.



(a) $D_{1,10}$ (b) $D_{2,10}$ (c) $D_{1,11}$ (d) $D_{2,11}$

(e) $D_{1,12}$ (f) $D_{2,12}$ (g) $D_{1,13}$ (h) $D_{2,13}$

(i) $D_{1,14}$ (j) $D_{2,14}$ (k) $D_{1,15}$ (l) $D_{2,15}$

(m) $D_{1,16}$ (n) $D_{2,16}$

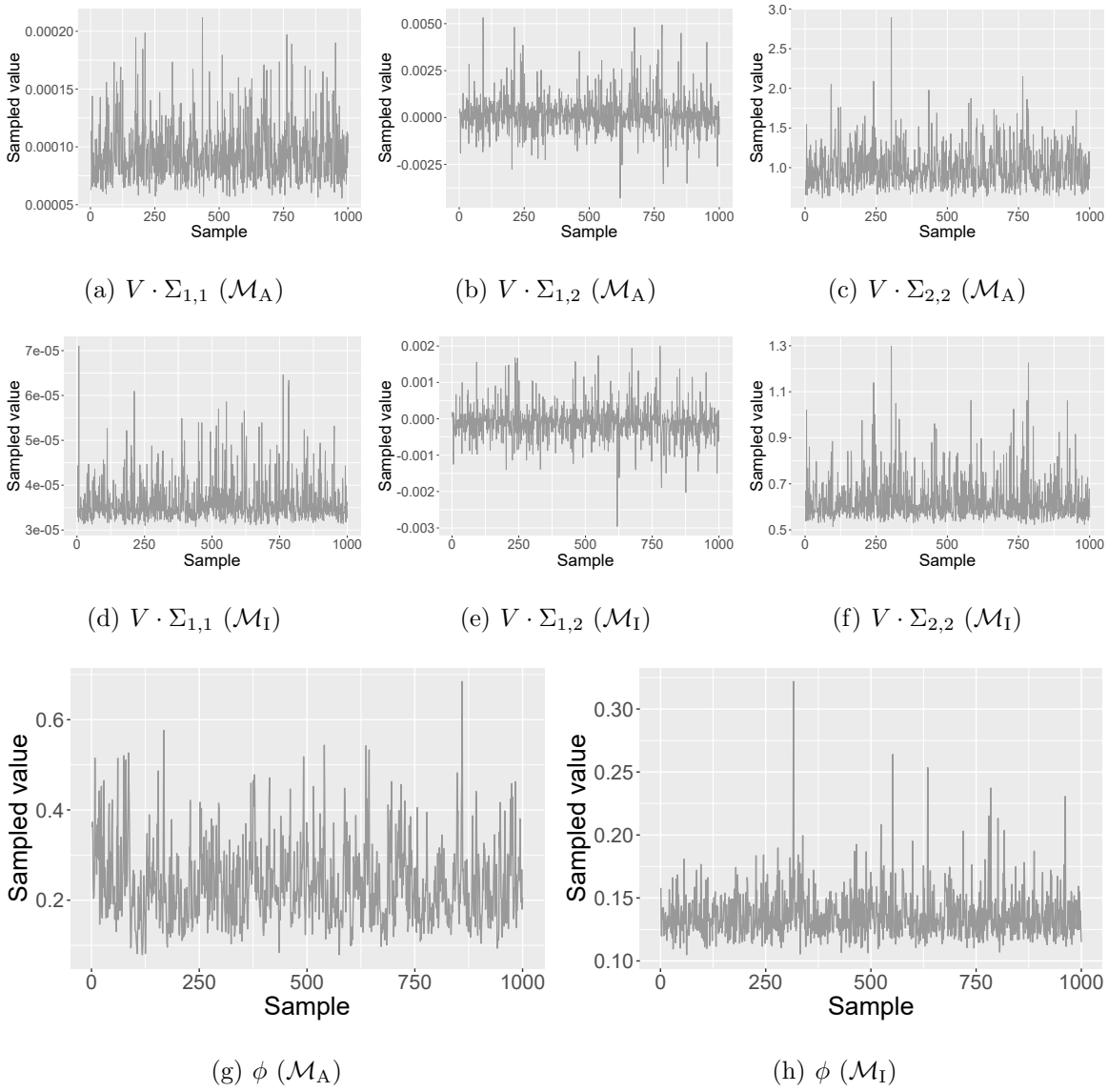
Figure C.23: Trace plots of the posterior distributions of $D_{1,n}$ and $D_{2,n}$, where $n \in \{10, 11, \ldots, 16\}$, resulting from the second simulation study in Chapter 3. True values are represented by solid purple line and the 2.5^{th} and 97.5^{th} posterior quantiles are represented by the black dashed lines.

