POLITECNICO
MILANO 1863

Solving Large Mixed-Effects Models in Spatial Regression with PDE Regularization

M.Sc. Thesis of
Martino Ischia

Supervisor:
Prof. Laura Sangalli

Co-supervisor:
Dr. Eleonora Arnone

Programme:
Mathematical Engineering

Department of Mathematics
Politecnico di Milano

Martino Ischia
Solving Large Mixed-Effects Models in Spatial Regression with PDE Regularization

Version 2.0

To my parents and to Qian.

Contents

List of Figures

Abstract

It is proposed an iterative solver for large linear systems, and it is implemented in R library `fdaPDE`. The equations originate from a functional minimization problem that is discretized with finite elements. In fact, in such context of mixed-effects models, each statistical unit increases the size of the system by $2N_\tau$, twice the number of the finite elements degrees of freedom.
To make computing a solution possible, at each iteration are solved m linear systems of dimensions $(2N_\tau)^2$, m being the number of statistical unit, instead of one of dimensions $(2mN_\tau)^2$.
A simulation study is carried out and the proposed method is finally applied to fMRI data in the cerebral cortex, allowing to largely increase the number of patients involved in the study, compared to existing ones.

Sommario

In questo lavoro viene proposto un algoritmo iterativo per sistemi lineari di grandi dimensioni e viene implementato nella libreria di R `fdaPDE`. Le equazioni originano da un problema di minimizzazione di un problema funzionale che viene discretizzato attraverso gli elementi finiti. Infatti, nel contesto di un modello a effetti misti, ogni unità statistica aumenta la dimensione del sistema di $2N_\tau$, due volte il numero di gradi di libertà degli elementi finiti.
Per rendere possibile il calcolo di una soluzione, ad ogni iterazione vengono risolti m sistemi lineari di dimensioni $(2N_\tau)^2$, con m il numero di unità statistiche, al posto di uno di dimensioni $(2mN_\tau)^2$.
Vengono condotte delle simulazioni al calcolatore e il modello proposto viene infine applicato a dati di risonanza magnetica funzionale nella corteccia cerebrale, permettendo di aumentare il numero di pazienti inclusi nello studio, rispetto agli studi esistenti.

1. Synopsis

This thesis is organized as follows.
Chapter 2 introduces the spatial regression with PDE regularization models and and mixed-effects models. Chapter 3 describes the iterative method proposed for solving problems with a large number of statistical units or in general large matrices generated by this type of models. Chapter 4 describes the statistical properties of the estimated parameters. Chapter 5 illustrates the simulations that have been performed to validate the proposed method. Chapter 6 shows an application of these models to several patients' fMRI data. Chapter 7 summarizes the results of this work.

2. Introduction

2.1. Spatial regression with partial differential equation regularization

This work naturally stems from the studies in the research field called *spatial regression with partial differential equation regularization*, abbreviated in the following as SR-PDE. SR-PDE constitutes a family of models which has been, and still is, under study beginning from 2013 in Sangalli, Ramsay, and Ramsay [SRR13].

SR-PDE effectively combines several branches of mathematics and statistics modeling. For a thorough analysis of its derivation and the enhancements it brings to the existing literature see [San21], I will limit myself to a brief introduction.
SR-PDE models feature the minimization of a functional composed of two terms: a classical least-square term, with the goal of estimating a vector of unknown regression coefficients (in case of presence of covariates), and a regularization term, estimating an unknown deterministic spatial field.
The spatial field contributes to the functional by integration over the spatial domain of the square of a differential operator, most commonly the Laplacian operator. By properties of the Laplace operator, this choice induces an isotropic and stationary smoothing effect on the unknown spatial field, meaning equal in every direction and independent on the position. Other differential operators considered in this work are generic second order differential operators, that come from the physics of the problem or from some preexisting knowledge about the phenomenon under study.

The inclusion of partial differential equations in the statistical model brings many advantages, like the aforementioned ability to include problem-specific knowledge, dealing with boundary conditions, etc., but makes computations more cumbersome.

2.2. Mixed-effects models

In a situation where observed data possess a natural grouping structure, mixed-effects models are often utilized. They combine fixed effects, meaning that a set of covariates have regression coefficients shared among all groups, and random effects, meaning the remaining set of covariates have regression coefficients varying along each group.

Given our interest in spatial data analysis, a typical situation in which mixed-models are used is in clinical data, where data of several patients are measured in the same area of the body. For example in chapter 6 Applications, we will see a mixed-effects model applied to human brain data, collected on a set of patients who undertook functional

Magnetic Resonance Imaging (fMRI).
We will denote as statistical unit the entity being observed that induces the grouping structure: in the just mentioned clinical example, the statistical units are the different patients being observed.

We are therefore ready to present, similarly as in [Kim20], the SR-PDE mixed-effects model.

2.3. Generic SR-PDE mixed-effects model

Consider m statistical units. To the i-th unit, with i varying from 1 to m, corresponds a unique spatial domain Ω_i, on which we observe n_i data at different positions $\mathbf{p}_{ij}$. Index j is therefore varying from 1 to n_i, for given statistical unit i.
The variable of interest z, observed at $\mathbf{p}_{ij}$, is modeled as

$$z_{ij} = \boldsymbol{w}_{ij}^\top \boldsymbol{\beta} + \boldsymbol{v}_{ij}^\top \mathbf{b}_i + f_i(\mathbf{p}_{ij}) + \epsilon_{ij}. \tag{2.1}$$

The notation used represents the following quantities:

- $\boldsymbol{w}_{ij}$ is the vector of fixed effects covariates for the observation z_{ij};
- $\boldsymbol{\beta}$ is the vector of regression coefficients for fixed effects;
- $\boldsymbol{v}_{ij}$ is the vector of random effects covariates for the observation z_{ij};
- $\mathbf{b}_i$ is the vector of regression coefficients of the random effects for unit i;
- f_i is the unknown deterministic field defined on domain Ω_i;
- ϵ_{ij}, for every i and j, are the random noise or errors, considered as the realizations of independent identically distributed (i.i.d.) random variables, with mean 0 and variance σ^2.

We will assume that the random effect $\mathbf{b}_i$ has average 0

$$\sum_{i=1}^{m} \mathbf{b}_i = 0. \tag{2.2}$$

In fact, if the average was different from 0, we would simply split the contribution of those covariates into two, a fixed effect one and a random effect covariate with 0 average across groups (more on this in section 2.4, Reparametrizing the model).
Let also N be equal to $\sum_{i=1}^{m} n_i$. For every i, j equation 2.1 constitutes a system of N equations, that is expressed concisely in algebraic form in the following two ways:

$$\begin{cases} \mathbf{z}_i = W_i\boldsymbol{\beta} + V_i\mathbf{b}_i + \mathbf{f}_i + \boldsymbol{\epsilon}_i \qquad i = 1 \dots m \\ \sum_{i=1}^{m} \mathbf{b}_i = 0 \end{cases} \tag{2.3}$$

and

$$\mathbf{z} = W\boldsymbol{\beta} + V\mathbf{b} + \mathbf{f}_N + \boldsymbol{\epsilon}, \tag{2.4}$$

in which the corresponding terms in the two equations have the following meanings:

- $z_i \in \mathbb{R}^{n_i}$ is the vector of observed data for unit i and z is the vector belonging to $\mathbb{R}^N$ obtained by concatenating the m vectors z_i;
- W_i is the matrix whose element $w_{j,k}$ is the k-th element of previously defined vector $\boldsymbol{w}_{ij}$, $(\boldsymbol{w}_{ij})_k$, whereas W is a concatenation as well

$$W = \begin{bmatrix} W_1 \\ \vdots \\ W_m \end{bmatrix}; \tag{2.5}$$

- V_i is the matrix whose element $v_{j,k}$ is the k-th element of previously defined vector $\boldsymbol{v}_{ij}$, $(\boldsymbol{v}_{ij})_k$;
- when moving to the N system of equations, we include constraint 2.2 by defining $\mathbf{b}$ as the concatenation of $m - 1$ vectors $\mathbf{b}_i$, omitting the last one

$$\mathbf{b} = \begin{bmatrix} \mathbf{b}_1 \\ \vdots \\ \mathbf{b}_{m-1} \end{bmatrix} \tag{2.6}$$

and defining V in the following manner:

$$V = \begin{bmatrix} V_1 & 0 & 0 & 0 \\ 0 & V_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & V_{m-1} \\ -V_m & -V_m & -V_m & -V_m \end{bmatrix}; \tag{2.7}$$

- $\mathbf{f}_i \in \mathbb{R}^{n_i}$ is the vector of observed data for unit i and $\mathbf{f}_N$ is the vector belonging to $\mathbb{R}^N$ obtained by concatenating the m vectors $\mathbf{f}_i$;
- $\boldsymbol{\epsilon}_i \in \mathbb{R}^{n_i}$ is the vector of errors for unit i and $\boldsymbol{\epsilon}_N$ is the vector belonging to $\mathbb{R}^N$ obtained by concatenating the m vectors $\boldsymbol{\epsilon}_i$.