C.2.3 Illustrative example

Here we present some trace plots of the MCMC samples of the parameters involved in the illustrative example of Chapter 3:

- Figure C.24 shows the trace plots of the parameters ϕ, $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$ and $V \cdot \Sigma_{2,2}$ for models $\mathcal{M}_{\text{A}}$ and $\mathcal{M}_{\text{I}}$.

- Figure C.25 shows the trace plots of the parameters $D_{m,n}$.

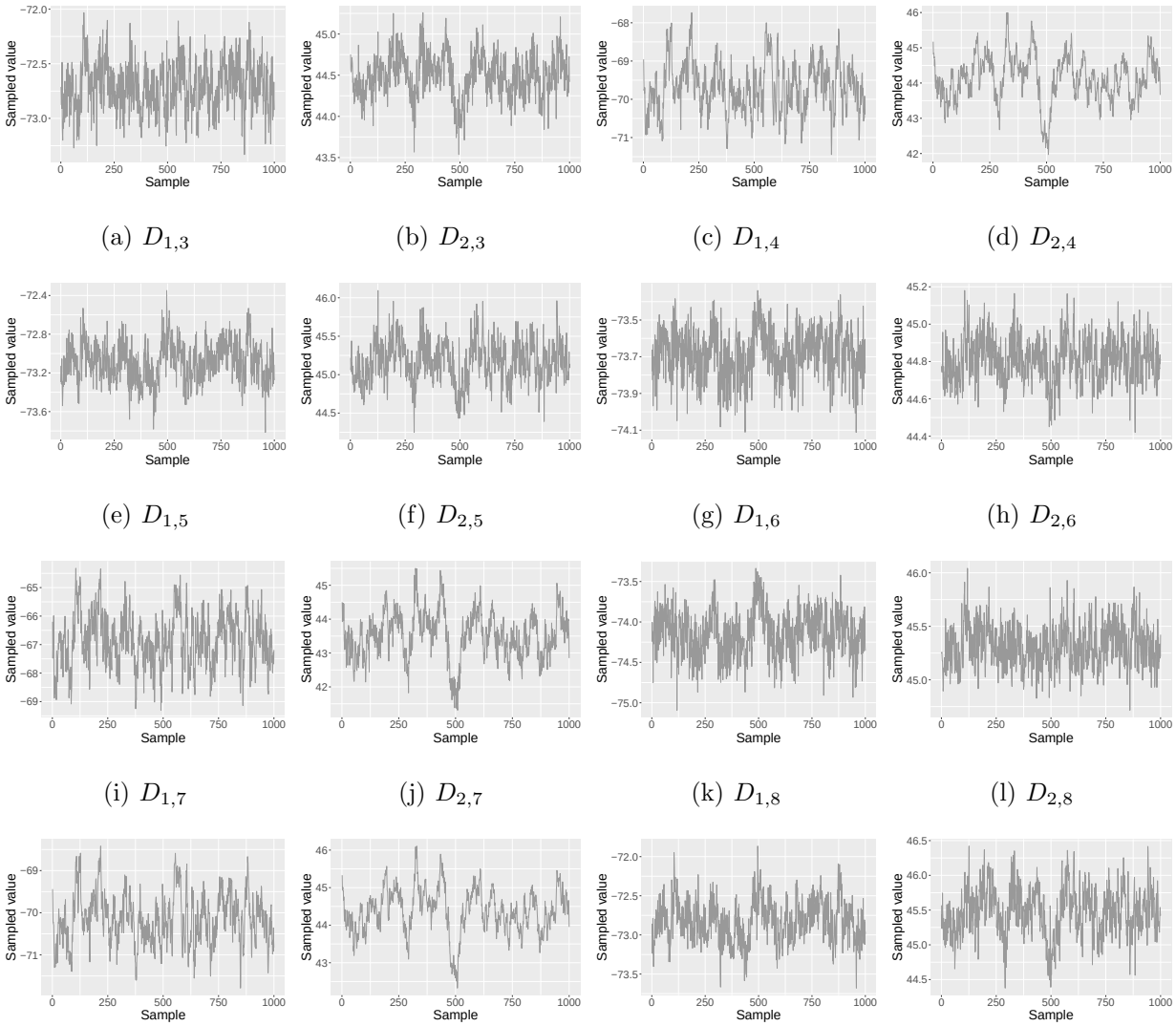


(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$)

(g) ϕ ($\mathcal{M}_{\mathrm{A}}$) (h) ϕ ($\mathcal{M}_{\mathrm{I}}$)

Figure C.24: Trace plots of the posterior distribution of the parameters $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$, $V \cdot \Sigma_{2,2}$ and ϕ by model, resulting from the illustrative example in Chapter 3.



(a) $D_{1,3}$ (b) $D_{2,3}$ (c) $D_{1,4}$ (d) $D_{2,4}$

(e) $D_{1,5}$ (f) $D_{2,5}$ (g) $D_{1,6}$ (h) $D_{2,6}$

(i) $D_{1,7}$ (j) $D_{2,7}$ (k) $D_{1,8}$ (l) $D_{2,8}$

(m) $D_{1,9}$ (n) $D_{2,9}$ (o) $D_{1,10}$ (p) $D_{2,10}$

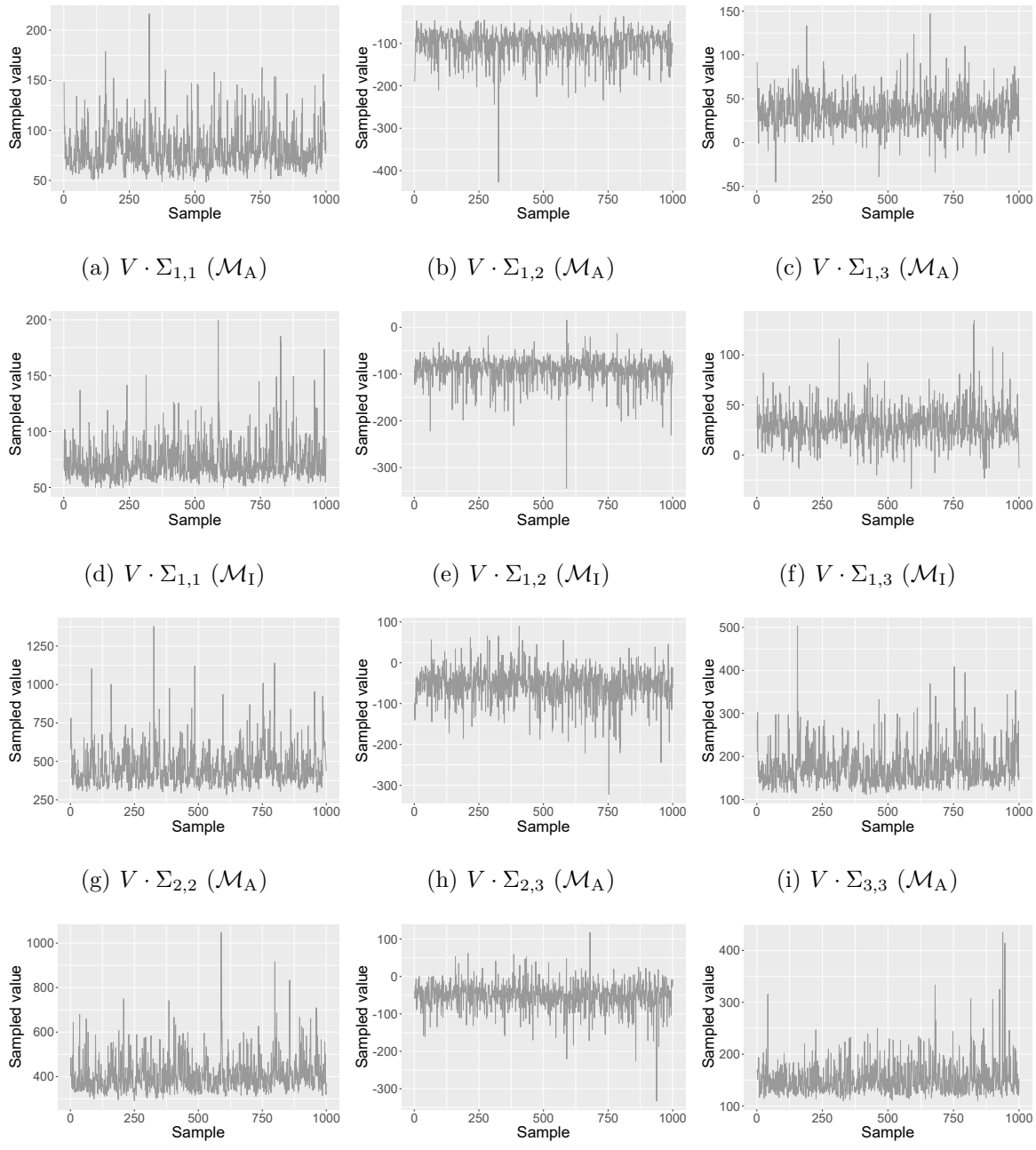
Figure C.25: Trace plots of the posterior distributions of $D_{1,n}$ and $D_{2,n}$, where $n \in \{3, 4, \ldots, 10\}$, resulting from the illustrative example in Chapter 3.