2.4. Reparametrizing the model

The N system of equations 2.4 is the model written in an unconstrained form, where the constraint 2.2 has been "injected" into the design matrix. We also call it *official parametrization* of the mixed-effects model.

As described in [Kim20] we prefer to adopt a slightly different approach, mainly for computational efficiency reasons. In the following we will assume that all covariates contribute to the fixed effect part of the model, whereas a subset of them will contribute to the random effect part. We will refer to this version of the model as the *implementation* one.

Implementation version is expressed by the following system:

$$z = W'\beta' + V'b' + f_N + \epsilon, \tag{2.8}$$

where we have used the following new quantities:

- W', similarly as before, is concatenation of W'_i matrices, where W'_i is composed of the covariates related only to fixed effects, meaning with no unit-specific contribution.
$$W' = \begin{bmatrix} W'_1 \\ W'_2 \\ \vdots \\ W'_m \end{bmatrix}; \tag{2.9}$$
- β' is the coefficient relative to covariates for fixed effects, as just described;
- V' is composed of V'_i matrices, which are the matrices of covariates having also a random effect. Unlike previous V, V' does not include the part deriving from constraint 2.2:
$$V' = \begin{bmatrix} V'_1 & 0 & 0 & 0 \\ 0 & V'_2 & 0 & 0 \\ \vdots & 0 & \ddots & \vdots \\ 0 & 0 & 0 & V'_m \end{bmatrix}; \tag{2.10}$$
- b' is the regression coefficient relative to the covariates with unit-specific effect,
$$b' = \begin{bmatrix} b'_1 \\ \vdots \\ b'_m \end{bmatrix}; \tag{2.11}$$

To properly refer to sizes of vectors and matrices defined before, we use the following notation. We indicate with q the number of covariates, and with p the number of covariates with unit-specific interest. Therefore $q - p$ is the number of covariates considered for their fixed effect contribution only.

Vector b' belongs to $\mathbb{R}^{mp}$, β' to $\mathbb{R}^{q-p}$ and we can relate it to previously defined vectors b, β ($\in \mathbb{R}^{m(p-1)}, \mathbb{R}^q$ respectively), by defining $\beta^* \in \mathbb{R}^p$, average of random effects coefficients

$$\beta^* = \frac{\sum_{i=1}^{m} b'_i}{m}. \tag{2.12}$$

In this way β is just the concatenation of β' and β^* while for b holds the following:

$$b = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_{m-1} \end{bmatrix} = \begin{bmatrix} b'_1 - \beta^* \\ b'_2 - \beta^* \\ \vdots \\ b'_{m-1} - \beta^* \end{bmatrix}. \tag{2.13}$$

For what concerns the sizes of the matrices defined before, we deduce that $W_i \in \mathbb{R}^{n_i \times q}$, $W \in \mathbb{R}^{N \times q}$, $V_i \in \mathbb{R}^{n_i \times p}$,$V \in \mathbb{R}^{N \times p}$, $W'_i \in \mathbb{R}^{n_i \times (q-p)}$, $W' \in \mathbb{R}^{N \times (q-p)}$, $V'_i \in \mathbb{R}^{n_i \times p}$,$V' \in \mathbb{R}^{N \times p}$.

2.5. Estimation problem

Similarly to what happens with every SR-PDE model, we estimate the unknown quantities of the model by solving, with the necessary approximations, the following minimization problem:

$$\text{Find} \quad \underset{\beta', b', f_1, \ldots, f_m}{\arg\min} \quad J_{\Omega_i, \lambda}(\beta', b', f_1, \ldots, f_m) \tag{2.14}$$

where the functional $J_{\Omega_i, \lambda}$ is defined as

$$J_{\Omega_i, \lambda}(\beta', b', f_1, \ldots, f_m) = \|z - W'\beta' - V'b' - f_N\|^2 + \lambda \sum_{i=1}^{m} \int_{\Omega_i} \Delta f_i(p)^2 \, d\Omega_i, \tag{2.15}$$

or equivalently, in a more lengthy expression:

$$J_{\Omega_i, \lambda}\left(\beta', b'_1, \ldots, b'_m, f_1, \ldots, f_m\right) = \sum_{i=1}^{m} \left(\sum_{j=1}^{n_i} \left(z_{ij} - {w'_{ij}}^T \beta' - {v'_{ij}}^T b'_i - f_i(p_{ij}) \right)^2 + \lambda \int_{\Omega_i} \Delta f_i(p)^2 \, d\Omega_i \right). \tag{2.16}$$

For simplicity, we have considered the Laplacian as differential operator, but more general choices are possible.

It shall be noticed that the unknown field must not satisfy the differential equation but contributes to the functional with the square of its misfit from the equation itself — besides its contribution in terms of distance from observed data.

We now introduce the notation necessary to tackle problem 2.14, and characterize the solution defined on suitable spaces. Assuming W'_i and V'_i for $i = 1, \ldots, m$ full rank, we define the following matrices:

$$X = \begin{bmatrix} W'_1 & V'_1 & 0 & \ldots & 0 & 0 \\ W'_2 & 0 & V'_2 & \ldots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ W'_{m-1} & 0 & 0 & \ldots & V'_{m-1} & 0 \\ W'_m & 0 & 0 & \ldots & 0 & V'_m \end{bmatrix} \tag{2.17}$$

$$H = X\left(X^T X\right)^{-1} X^T \tag{2.18}$$

$$Q = I_N - H \tag{2.19}$$

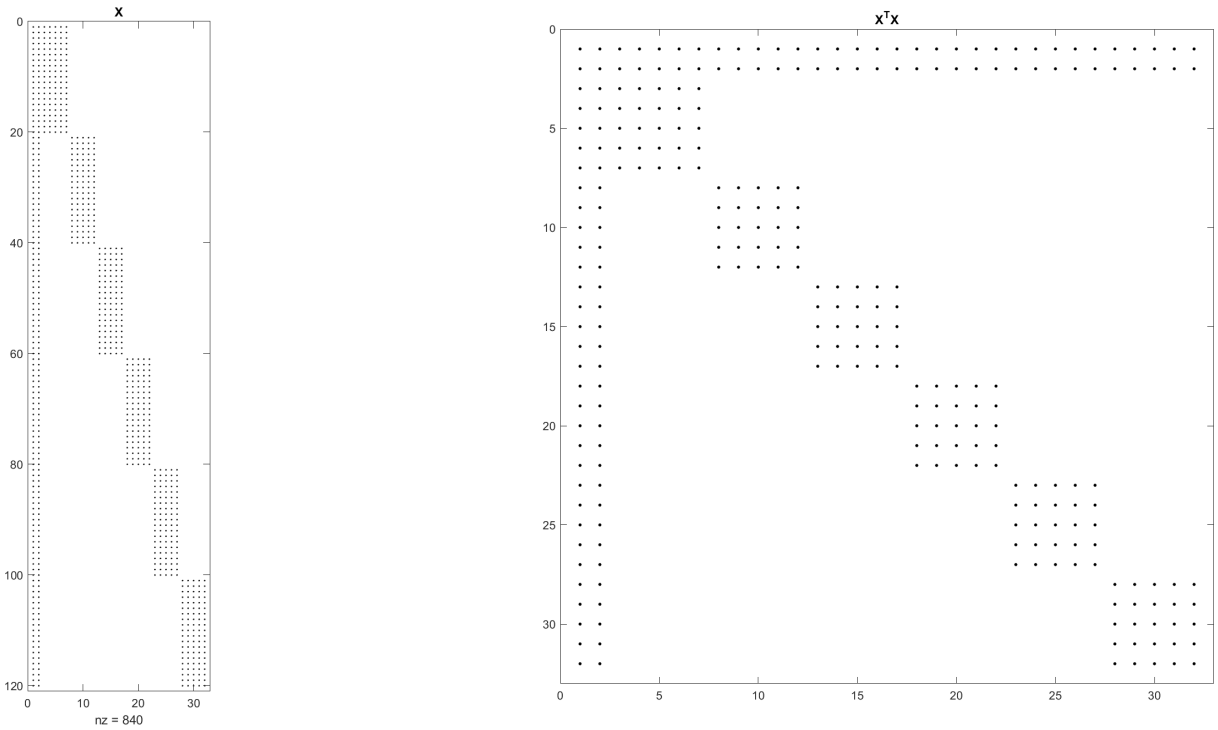


Figure 2.1.: *On the left, sparsity pattern of an arbitrary matrix X as defined in 2.17. On the right, correspondent pattern of symmetric matrix* $X^T X$. *Inverse of this matrix is in general dense but does not get computed due to the high condition number of* $X^T X$.

with $X \in \mathbb{R}^{N \times (mp+q-p)}$, H and $Q \in \mathbb{R}^{N \times N}$ are the matrices that project a vector, respectively, onto the subspace spanned by the columns of X and onto its orthogonal complement with respect to $\mathbb{R}^N$. Notice that matrix $X^T X$ exhibits the pattern in figure 2.1, but as it happens with the normal equations, in general this type of matrices have large condition number. An ..... factorization is stored in our implementation for performing several times this matrix multiplication.

Define also the vector of coefficients $\boldsymbol{\nu} = (\boldsymbol{\beta}', \mathbf{b}_1', \ldots, \mathbf{b}_m') \in \mathbb{R}^{mp+q-p}$.
Given these definitions, we can write the mixed-effects model in *implementation* version with the formula

$$\boldsymbol{z} = X\boldsymbol{\nu} + \mathbf{f}_N + \boldsymbol{\epsilon}, \tag{2.20}$$

which separates the two components of our model, the parametric one and the non-parametric. The estimation functional 2.15 can then be expressed as

$$J_{\Omega_i,\lambda}(\boldsymbol{\nu}, f_1, \ldots, f_m) = \|\boldsymbol{z} - X\boldsymbol{\nu} - \mathbf{f}_N\|^2 + \lambda \sum_{i=1}^{m} \int_{\Omega_i} \Delta f_i(\mathbf{p})^2 \, d\Omega_i, \tag{2.21}$$

Next section describes the properties of a possible minimizer $\left(\hat{\boldsymbol{\nu}}, \hat{f}_1, \ldots, \hat{f}_m\right)$ of functional 2.21.

2.6. Characterization of the solution

Assuming that for each unit i, spatial field f_i belongs to $\mathcal{H}^2(\Omega_i)$, Sobolev space of functions whose first and second derivative are in $\mathcal{L}^2(\Omega_i)$, the functional 2.21 is well-defined.
We also make the assumption, not strictly necessary, of imposing on every field f_i homogeneous boundary condition of Neumann type, meaning the normal derivative on the boundary is null almost everywhere. Interesting applications of different type of boundary conditions in the context of SR-PDE models are treated in [Azz+14].

The field $f = (f_1 \dots f_m)$ is therefore naturally set in the Hilbert space

$$\mathcal{V} = \bigoplus_{i=1}^{m} \mathcal{H}^2_{n0}(\Omega_i) \tag{2.22}$$

where $\mathcal{H}^2_{n0}(\Omega_i)$ is the set $\left\{g \in \mathcal{H}^2(\Omega_i) \mid \nabla g \cdot \mathbf{n} = 0 \text{ a.e. on } \partial\Omega_i\right\}$.

For the estimation problem introduced before, it holds the following:

Theorem 2.1. *With* $\boldsymbol{\nu} \in \mathbb{R}^{mp+q-p}$ *and* $f = (f_1 \dots f_m) \in \mathcal{V}$*, for functional* $J_{\Omega_i,\lambda}(\boldsymbol{\nu}, f)$ *there exists one and only one minimizer* $\left(\hat{\boldsymbol{\nu}}, \hat{f}\right)$.

By denoting also with $\boldsymbol{\varphi}_N$ the vector obtained by concatenation of the evaluations of a generic element φ belonging to functional space $\mathcal{V}$, at locations n_i for every i, we have the following characterization for the solution $\left(\hat{\boldsymbol{\nu}}, \hat{f}\right) = \left(\hat{\boldsymbol{\nu}}, \hat{f}_1, \dots, \hat{f}_m\right)$ of the estimation problem:

$$\begin{cases} \boldsymbol{\varphi}_N^T Q \hat{\mathbf{f}}_N + \lambda \sum_{i=1}^{m} \int_{\Omega_i} \Delta\varphi_i \Delta\hat{f}_i \, d\Omega_i = \boldsymbol{\varphi}_N^T Q \boldsymbol{z} \quad \forall \varphi \in \mathcal{V} \\ \hat{\boldsymbol{\nu}} = \left(X^T X\right)^{-1} X^T (\boldsymbol{z} - \hat{\mathbf{f}}_N) \end{cases} . \tag{2.23}$$

First of previous equations can be equivalently rewritten into two equations. With the introduction of an auxiliary function $\hat{g}$, whose vector components are $-\Delta\hat{f}_i$, we write system 2.23 as

$$\begin{cases} \boldsymbol{\varphi}_N^T Q \hat{\mathbf{f}}_N + \lambda \sum_{i=1}^{m} \int_{\Omega_i} \nabla\varphi_i \nabla g_i \, d\Omega_i = \boldsymbol{\varphi}_N^T Q \boldsymbol{z} \\ \sum_{i=1}^{m} \left(\int_{\Omega_i} \eta_i \hat{g}_i \, d\Omega_i - \int_{\Omega_i} \nabla\eta_i \cdot \nabla\hat{f}_i \, d\Omega_i \right) = 0 \\ \hat{\boldsymbol{\nu}} = \left(X^T X\right)^{-1} X^T (\boldsymbol{z} - \hat{\mathbf{f}}_N) \end{cases} , \tag{2.24}$$

valid for every couple (φ, η) belonging to appropriate functional spaces (essentially, we can work here with Sobolev spaces of degree one).
This form of the problem is the most suitable for the use of the finite element method to compute the desired estimation.

2.7. Numerical solution

In the library `fdaPDE` we solve the estimation problem by mean of finite elements (for detailed description see e.g. [Qua23]). We generate a triangular mesh, by partitioning

domains Ω_i with a regular triangulation. Domain Ω_i is therefore approximated as $\Omega_i^{\tau_i}$, union of the triangles.

On the approximated domains we define Sobolev spaces, $\mathcal{H}_i^1\left(\Omega_i^{\tau_i}\right)$, for every i, and relative subspaces, $\mathcal{V}_i^1\left(\Omega_i^{\tau_i}\right)$, *i.e.* the finite element space of degree one on triangulation τ_i: in this work we will limit ourselves to Lagrangian finite elements of degree one with nodes in the vertices of the triangles.

Having defined this spaces, we move to the discrete counterpart of problem 2.24. The equations remain the same if not for the domain Ω_i , approximated as $\Omega_i^{\tau_i}$. But as we change functional spaces from infinite dimension to finite dimension:

- we can write functions as linear (finite) combinations of finite element basis; we denote as $\boldsymbol{\psi}_i$ or $\boldsymbol{\psi}^{(i)}$ if necessary, the vector of fields that constitute a basis for $\mathcal{V}_i^1\left(\Omega_i^{\tau_i}\right)$;
- we can compute integrals by numerical techniques, see [Qua23] for technical details;
- thanks to previous two points, problem 2.24 is equivalently written in algebraic form as a linear system of equations.

The following matrices definitions are used:

$$\Psi_i = \begin{bmatrix} \psi_1^{(i)}(\mathbf{p}_{i1}) & \dots & \psi_{N_{\tau_i}}^{(i)}(\mathbf{p}_{i1}) \\ \vdots & \ddots & \vdots \\ \psi_1^{(i)}(\mathbf{p}_{in_i}) & \dots & \psi_{N_{\tau_i}}^{(i)}(\mathbf{p}_{in_i}) \end{bmatrix}. \tag{2.25}$$

Ψ_i belongs to $\mathbb{R}^{n_i \times N_{\tau_i}}$, where we define N_{τ_i} as the number of basis functions deriving from triangulation τ_i relative to Ω_i. Given Neumann boundary conditions, this is also the number of nodes of the i-th mesh.

$$R_{0i} = \int_{\Omega_{\tau_i}} \boldsymbol{\psi}_i \boldsymbol{\psi}_i^T \, d\Omega_{\tau_i}. \tag{2.26}$$

R_{0i} lies in $\mathbb{R}^{N_{\tau_i} \times N_{\tau_i}}$.

$$R_{1i} = \int_{\Omega_{\tau_i}} (\nabla \boldsymbol{\psi}_i)^T \nabla \boldsymbol{\psi}_i \, d\Omega_{\tau_i}. \tag{2.27}$$

R_{1i} lies in $\mathbb{R}^{N_{\tau_i} \times N_{\tau_i}}$ as well, with $\nabla \boldsymbol{\psi}_i$ being the matrix whose element in m-th row and n-th column is $\frac{\partial (\boldsymbol{\psi}_i)_n}{\partial x_m}$.

Before we used the operator mapping a generic element φ, belonging to functional space $\mathcal{V}$, to $\boldsymbol{\varphi}_N$, the vector obtained by concatenation of the evaluations of φ, at locations n_i for every i. Define $N_\tau = \sum_{i=1}^m N_{\tau_i}$ the total number of degrees of freedom for all the statistical units. By defining $\tilde{\Psi}, \in \mathbb{R}^{N \times N_\tau}$ as

$$\tilde{\Psi} = \begin{bmatrix} \Psi_1 & & 0 \\ & \ddots & \\ 0 & & \Psi_m \end{bmatrix}, \tag{2.28}$$

the following identity shows the role of Ψ_i matrices:

$$\boldsymbol{\varphi}_N = \tilde{\Psi}\boldsymbol{\varphi} \tag{2.29}$$

where $\boldsymbol{\varphi}$ is the coefficient vector of expansion of φ with respect to the considered finite element basis.