C.3 Supplemental material of Chapter 4

Here we present some trace plots of the MCMC samples of the parameters involved in the practical example of Chapter 4:

- Figure C.26 shows the trace plots of the parameters $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$, $V \cdot \Sigma_{1,3}$, $V \cdot \Sigma_{2,2}$, $V \cdot \Sigma_{2,3}$ and $V \cdot \Sigma_{3,3}$ for models $\mathcal{M}_{\mathrm{A}}$ and $\mathcal{M}_{\mathrm{I}}$.

- Figure C.27 shows the trace plots of the parameters ϕ and $D_{m,n}$.



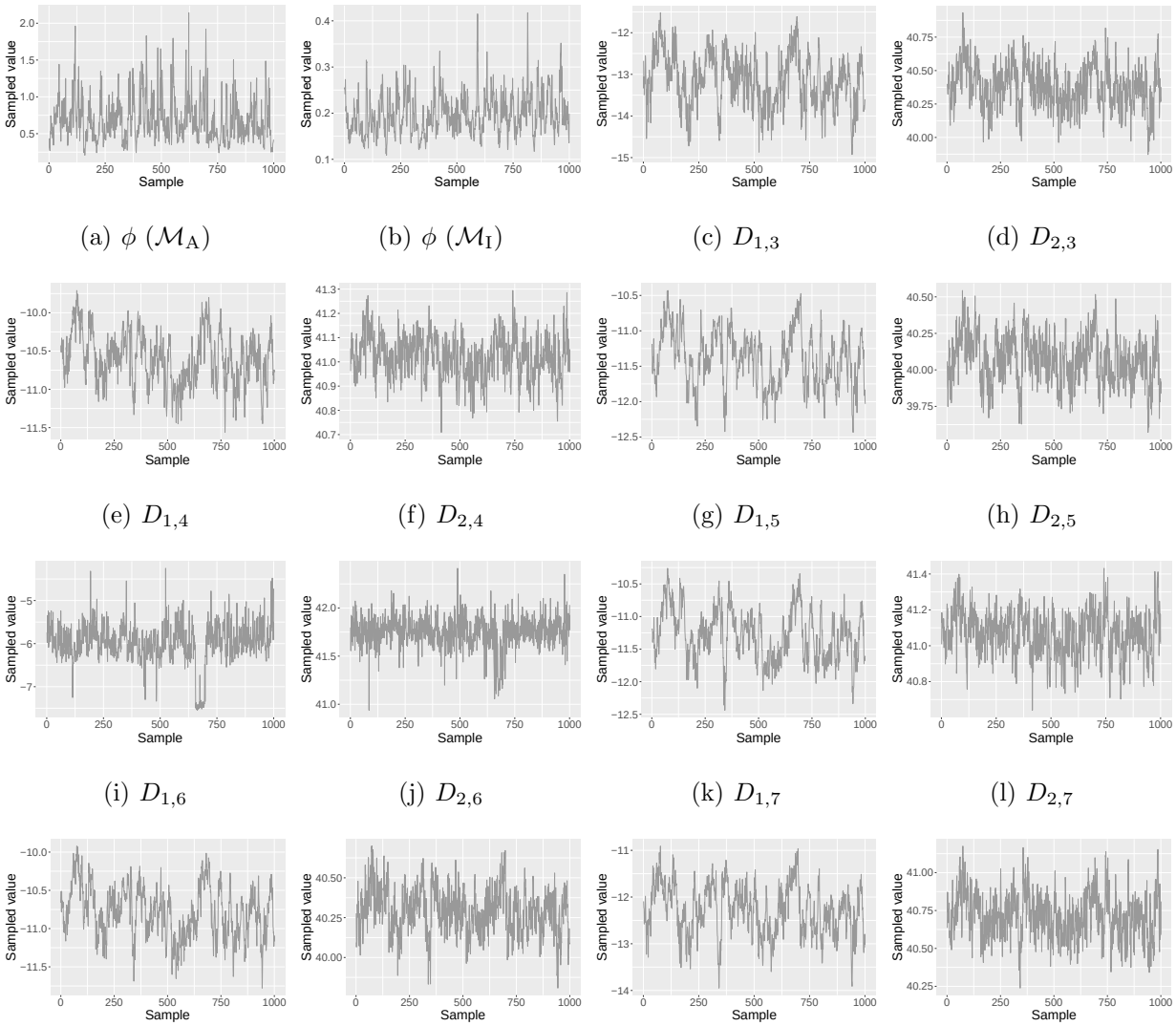
(a) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{A}}$) (b) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{A}}$) (c) $V \cdot \Sigma_{1,3}$ ($\mathcal{M}_{\mathrm{A}}$)

(d) $V \cdot \Sigma_{1,1}$ ($\mathcal{M}_{\mathrm{I}}$) (e) $V \cdot \Sigma_{1,2}$ ($\mathcal{M}_{\mathrm{I}}$) (f) $V \cdot \Sigma_{1,3}$ ($\mathcal{M}_{\mathrm{I}}$)