Analogously as $\tilde{\Psi}$, we define the tensorised versions of R_0 and R_1 ($\in \mathbb{R}^{N_\tau \times N_\tau}$) matrices as:

$$\tilde{R_0} = \begin{bmatrix} R_{01} & & 0 \\ & \ddots & \\ 0 & & R_{0m} \end{bmatrix}, \tag{2.30}$$

$$\tilde{R_1} = \begin{bmatrix} R_{11} & & 0 \\ & \ddots & \\ 0 & & R_{1m} \end{bmatrix}. \tag{2.31}$$

By mean of simple algebraic manipulations, we express equivalently the discrete counterpart of problem 2.24 with the following linear system of equations:

$$\begin{bmatrix} \tilde{\Psi}^T Q \tilde{\Psi} & -\lambda \tilde{R}_1^T \\ -\lambda \tilde{R}_1 & -\lambda \tilde{R}_0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} \tilde{\Psi}^T Q \mathbf{z} \\ 0 \end{bmatrix} \tag{2.32}$$

together with the least square equation for parameter $\boldsymbol{\nu}$, which can be rewritten as

$$\hat{\boldsymbol{\nu}} = \left(X^T X\right)^{-1} X^T (\mathbf{z} - \tilde{\Psi}\hat{\mathbf{f}}) \tag{2.33}$$

System 2.32 has dimensions $(2N_\tau)^2$. We call system 2.32 monolithic because it might be tough to solve it by standard numerical techniques. In fact N_τ might be large, for both the presence of a large number of statistical units m or thanks to a large number of nodes N_{τ_i} for each unit.

Therefore, aim of this work is to avoid the solution of such high dimension linear system, in favour of splitting it into many systems (m) of lower dimension.

In case the dimensions of the monolithic system are treatable, the Woodbury decomposition formula, described in appendix A.1, can be used to speed up the computation of the solution for different values of λ. The decomposition for system 2.32 is analogous to the one described in the appendix, with

$$E = \begin{bmatrix} \tilde{\Psi}^T \tilde{\Psi} & -\lambda \tilde{R}_1^T \\ -\lambda \tilde{R}_1 & -\lambda \tilde{R}_0 \end{bmatrix} \quad U = \begin{bmatrix} \tilde{\Psi}^T X \\ 0 \end{bmatrix} \tag{2.34}$$

$$C = -\left[\left(X^T X\right)^{-1}\right] \quad V = U^T;$$

again, despite E being of same dimension as the monolithic system, it has the property of being fully sparse.

This was the approach used by Kim in her work [Kim20]. The monolithic system, as data was coming from hundreds of units, quickly failed to scale at such dimension. In

next section we present an iterative method that, under some hypothesis discussed in section 5.1, does not require storing matrices such as $\tilde{R}_0$, $\tilde{R}_1$, $\tilde{\Psi}$. Despite this hypothesis being quite restrictive, for example the domain Ω_i being the same across units, in many applications they can be relaxed. As an example, when applying this model on brain data, we will see in chapter 6 that different brain surfaces for each patient get approximated as a common standardized surface, making the iterative method particularly effective.

3. Solving large linear systems for the mixed-effects model

3.1. Iterative methods

Following the ideas stemmed from the spatio-temporal regression in Pollini and Ponti [PP13] and Massardi and Spaziani [MS21], we consider an iterative scheme. At each step k the scheme computes an approximate solution $(\hat{\mathbf{f}}^k, \hat{\mathbf{g}}^k)$ of the monolithic system by solving a single-unit problem for each statistical unit. After initializing the needed quantities, we execute iterations until we satisfy a stopping criterion.
Denote with J^k the functional $J_{\Omega_i,\lambda}$ in 2.15, evaluated in the estimated solution at the k-th iteration:

$$J^k = J_{\Omega_i,\lambda}\left(\hat{\boldsymbol{\beta}}'^k, \hat{\mathbf{b}}_1'^k, \ldots, \hat{\mathbf{b}}_m'^k, \hat{f}_1^k, \ldots, \hat{f}_m^k\right). \tag{3.1}$$

The algorithm stops with two possible criteria:

- A maximum number of iterations is reached;
- The following two conditions hold:

  a) J_k has reached stagnation, that is the relative increment $\left(J^k - J^{k-1}\right)/J^k$ is below a certain threshold. In the code such threshold is an input parameter, it was set to 10^{-6}. The term $\int_{\Omega_i} \Delta f_i\,(\mathbf{p})^2\, d\Omega_i$ of the functional 2.15 is easily computed by exploiting the expansion into the finite element basis functions: it is equal to $\hat{\mathbf{g}}_i^T R_0 \hat{\mathbf{g}}_i$, where $\hat{\mathbf{g}}_i$ is the sub-vector of $\hat{\mathbf{g}}$ corresponding to unit i.

  b) The estimated solution is close to the exact solution of system 2.32. This condition is verified by checking that the residual, normalized by the Euclidean norm of the right-hand side, is below a certain threshold (this is another input parameter, 10^{-6} was used).

The details of the method that was implemented are described in the following sections.

3.1.1. A block diagonal approximation

Looking at the monolithic equation 2.32, the question of how to formulate an approximation of the term $\tilde{\Psi}^T Q \tilde{\Psi}$ naturally arises. In particular, a suitable block approximation allows to make the estimated field $\hat{f}_i$ of statistical unit i independent of the observations in the other units, for every unit i ($i = 1 \ldots m$). To this purpose, the following approximation is considered:

$$\tilde{\Psi}^T Q \tilde{\Psi} \simeq \Gamma := \begin{bmatrix} \Psi_1^T Q_1 \Psi_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \Psi_m^T Q_m \Psi_m \end{bmatrix}, \tag{3.2}$$

where Q_i indicates the i-th diagonal block of Q, of dimensions $n_i \times n_i$. By definition of Q, this is equal to $I_{n_i} - X_i \left(X^T X\right)^{-1} X_i^T$, with X_i being the submatrix of X with coefficients x_{jk}, with j belonging to $\{\sum_{l=0}^{i-1} n_l + 1, \ldots, \sum_{l=0}^{i} n_l\}$, k in $\{1, \ldots, q - p + mp\}$.
This choice naturally stems from the idea behind the iterative method as we will see in section 3.1.3, but also $\Psi^T Q_i \Psi$ is the i-th diagonal block of matrix $\tilde{\Psi}^T Q \tilde{\Psi}$. In fact, this is equal to

$$\tilde{\Psi}_i^T Q \tilde{\Psi}_i \tag{3.3}$$

where here we have indicated with $\tilde{\Psi}_i^T$, $\tilde{\Psi}_i$ the i-th block row submatrix, *c.f.* 2.28 with the definition of $\tilde{\Psi}$. By denoting with $Q_{i,j}$ the submatrix of Q naturally arising from the subdivision per statistical unit (recall that $Q \in N \times N$ with $N = \sum_{i=1}^{m} n_i$) and substituting we get

$$\begin{aligned} \tilde{\Psi}_i^T Q \tilde{\Psi}_i = \begin{bmatrix} 0 & \ldots & 0 & \Psi_i^T & 0 & \ldots & 0 \end{bmatrix} Q \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Psi_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \\ \begin{bmatrix} \Psi_i^T Q_{i,1} & \ldots \Psi_i^T Q_{i,m} \end{bmatrix} \begin{bmatrix} 0 \\ \vdots \\ 0 \\ \Psi_i \\ 0 \\ \vdots \\ 0 \end{bmatrix} = \Psi_i^T Q_i \Psi_i. \end{aligned} \tag{3.4}$$

3.1.2. Initialization

The initialization consists in finding a good guess $(\hat{\mathbf{f}}^0, \hat{\mathbf{g}}^0)$ to start the algorithm from; for this purpose, the following m problems are solved: for $i = 1, \ldots, m$ solve

$$\begin{bmatrix} \Psi_i^T Q_i \Psi_i & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}}_i^0 \\ \hat{\mathbf{g}}_i^0 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_i \\ 0 \end{bmatrix}, \tag{3.5}$$

where $\mathbf{u}_i$ is the subvector $\left(\tilde{\Psi}^T Q \boldsymbol{z}\right)_k$ such that $k = \sum_{l=1}^{l=i-1} + 1, \ldots, \sum_{l=1}^{l=i}$.

3.1.3. Iterations

The idea behind the iterative scheme is, having a guess of a solution $(\hat{\mathbf{f}}^{k-1}, \hat{\mathbf{g}}^{k-1})$, to compute a new guess $(\hat{\mathbf{f}}^{k}, \hat{\mathbf{g}}^{k})$ by replacing inside the monolithic system 2.32 the unknown coefficients relative to all but one unit, with their value computed at previous step, for every unit.