(g) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{A}}$) (h) $V \cdot \Sigma_{2,3}$ ($\mathcal{M}_{\mathrm{A}}$) (i) $V \cdot \Sigma_{3,3}$ ($\mathcal{M}_{\mathrm{A}}$)

(j) $V \cdot \Sigma_{2,2}$ ($\mathcal{M}_{\mathrm{I}}$) (k) $V \cdot \Sigma_{2,3}$ ($\mathcal{M}_{\mathrm{I}}$) (l) $V \cdot \Sigma_{3,3}$ ($\mathcal{M}_{\mathrm{I}}$)

Figure C.26: Trace plots of the posterior distribution of the parameters $V \cdot \Sigma_{1,1}$, $V \cdot \Sigma_{1,2}$, $V \cdot \Sigma_{1,3}$, $V \cdot \Sigma_{2,2}$, $V \cdot \Sigma_{2,3}$ and $V \cdot \Sigma_{3,3}$ by model, resulting from the practical example in Chapter 4.



(a) ϕ ($\mathcal{M}_{\mathrm{A}}$) (b) ϕ ($\mathcal{M}_{\mathrm{I}}$) (c) $D_{1,3}$ (d) $D_{2,3}$

(e) $D_{1,4}$ (f) $D_{2,4}$ (g) $D_{1,5}$ (h) $D_{2,5}$

(i) $D_{1,6}$ (j) $D_{2,6}$ (k) $D_{1,7}$ (l) $D_{2,7}$

(m) $D_{1,8}$ (n) $D_{2,8}$ (o) $D_{1,9}$ (p) $D_{2,9}$

Figure C.27: Trace plots of the posterior distributions of the parameters ϕ, $D_{1,n}$ and $D_{2,n}$, where $n \in \{3, 4, \ldots, 9\}$, resulting from the practical example in Chapter 4.

Bibliography

Banerjee, S., Carlin, B. P., and Gelfand, A. E. (2014). *Hierarchical Modeling and Analysis for Spatial Data*. 2nd ed. Chapman & Hall/CRC.

Berliner, L. M. (1996). "Hierarchical Bayesian Time Series Models." In *Maximum Entropy and Bayesian Methods*, eds. K. M. Hanson and R. N. Silver, 15–22. Springer.

Bruno, F., Guttorp, P., Sampson, P. D., and Cocchi, D. (2009). "A simple non-separable, non-stationary spatiotemporal model for ozone." *Environmental and Ecological Statistics*, 16, 4, 515–529.

Carter, C. K. and Kohn, R. (1994). "On Gibbs sampling for state space models." *Biometrika*, 81, 3, 541–553.

Carvalho, M. L. S. and Natário, I. C. M. (2008). *Análise de Dados Espaciais*. Sociedade Portuguesa de Estatística.

Chib, S. and Greenberg, E. (1995). "Hierarchical analysis of SUR models with extensions to correlated serial errors and time-varying parameter models." *Journal of Econometrics*, 68, 2, 339–360.

Corradi, F. and Guagnano, G. (1993). "Missing data and forecasting in multivariate time series: An application of the common components dynamic linear model." *Journal of the Italian Statistical Society*, 2, 2, 193–211.

Costa, A. C. C. (2011). "Modelos dinâmicos hierárquicos espaço-temporais para dados na família exponencial." Master's thesis, Federal University of Rio de Janeiro.

Cressie, N. A. C. (1993). *Statistics for Spatial Data*. John Wiley & Sons.

Damian, D., Sampson, P. D., and Guttorp, P. (2001). "Bayesian estimation of semi-parametric non-stationary spatial covariance structures." *Environmetrics*, 12, 2, 161–178.

— (2003). "Variance modeling for nonstationary spatial processes with temporal replications." *Journal of Geophysical Research: Atmospheres*, 108, D24.

Dawid, A. P. (1981). "Some matrix-variate distribution theory: Notational considerations and a Bayesian application." *Biometrika*, 68, 1, 265–274.

De Iaco, S., Posa, D., Cappello, C., and Maggio, S. (2020). "On some characteristics of Gaussian covariance functions." *International Statistical Review*, 1–18.

Diggle, P. J. and Ribeiro-Jr., P. J. (2007). *Model-Based Geostatistics*. Springer.

Eriksson, M. and Siska, P. P. (2000). "Understanding anisotropy computations." *Mathematical Geology*, 32, 6, 683–700.

Fonseca, T. C. O. and Steel, M. F. J. (2011). "A general class of nonseparable space–time covariance models." *Environmetrics*, 22, 2, 224–242.

Frühwirth-Schnatter, S. (1994). "Data augmentation and dynamic linear models." *Journal of Time Series Analysis*, 15, 2, 183–202.

Gaetan, C. and Guyon, X. (2010). *Spatial Statistics and Modeling*. Springer.

Gamerman, D., Ippoliti, L., and Valentini, P. (2022). "A dynamic structural equation approach to estimate the short-term effects of air pollution on human health." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 71, 1–31.

Gamerman, D. and Lopes, H. (2006). *Markov Chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. 2nd ed. Chapman & Hall/CRC.

Gelman, A., Carlin, J., Stern, H., Dunson, D., Vehtari, A., and Rubin, D. (2013). *Bayesian Data Analysis*. 3rd ed. Chapman & Hall/CRC.

Givens, G. H. and Hoeting, J. A. (2012). *Computational Statistics*. John Wiley & Sons.

Gómez-Rubio, V. (2020). *Bayesian Inference with INLA*. Chapman & Hall/CRC.

Gonçalves, E. and Mendes-Lopes, N. (2008). *Séries Temporais: Modelações Lineares e Não Lineares*. 2nd ed. Sociedade Portuguesa de Estatística.

Grange, S. K. (2019). *Technical note: saqgetr R package*.

Gupta, A. K. and Nagar, D. K. (2000). *Matrix Variate Distributions*. Chapman & Hall/CRC.

Guttman, I. and Menzefricke, U. (1983). "Bayesian inference in multivariate regression with missing observations on the response variables." *Journal of Business & Economic Statistics*, 1, 3, 239–248.

Guttorp, P. and Schmidt, A. M. (2013). "Covariance structure of spatial and spatiotemporal processes." *Wiley Interdisciplinary Reviews: Computational Statistics*, 5, 4, 279–287.

Hansen, N. C. (2009). "Modelos com coeficientes dinâmicos variando no espaço para dados da família exponencial." Master's thesis, Federal University of Rio de Janeiro.

Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., Wieser, E., Taylor, J., Berg, S., Smith, N. J., Kern, R., Picus, M., Hoyer, S., van Kerkwijk, M. H., Brett, M., Haldane, A., del Río, J. F., Wiebe, M., Peterson, P., Gérard-Marchant, P., Sheppard, K., Reddy, T., Weckesser, W., Abbasi, H., Gohlke, C., and Oliphant, T. E. (2020). "Array programming with NumPy." *Nature*, 585, 7825, 357–362.

House, D. and Keyser, J. C. (2016). *Foundations of Physically Based Modeling and Animation*. CRC Press.

Huerta, G., Sansó, B., and Stroud, J. R. (2004). "A spatiotemporal model for Mexico City ozone levels." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 53, 2, 231–248.

Isaaks, E. H. and Srivastava, R. M. (1989). *Applied Geostatistics*. Oxford University Press.

Jiménez, J. C. and Pereira, C. A. B. (2021). "Assessing dynamic effects on a Bayesian matrix-variate dynamic linear model: An application to task-based fMRI data analysis." *Computational Statistics & Data Analysis*, 163, 107297.

Jung, S., Schwartzman, A., and Groisser, D. (2015). "Scaling-rotation distance and interpolation of symmetric positive-definite matrices." *SIAM Journal on Matrix Analysis and Applications*, 36, 3, 1180–1201.

Kahle, D. and Wickham, H. (2013). "ggmap: Spatial Visualization with ggplot2." *The R Journal*, 5, 1, 144–161.

Kassambara, A. (2020). *ggpubr: 'ggplot2' Based Publication Ready Plots*. R package version 0.4.0.

Kitanidis, P. K. (1997). *Introduction to Geostatistics: Applications in Hydrogeology*. Cambridge University Press.

Lam, S. K., Pitrou, A., and Seibert, S. (2015). "Numba: A LLVM-Based Python JIT Compiler." In *Proceedings of the Second Workshop on the LLVM Compiler Infrastructure in HPC*, no. 7, 1–6. Association for Computing Machinery.

Landim, F. M. P. F. and Gamerman, D. (2000). "Dynamic hierarchical models: An extension to matrix-variate observations." *Computational Statistics & Data Analysis*, 35, 1, 11–42.

Lay, D. C., Lay, S. R., and McDonald, J. J. (2022). *Linear Algebra and Its Applications*. 6th ed. Pearson.

Li, Y., Shi, G., and Chen, Z. (2021). "Spatial and temporal distribution characteristics of ground-level nitrogen dioxide and ozone across China during 2015–2020." *Environmental Research Letters*, 16, 12, 124031.

Little, R. J. A. and Rubin, D. B. (2019). *Statistical Analysis with Missing Data*. 3rd ed. John Wiley & Sons.