For example, the first n_i equations of the monolithic system 2.32 read

$$\begin{bmatrix} \Psi_1^T Q_{1,1} \Psi_1 & \Psi_1^T Q_{1,2} \Psi_2 & \dots & \Psi_1^T Q_{1,m} \Psi_m & -\lambda R_1^T \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}}_1 \\ \hat{\mathbf{f}}_2 \\ \vdots \\ \hat{\mathbf{f}}_m \\ \hat{\mathbf{g}}_1 \end{bmatrix} = \begin{bmatrix} \mathbf{u}_1 \end{bmatrix}, \tag{3.6}$$

where $Q_{i,j}$ indicates block of row i and column j of matrix Q, as seen in section 3.1.1. Substituting $\hat{\mathbf{f}}_j$ for $j \neq 1$ with $\hat{\mathbf{f}}_j^{k-1}$, taking previous step values to the right-hand side and generalizing, the iterative scheme reads: for $i = 1, \dots, m$ solve

$$\begin{bmatrix} \Psi_i^T Q_i \Psi_i & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{f}}_i^k \\ \hat{\mathbf{g}}_i^k \end{bmatrix} = \begin{bmatrix} \mathbf{r}_i \\ 0 \end{bmatrix} \tag{3.7}$$

where

$$\mathbf{r}_i = \mathbf{u}_i - \sum_{\substack{j=1 \\ j \neq i}}^{m} \Psi_i^T Q_{i,j} \Psi_j \hat{\mathbf{f}}_j^{k-1} \tag{3.8}$$

An estimate of $\boldsymbol{\nu}$, $\hat{\boldsymbol{\nu}}^i = (\hat{\boldsymbol{\beta}}'^i, \hat{\mathbf{b}}_1'^i, \dots, \hat{\mathbf{b}}_m'^i)$ has to be computed at each iteration for the estimation of the functional 2.15, according to equation 2.33.

3.1.4. The iterative method as a preconditioned Richardson method

Given a generic preconditioning matrix P, defining $\mathbf{r}_k = A\mathbf{x}_k - \mathbf{b}$ the residual of the linear system $A\mathbf{x} = \mathbf{b}$ at step k, Richardson method (*cf.*, for example, [Ger+14]) consists in solving (or rather trying to) the linear system iterating the following steps:

1. Solve $P\mathbf{z}_k = \mathbf{r}_k$
2. Compute the acceleration parameter α_k (for simplicity we use $\alpha_k = 1$)
3. Update the solution $\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{z}_k$
4. Update the residual $\mathbf{r}_{k+1} = \mathbf{r}_k - \alpha_k A\mathbf{z}_k$

The iterative scheme described in the previous section is a Richardson scheme with the following preconditioning matrix:

$$P = \begin{bmatrix} \Gamma & -\lambda \tilde{R}_1^T \\ -\lambda \tilde{R}_1 & -\lambda \tilde{R}_0 \end{bmatrix} \tag{3.9}$$

Solving a large linear system involving matrix P is indeed solving m independent linear systems of dimensions $2n \times 2n$.

Notice that we have approximated $\tilde{\Psi}^T Q \tilde{\Psi}$ by keeping its block-diagonal pattern. On the whole monolithic system 2.32 though, we are not performing a block diagonal approximation: we are in fact keeping the terms outside of the main diagonal involving $\tilde{R}_1$ matrices.

3.2. Generalized cross validation

Problem 2.32 is solved multiple times on a grid of λs. The best λ is then chosen according to a generalized cross-validation criterion. In particular, as in [San21], the minimum of the generalized cross-validation (GCV) parameter is used as a model selection criterion. The corresponding parameter in the mixed-effect model reads as follows:

$$GCV(\lambda) = \frac{1}{N\left(1 - (q - p + mp + \mathrm{tr}\,(S))/N\right)^2} \|z - \hat{z}\|^2 \tag{3.10}$$

where for q, m, p we follow the notation of section 2.4, and S indicates the smoothing matrix, which is the matrix that maps the observations vector z into the estimated spatial field $\hat{f}$ evaluated at the location of the observations ($\hat{f}_N = \tilde{\Psi}\hat{f} = Sz$). Its value is

$$S = \tilde{\Psi}\left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1\right)^{-1} \tilde{\Psi}^T Q, \tag{3.11}$$

and it stems as a Schur complement for system 2.32 with respect to the estimated field $\hat{f}$.

The computation of the GCV parameter 3.10 involves the expensive computation of the trace of the smoothing matrix S. Since for any two matrices A and B such that the product AB is defined, it holds that $\mathrm{tr}\,(AB) = \mathrm{tr}\,(BA)$, the trace of S can be expressed as the trace of

$$\tilde{S} = \left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1\right)^{-1} \tilde{\Psi}^T Q \tilde{\Psi}. \tag{3.12}$$

Several approaches are possible here for this computation.

1. In an exact way, through Woodbury decomposition. Woodbury decomposition, see A.1, can be exploited here too for the matrix inside the parenthesis. By choosing the following matrices

$$E = \tilde{\Psi}^T \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \quad U = \tilde{\Psi}^T X \tag{3.13}$$
$$C = -\left(X^T X\right)^{-1} \quad V = X^T \tilde{\Psi},$$

   E is a block diagonal matrix that allows relatively easy system solving. Other than E, with this approach we move the complex system to solve to the dimensions of $X^T X$, that are $q - p + mp$. One obtains that

$$\mathrm{tr}\left(\tilde{S}\right) = \mathrm{tr}\left(\left(E^{-1} - E^{-1}U\left(C^{-1} + VE^{-1}U\right)^{-1} VE^{-1}\right)\tilde{\Psi}^T Q \tilde{\Psi}\right). \tag{3.14}$$

Rearranging, substituting, defining $G = E^{-1}\tilde{\Psi}^T Q\tilde{\Psi}$, we get

$$\operatorname{tr}\left(\tilde{S}\right) = \operatorname{tr}(G) - \operatorname{tr}\left(E^{-1}\tilde{\Psi}^T X\left(X^T\tilde{\Psi}E^{-1}\tilde{\Psi}^T X - X^T X\right)^{-1} X^T\tilde{\Psi}G\right). \tag{3.15}$$

2. As an approximation, for convenience. In the library was first implemented a simplified method to compute the degrees of freedom. It follows a logic similar to the one of the iterative method. In the cases of chapter 5, the difference with the exact method was basically negligible, but no further analysis has been made.

   In this method we approximate $\tilde{S}$ in equation 3.12 by considering the block diagonal approximation of the group $\tilde{\Psi}^T Q\tilde{\Psi}$. As we have seen before, its k-th diagonal block is $\Psi_i^T Q_i \Psi_i$. Trace of $\tilde{S}$ can therefore be written as

$$\operatorname{tr}\left(\tilde{S}\right) = \sum_{i=1}^{m}\left(\Psi_i^T Q_i\Psi_i + \lambda R_1^T R_0^{-1} R_1\right)^{-1}\Psi_i^T Q_i\Psi_i. \tag{3.16}$$

3-4. As an estimation (possible both for method 1-2) deriving from the following result.

   Theorem 3.1 (Hutchinson)**.** *Let S be a symmetric matrix and let $\mathbf{u} = \begin{bmatrix} u_1 & \dots & u_N \end{bmatrix}$ be a vector of N independent samples from a random variable U which takes values -1 and 1 each with probability $1/2$. Then $\mathbb{E}[\mathbf{u}^T S\mathbf{u}] = \operatorname{tr}(S)$.*

   Matrix $\tilde{S}$ is symmetric, we can therefore exploit the theorem. Results are available in the literature on the asympotic properties of this estimator, which most importantly they do not depend on the size of the matrix. In practice, one makes a compromise between the speed of execution and the accuracy of the computation of the degrees of freedom. For our purposes, 60 realizations of the random vector $\mathbf{u}$ were sufficient (tested on the C-shape domain, the grid of λ was leading to larger variations).

   This method allows to reduce drastically the memory necessary to compute the trace of this matrix. Consider for example just the last matrix multiplication, $Q_i\Psi_i$. The result is a dense matrix because Q_i is dense and will have dimensions $n_i \times n_{\tau_i}$, which could be already a large number, if the mesh is fine on a manifold or a 3D domain. By using previous theorem instead, $\Psi_i\mathbf{u}$ has length n_i, which multiplied by Q_i is again a vector of length n_i.