Maity, A. and Sherman, M. (2012). "Testing for spatial isotropy under general designs." *Journal of Statistical Planning and Inference*, 142, 5, 1081–1091.

Mardia, K. V. and Goodall, C. R. (1993). "Spatial-temporal analysis of multivariate environmental monitoring data." In *Multivariate Environmental Statistics*, eds. G. P. Patil and C. R. Rao, chap. 16, 347–386. Elsevier.

Migon, H. S., Gamerman, D., and Louzada, F. (2014). *Statistical Inference: An Integrated Approach*. 2nd ed. Chapman & Hall/CRC.

Morales, F. E. C. (2010). "Modelos dinâmicos para deformação espacial." Ph.D. thesis, Federal University of Rio de Janeiro.

Morales, F. E. C., Gamerman, D., and Paez, M. S. (2013). "State space models with spatial deformation." *Environmental and Ecological Statistics*, 20, 2, 191–214.

Morales, F. E. C., Politis, D. N., Leskow, J., and Paez, M. S. (2022). "Student's-t process with spatial deformation for spatio-temporal data." *Statistical Methods & Applications*, 1–28.

Morales, F. E. C. and Vicini, L. (2020). "A non-homogeneous Poisson process geostatistical model with spatial deformation." *AStA Advances in Statistical Analysis*, 104, 3, 503–527.

Morettin, P. A. and Toloi, C. M. (2018). *Análise de Séries Temporais: Modelos Lineares Univariados*, vol. 1. 3rd ed. Blucher.

Moritz, S. and Bartz-Beielstein, T. (2017). "imputeTS: Time series missing value imputation in R." *The R Journal*, 9, 1, 207–218.

Murteira, B. J. F., Müller, D. A., and Turkman, K. F. (1993). *Análise de Sucessões Cronológicas*. McGraw-Hill.

Neal, R. M. (2003). "Slice sampling." *The Annals of Statistics*, 31, 3, 705–767.

— (2008). "Software for Slice Sampling." https://www.cs.toronto.edu/~radford/slice.software.html. Published in: 2008-03-17, Accessed in: 2022-04-10.

Paez, M. S. (2004). "Análise de modelos para a estimação e previsão de processos espaço-temporais." Ph.D. thesis, Federal University of Rio de Janeiro.

Paez, M. S. and Gamerman, D. (2013). "Hierarchical dynamic models." In *The SAGE Handbook of Multilevel Modeling*, eds. M. A. Scott, J. S. Simonoff, and B. D. Marx, chap. 19, 335–355. SAGE.

Paez, M. S., Gamerman, D., Landim, F. M. P. F., and Gonzales, E. S. (2008). "Spatially varying dynamic coefficient models." *Journal of Statistical Planning and Inference*, 138, 4, 1038–1058.

Paulino, C. D. M., Turkman, M. A. A., Murteira, B. J. F., and Silva, G. L. (2018). *Estatística Bayesiana*. 2nd ed. Fundação Calouste Gulbenkian.

Perrin, O. and Meiring, W. (1999). "Identifiability for non-stationary spatial structure." *Journal of Applied Probability*, 36, 4, 1244–1250.

Petris, G., Petrone, S., and Campagnoli, P. (2009). *Dynamic Linear Models with R*. Springer.

Plummer, M., Best, N., Cowles, K., and Vines, K. (2006). "CODA: Convergence diagnosis and output analysis for MCMC." *R News*, 6, 1, 7–11.

Press, S. J. (2005). *Applied Multivariate Analysis: Using Bayesian and Frequentist Methods of Inference*. 2nd ed. Dover Publications.

Quintana, J. M. (1987). "Multivariate Bayesian forecasting models." Ph.D. thesis, University of Warwick.

Quintana, M. S. B. (2022). "Bayesian modeling for spatial point process with nonstationary covariance structure via spatial deformations." Ph.D. thesis, Federal University of Rio de Janeiro.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Rostami-Tabar, B. and Rendon-Sanchez, J. F. (2021). "Forecasting COVID-19 daily cases using phone call data." *Applied Soft Computing*, 100, 106932.

Salvador, M., Gallizo, J. L., and Gargallo, P. (2004). "Bayesian inference in a matrix normal dynamic linear model with unknown covariance matrices." *Statistics*, 38, 4, 307–335.

Sampson, P. and Meiring, W. (2014). "Nonstationary Spatial Covariance Modeling Through Spatial Deformation." https://www.stat.washington.edu/peter/PASI/PASI_nonstat1_deform.pdf. Talk in the Pan-American Advanced Study Institute on Spatio-Temporal Statistics, Búzios, Brazil.

Sampson, P. D. (2010). "Constructions for Nonstationary Spatial Processes." In *Handbook of Spatial Statistics*, eds. A. E. Gelfand, P. Diggle, P. Guttorp, and M. Fuentes, chap. 9, 119–130. Chapman & Hall/CRC.