4. Statistical inference

4.1. Properties of the estimators

4.1.1. Spatial field component

To study the statistical properties of the estimated spatial field that is obtained by solving the monolithic system 2.32, we write it in the following equivalent manner

$$\begin{cases} \tilde{\Psi}^T Q \tilde{\Psi} \hat{\mathbf{f}} - \lambda \tilde{R}_1^T \hat{\mathbf{g}} = \tilde{\Psi}^T Q \mathbf{z} \\ \hat{\mathbf{g}} = -\tilde{R}_0^{-1} \tilde{R}_1 \hat{\mathbf{f}} \end{cases} \tag{4.1}$$

By recalling the mixed-effect model in the implementative form 2.20, that is $\mathbf{z} = X\boldsymbol{\nu} + \mathbf{f}_N + \boldsymbol{\epsilon}$, and substituting the second equation inside the first we get

$$\tilde{\Psi}^T Q \tilde{\Psi} \hat{\mathbf{f}} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \hat{\mathbf{f}} = \tilde{\Psi}^T Q \left(X\boldsymbol{\nu} + \mathbf{f}_N + \boldsymbol{\epsilon} \right) \tag{4.2}$$

Recalling that Q projects on the orthogonal space of the columns of X, implying $QX = 0$,

$$\left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right) \hat{\mathbf{f}} = \tilde{\Psi}^T Q \mathbf{f}_N + \tilde{\Psi}^T Q \boldsymbol{\epsilon} \tag{4.3}$$

That can be rearranged as follows

$$\left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right) \left(\hat{\mathbf{f}} - \mathbf{f} \right) + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \mathbf{f} = \tilde{\Psi}^T Q \boldsymbol{\epsilon} \tag{4.4}$$

where we have defined $\mathbf{f}$ as the coefficients of the finite element expansion of the true vector field $f(\cdot)$. Deriving now the term $\hat{\mathbf{f}} - \mathbf{f}$

$$\hat{\mathbf{f}} - \mathbf{f} = -\left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right)^{-1} \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \mathbf{f} + \left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right)^{-1} \tilde{\Psi}^T Q \boldsymbol{\epsilon} \tag{4.5}$$

The first term above is not random, thus we can derive the following:

1. Thanks to the property that, for generic suitably dimensioned matrices and vectors, $\mathbb{E}[M\mathbf{x}] = M\mathbb{E}[\mathbf{x}]$, we get that $\mathbb{E}\left[\hat{\mathbf{f}} - \mathbf{f}\right] = -\left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right)^{-1} \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \mathbf{f}$;
2. Thanks to the property that again, for generic suitably dimensioned matrices and vectors, $Var[M\mathbf{x}] = MVar[\mathbf{x}]M^T$, we get this result for $Var\left[\hat{\mathbf{f}} - \mathbf{f}\right] = Var\left[\hat{\mathbf{f}}\right]$

$$Var\left[\hat{\mathbf{f}} - \mathbf{f}\right] = \sigma^2 \left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right)^{-1} \tilde{\Psi}^T Q \tilde{\Psi} \left(\tilde{\Psi}^T Q \tilde{\Psi} + \lambda \tilde{R}_1^T \tilde{R}_0^{-1} \tilde{R}_1 \right)^{-1} \tag{4.6}$$

where we have exploited symmetric matrices and the fact that $Q^2 = Q$.

4.1.2. Parametric component

For what concerns the estimation of the parameter $\boldsymbol{\nu}$, we recall that $\hat{\boldsymbol{\nu}}$ solves the normal equations

$$X^T X\hat{\boldsymbol{\nu}} = X^T\left(\boldsymbol{z} - \tilde{\Psi}\hat{\mathbf{f}}\right) \tag{4.7}$$

that with model 2.20 can be written as

$$X^T X\hat{\boldsymbol{\nu}} = X^T\left(X\boldsymbol{\nu} + \tilde{\Psi}\mathbf{f} + \boldsymbol{\epsilon} - \tilde{\Psi}\hat{\mathbf{f}}\right) \tag{4.8}$$

or

$$X^T X\left(\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}\right) = -X^T\tilde{\Psi}\left(\hat{\mathbf{f}} - \mathbf{f}\right) + X^T\boldsymbol{\epsilon} \tag{4.9}$$

From this we can derive:

$$\mathbb{E}\left[\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}\right] = \left(X^T X\right)^{-1} X^T\tilde{\Psi}\left(\tilde{\Psi}^T Q\tilde{\Psi} + \lambda\tilde{R}_1^T\tilde{R}_0^{-1}\tilde{R}_1\right)^{-1}\lambda\tilde{R}_1^T\tilde{R}_0^{-1}\tilde{R}_1\mathbf{f} \tag{4.10}$$

and the expression of $\mathbb{V}\mathrm{ar}\left[\hat{\boldsymbol{\nu}} - \boldsymbol{\nu}\right]$ that is not reported for brevity.

5. Simulation studies

5.1. Assumptions

The general setting described in section 2.7 is simplified in the rest of this work by assuming:

- same domain for all the patients, *i.e.* $\Omega_1 = \ldots = \Omega_m := \Omega$;
- same locations $\mathbf{p}_{ij}$ for every patient i, that is $\mathbf{p}_{ij} = \mathbf{p}_{kj} \quad \forall (i,k) \in \{1 \ldots m\}^2$. This also implies that n_i, number of observations for patient i, is the same for every i. In the following we will use $n = n_i$.
- same finite element basis used across all m domains Ω ($N_\tau = m \times n$).

These properties allow storing in memory some of the matrices described above just for one patient rather than for all patients. In the problems where assumptions above are valid but some data are missing, as it happens in concrete cases, some adjustments are necessary. The treatment is analogous to the one for the basic spatial regression case described in appendix A.2. In fact, in theory the mixed-effects model is not distinguishable from a model in which the random effects have been treated as different covariates for each unit.

To treat missing values the following adjustments are therefore sufficient:

- replace the missing values with 0;
- replace the covariates corresponding to the missing value with 0s;
- replace the rows of $\tilde{\Psi}$ matrix corresponding to the missing value with 0s.

We make a comparative study on the performance of the iterative method, with respect to solving the monolithic matrix. Several cases are considered in the following sections.

From an implementative point of view, I have extended the capabilities of the version of the library implemented by Kim, who in turn developed a branch of the `fdaPDE` package available on `CRAN`. Her branch is called `fdaPDE_mixed`, while my code is available on GitHub, see [Isc].

It is worth mentioning that when I began to work on this project, several new versions of `fdaPDE` had come out, so I took the opportunity to update the APIs that were provided by Kim and refactor her code in such a way that it will be easear to maintain a *mixed-effects* branch in the future.

5.2. Simulations

5.2.1. Bidimensional domain

The first case that we study is the same treated in [Kim20]. The domain considered is a C-shape, with a surface test function obtained from Wood[WBH08], originally introduced by Ramsay [Ram02]. The C-shaped function features the property that the inner part of the domain is close in terms of distance, yet the evaluations are in contrast between the upper and lower sides.

In her work Kim showed that the SRPDE approach leaded to results similar to the one provided by a generalized additive model, and performed better in terms of mean root square error concerning the spatial component of the model.

In the simulation, we generate randomly 50 scenarios for the mixed-effects model with 3 statistical units each. Errors ϵ are simulated as $\mathcal{N}(0, 0.05 * k)$ where k is the range (difference between maximum and minimum) of the spatial field over the observed locations. For each statistical unit we observe 100 fixed locations, more or less uniformly on the C-shaped domain.

The true parameter values are set to $\beta_1 = 3$, $\beta_2 = 0.5$, $b_{11} = -5$, $b_{21} = 0$, and $b_{31} = 5$. Covariates are generated as $sin(2\pi x)cos(2\pi y)$ and $N(0, 4)$, respectively. Both covariates are the same across all statistical units.

The model estimates coefficients given a fixed grid of λ values, and the λ value with the minimum generalized cross-validation (GCV) is chosen as the best λ from the grid. The logic for choosing the appropiate values of the grid is the following: if the same λ is observed most of the times on different simulations, it means that the grid is not fine enough near the optimal λ. Moreover the grid of λ should be big enough so that the boundary of the grid is not chosen. Cleary, on the other hand, if the grid is too fine, the method would be very slow. In practice the grid is chosen through trial and error.

Tolerance for the iterative method was chosen to be 10^{-6}, the number of random vectors to estimate the trace of the smoothing matrix.

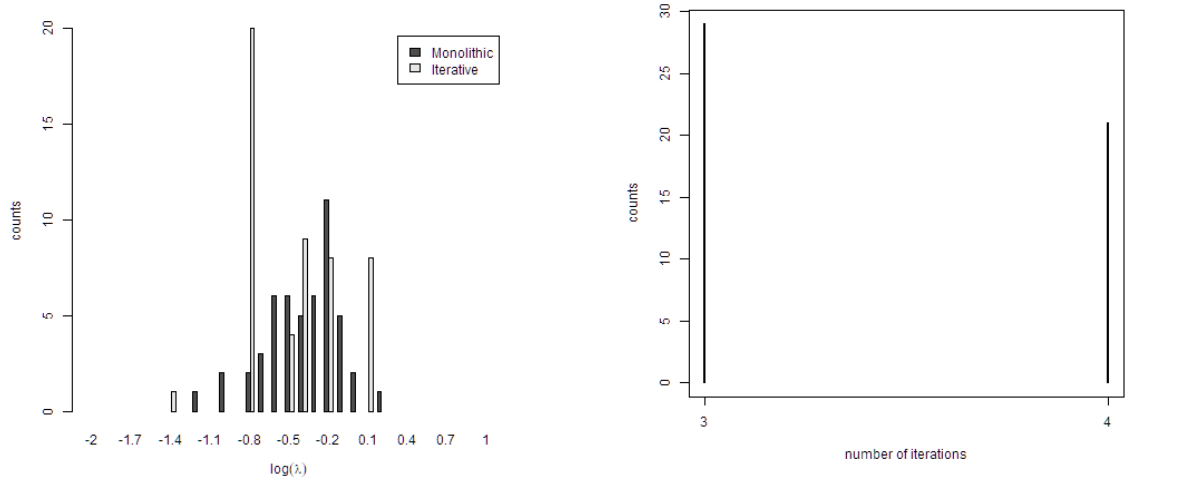
Solutions for a fixed λ are here always the same, if not for the tolerance. In figure 5.1a we show the comparison of the values of λ selected, where the degrees of freedom have been estimated on the full smoothing matrix for the monolithic case and on the diagonal approximation for the iterative method.

Figure 5.1b shows how many iterations were needed for convergence on the grid of λ. In all cases it converged very rapidly, in less than 4 iterations.

In figure 5.2a we show the boxplot for the estimated β_1, β_2, with no significant difference between the two solvers. Same goes for figure 5.2b for what concerns the values of b_i.

Finally, figure 5.3 shows that there is no degradation of performance on the global error estimated for the model.

We tried a bit varying the number of units and the number of locations and no problems have been yet observed. However, it is important to note that the simulation study was conducted under specific assumptions and conditions, and the performance of the method may vary in different settings.



(a) *Frequency of selected λ over 50 simulations.*

(b) *Iterations to reach convergence across different selected λs.*

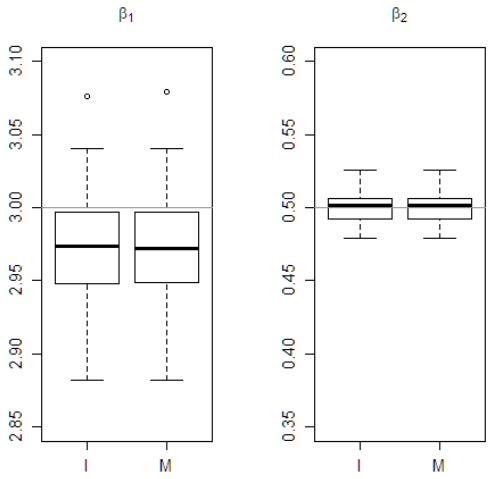
Figure 5.1.: *Simulation on C-shape domain. Both figures refer to the optimal λ on a grid according to a generalized cross-validation criteria.*

5.2.2. Manifold domain

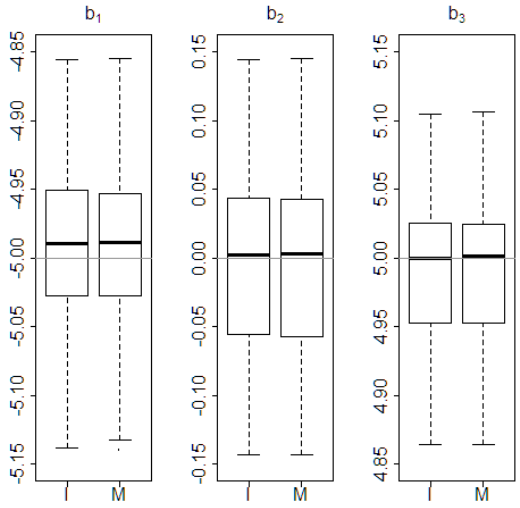
Simulations have been carried out on the surface of a sphere, see the mesh in 5.4. Again, the iterative method converged to the desired solution in less than 6 iterations.

5.2.3. NAs handling

The ability of the model to deal with missing values has been tested on the C-shape domain. Convergence was within the 6 iterations number again. See 5.5a 5.5b 5.6 for the results.



(a) *Boxplots of fixed effects parameters.* β_1 *is related to the covariate considered also as a random effect for each unit (which is also space dependant), while* β_2 *corresponds to the covariate considered for its contribution as a fixed effect.*



(b) *Boxplots of random effects parameters, for every statistical unit.*

Figure 5.2.: *Estimated parametric coefficients for the mixed-effects on the C-shape domain.* I *label is relative to the iterative method while* M *is relative to the monolithic solver. The gray horizontal line represents the true vale for the parameter. No meaningful differences are observed between the two solvers.*

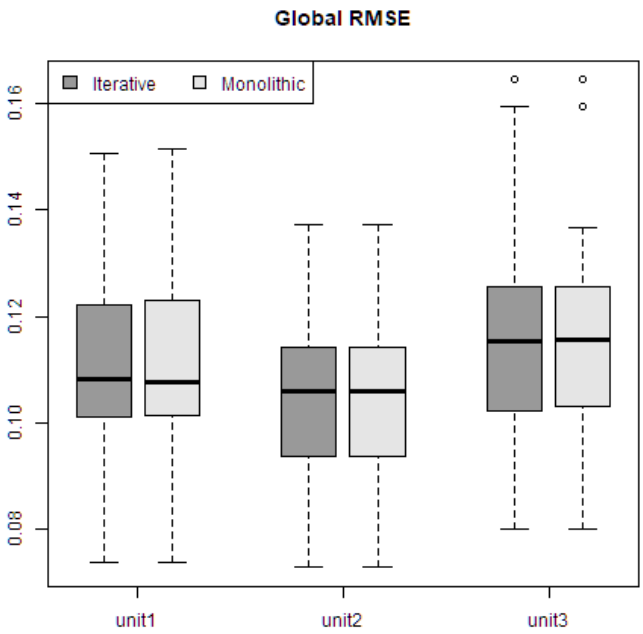


Figure 5.3.: *Root mean square error for the estimated model.*

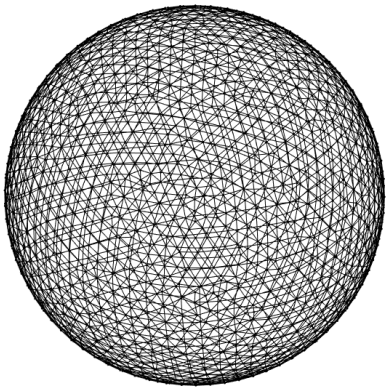
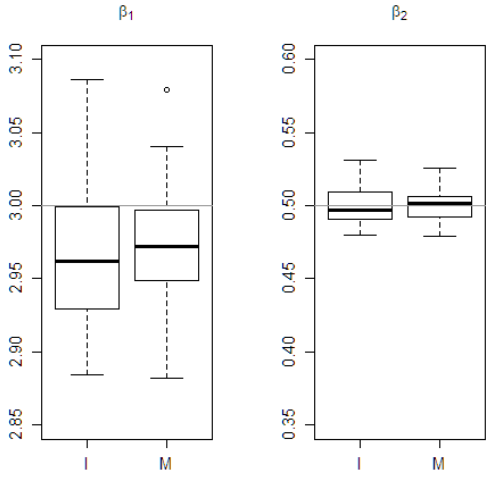
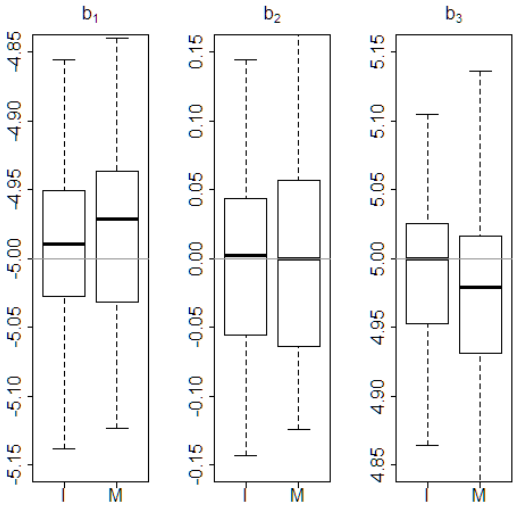


Figure 5.4.: *Mesh for the example of SR-PDE model on a manifold.*



(a) *Boxplots of fixed effects parameters.* β_1 *is related to the covariate considered also as a random effect for each unit, while* β_2 *corresponds to the covariate considered for its contribution as a fixed effect.*



(b) *Boxplots of random effects parameters, for every statistical unit.*

Figure 5.5.: *Estimated parametric coefficients for the mixed-effects on the C-shape domain in case of missing values.*

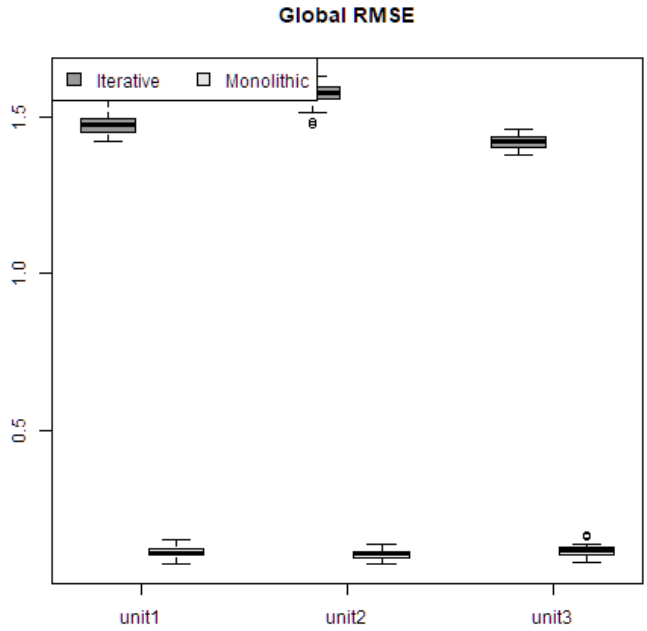


Figure 5.6.: *Root mean square error for the estimated model in case of missing values.*

6. Applications

Functional Magnetic Resonance Imaging (fMRI) is an imaging technique developed to detect regional, time-varying changes in brain metabolism, which can arise from task-induced cognitive state changes or unregulated processes in the resting brain. Since the 90s, fMRI has become widely used in the context of cognitive neurosciences, psychiatry and psychology and a large number of studies are available on it. The popularity of fMRI can be attributed to its widespread availability, non-invasive nature, relatively low cost, and good spatial resolution.