Sampson, P. D., Damian, D., and Guttorp, P. (2001). "Advances in Modeling and Inference for Environmental Processes with Nonstationary Spatial Covariance." In *geoENV III — Geostatistics for Environmental Applications*, eds. P. Monestiez, D. Allard, and R. Froidevaux, 17–32. Springer.

Sampson, P. D. and Guttorp, P. (1992). "Nonparametric estimation of nonstationary spatial covariance structure." *Journal of the American Statistical Association*, 87, 417, 108–119.

Sansó, B. and Guenni, L. (1999). "Venezuelan rainfall data analysed by using a Bayesian space-time model." *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48, 3, 345–362.

Schlager, S. (2017). "Morpho and Rvcg – Shape Analysis in R." In *Statistical Shape and Deformation Analysis*, eds. G. Zheng, S. Li, and G. Szekely, 217–256. Academic Press.

Schmidt, A. M. and Gelfand, A. E. (2003). "A Bayesian coregionalization approach for multivariate pollutant data." *Journal of Geophysical Research: Atmospheres*, 108, D24.

Schmidt, A. M. and Guttorp, P. (2020). "Flexible spatial covariance functions." *Spatial Statistics*, 37, 100416.

Schmidt, A. M., Guttorp, P., and O'Hagan, A. (2011). "Considering covariates in the covariance structure of spatial processes." *Environmetrics*, 22, 4, 487–500.

Schmidt, A. M., Nobre, A. A., and Ferreira, G. S. (2002). "Alguns aspectos da modelagem de dados espacialmente referenciados." *Revista Brasileira de Estatística*, 63, 220, 59–88.

Schmidt, A. M. and O'Hagan, A. (2003). "Bayesian inference for non-stationary spatial covariance structure via spatial deformations." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65, 3, 743–758.

Schuenemeyer, J. H. and Drew, L. J. (2011). *Statistics for Earth and Environmental Scientists*. John Wiley & Sons.

Shen, Y. and Gelfand, A. (2019). "Exploring geometric anisotropy for point-referenced spatial data." *Spatial Statistics*, 32, 100370.

Slowikowski, K. (2021). *ggrepel: Automatically Position Non-Overlapping Text Labels with 'ggplot2'*. R package version 0.9.1.

Soares, A. (2006). *Geoestatística para as Ciências da Terra e do Ambiente*. 2nd ed. IST Press.

Spiegelhalter, D. J., Best, N. G., Carlin, B. P., and van der Linde, A. (2002). "Bayesian measures of model complexity and fit." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 4, 583–639.

Stroud, J., Müller, P., and Sansó, B. (2001). "Dynamic models for spatio-temporal data." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 63, 673–689.

Triantafyllopoulos, K. (2008). "Missing observation analysis for matrix-variate time series data." *Statistics & Probability Letters*, 78, 16, 2647–2653.

— (2021). *Bayesian Inference of State Space Models: Kalman Filtering and Beyond*. Springer.

Tsiko, R. G. (2015). "Bayesian spatial analysis of childhood diseases in Zimbabwe." *BMC Public Health*, 15, 842, 1–24.

Turkman, M. A. A., Paulino, C. D., and Müller, P. (2019). *Computational Bayesian Statistics: An Introduction*. Cambridge University Press.

Twiss, R. J. and Moores, E. (2007). *Structural Geology*. 2nd ed. W. H. Freeman and Company.

van Lieshout, M. N. M. (2019). *Theory of Spatial Statistics: A Concise Introduction*. 1st ed. Chapman & Hall/CRC.

van Rossum, G. and Drake, F. L. (2011). *The Python Language Reference Manual*. Network Theory Ltd.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). "SciPy 1.0: Fundamental algorithms for scientific computing in Python." *Nature Methods*, 17, 261–272.

Wackernagel, H. (2003). *Multivariate Geostatistics: An Introduction with Applications*. 3rd ed. Springer.

West, M. and Harrison, J. (1997). *Bayesian Forecasting and Dynamic Models*. 2nd ed. Springer.

Wickham, H. (2016). *ggplot2: Elegant Graphics for Data Analysis*. 2nd ed. Springer.

Winkler, R. L. (1972). "A decision-theoretic approach to interval estimation." *Journal of the American Statistical Association*, 67, 337, 187–191.

Zhu, H. and Zhang, L. M. (2013). "Characterizing geotechnical anisotropic spatial variations using random field theory." *Canadian Geotechnical Journal*, 50, 7, 723–734.

Zimmerman, D. L. (1993). "Another look at anisotropy in geostatistics." *Mathematical Geology*, 25, 4, 453–470.