In this study, we want to prosecute the work started in [Kim20]. In the appendix of her thesis the interested reader can find the description on how to access all the databases we used, and also the patients that she utilized in her study. For the sake of the mixed-effects model, the statistical units are several patients on which fMRI observations were taken as a time-series.

As described by Kim, we first process the large amount of time-series data (1200 rows per brain location), and we move to analyze the so-called Functional Connectivity (FC) maps. FC maps are constructed by computing pairwise correlations between the fMRI time-series of all brain locations and the mean time-series of a Region of Interest (ROI) on the cortical surface. Further, a Fisher transformation is applied on the obtained variable. In this way, we can shed some light in identifyng the areas of the brain that are functionally connected to the ROI.

As done before in similar studies, we consider precuneus as the Region Of Interest. We use the available data of the cortical thickness of the cerebral cortex as a covariate for our mixed-effects model. The cortex is in fact known to stimulate neural activity.

For this type of application it is possible to consider the thin cerebral cortex as a 2D surface embedded in a 3D space. Despite the unique geometrical properties of each individual brain, we use a common standard surface provided by the Conte69 brain atlas, as well as a common mesh. Moreover, as it was done in [Kim20] we limit ourselves to the left hemisphere. The Conte69 brain atlas contains a large number of locations (32492) in the brain. This locations correspond with the locations on which we observe the fMRI data.

Kim, not having at her disposal a version of the library that exploits the convenience of the use of sparse matrices, as well as the iterative method that greatly reduces the amount of RAM needed, had to perform some simplifications. She used two different meshes, the original one with 32492 nodes and a simplified one with 10000 nodes. The latter allowed her to study at same time more patients while keeping the system manageable. She considered 12 patients for the full mesh study and 30 for the simplified mesh study, but several hundreds are available. In my case, I moved to 60 patients, as my small hard disk was not allowing more.

The mesh strategy that she adopted was well studied and took into account various criteria, such as preserving the topology and shape of the original mesh, but the results were not validated.

The iterative method successfully run on a grid of 10 λs, using again 60 as the number of stochastic realizations to estimate the degrees of freedom. Again the method converged in less than 6 iterations, taking less than two hours to run, and the parameters estimated are in line with those of [Kim20].

7. Conclusions and future developments

In previous sections we have shown that the iterative method is a valid alternative to the more direct computation for the solution. It has been implemented in the `fdaPDE` library for the bidimensional case and the manifold case. Some further extension and generalization has to be carried out, as well as some optimization in the regarding the layout of the matrices and their multiplication.

While the focus has been on models sharing the same domain and location points, future work should include multiple domains with different location points. The 3D case has still to be implemented, and the only strategy used here for estimating the value of `lambda` was computing the model on a grid of values. The methodology could also be possibly extended to spatio-temporal data.

Appendices

A. Formulae

A.1. Woodbury decomposition

The following matrix identity holds.

Proposition A.1 (Woodbury matrix identity)**.** *Let* M *be a square* $m \times m$ *matrix which can be written as the sum* $E + UCV$*, with* E *being* $m \times m$*,* U *being* $m \times n$*,* C *being a square* $n \times n$ *matrix, and* V $n \times m$*. Then*

$$M^{-1} = (E + UCV)^{-1} = E^{-1} - E^{-1}U\left(C^{-1} + VE^{-1}U\right)^{-1} VE^{-1}. \tag{A.1}$$

Woodbury decomposition is useful in the case where computing a factorization of matrix E can be considered cheap or advantageous, for example when solving a system involving this matrix is necessary in many places. In fact, looking at the right hand side of previous equation, solving a system involving matrix M can be done by solving:

- several $m \times m$ systems involving matrix E;
- an $n \times n$ system, where n is supposed to be smaller than m, characterized by matrix $\left(C^{-1} + VE^{-1}U\right)$, as well as several $n \times n$ systems to compute the inverse of C, but we will see it is not necessary in our case.

This equation is exploited for faster system solving in the `fdaPDE` library.
As an example, consider the space-only problem, described *e.g.* in [San21]: the presence of covariates leads to a linear system involving the following matrix M:

$$M = \begin{bmatrix} \Psi^T Q \Psi & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix}.$$

Remembering that the projection matrix Q is defined as $I - H$, M can be split into the following two components, one independent from λ:

$$M = \begin{bmatrix} \Psi^T \Psi & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix} + \begin{bmatrix} -\Psi^T H \Psi & 0 \\ 0 & 0 \end{bmatrix}.$$

By defining E the left matrix of the two above (which is also the matrix corresponding to the problem without covariates), and remembering that $H = W\left(W^T W\right)^{-1} W^T$, Woodbury decomposition can be exploited, by defining the matrices U, C, V in the following way:

$$UCV = \begin{bmatrix} -\Psi^T W \left(W^T W\right)^{-1} W^T \Psi & 0 \\ 0 & 0 \end{bmatrix},$$

$$U = \begin{bmatrix} \Psi^T W \\ 0 \end{bmatrix},$$
$$C = -\left[\left(W^T W \right)^{-1} \right],$$
$$V = U^T = \begin{bmatrix} W^T \Psi & 0 \end{bmatrix},$$

where the 0s indicate matrices of zeros of suitable dimensions.

In this case solving a system involving matrix C is simply a multiplication by $W^T W$.

Moreover, matrix

$$E = \begin{bmatrix} \Psi^T \Psi & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix} \tag{A.2}$$

is fully sparse and this is very convenient when the size is large.

A.2. Handling missing values

Consider, as in appendix A.1, the basic space-only problem, characterized by equations

$$\begin{bmatrix} \Psi^T Q \Psi & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} \Psi^T Q z \\ 0 \end{bmatrix}. \tag{A.3}$$

We could use the notation Ψ_n, Q_n, z_n since these are the quantities that depend on the n observed values.

Now suppose that the m-th observation z_m is not available. If we were unaware of this information, we would build the system with quantities of index $n-1$:

$$\begin{bmatrix} \Psi_{n-1}^T Q_{n-1} \Psi_{n-1} & -\lambda R_1^T \\ -\lambda R_1 & -\lambda R_0 \end{bmatrix} \begin{bmatrix} f \\ g \end{bmatrix} = \begin{bmatrix} \Psi_{n-1}^T Q_{n-1} z_{n-1} \\ 0 \end{bmatrix} \tag{A.4}$$

From a computational point of view, when implementing the mixed-effects model with multiple systems to solve analogous to the previous one, or simililarly in spatio-temporal regression, it is certainly preferable to keep the matrices with the original size rather than building matrices of different sizes. This fact justifies the need to write a system equivalent to A.4 as a system corresponding to an n observations case.
We impose equality of correspondant terms.
By definition of $Q = I - W\left(W^T W\right) W^T$, it is logical to set to 0 the covariates corresponding to the m-th unused observation: in such a way $W_n^T W_n = W_{n-1}^T W_{n-1}$. This operation leads to Q_n being the matrix obtained from Q_{n-1} by adding the m-th row and the m-th column of zeros except for the unitary diagonal element.
For the left-hand side,

$$\left(\Psi_n^T Q_n \Psi_n \right)_{ij} = \left(\Psi_{n-1}^T Q_{n-1} \Psi_{n-1} \right)_{ij} \tag{A.5}$$

leads to

$$\sum_{k,l} (\Psi_n)_{ki} (Q_n)_{kl} (\Psi_n)_{lj} = \sum_{k,l} (\Psi_{n-1})_{ki} (Q_{n-1})_{kl} (\Psi_{n-1})_{lj}. \tag{A.6}$$

Removing the identical terms, we are left with:

$$\sum_{k=m,l} (\Psi_n)_{ki} (Q_n)_{kl} (\Psi_n)_{lj} + \sum_{k\neq m,l=m} (\Psi_n)_{ki} (Q_n)_{kl} (\Psi_n)_{lj} = 0 \tag{A.7}$$

Removing the zero terms, we obtain

$$(\Psi_n)_{mi} (\Psi_n)_{mj} = 0 \tag{A.8}$$

We obtain then that Ψ_n is the same as Ψ_{n-1} except for the m-th row, where its original value is replaced by zeros. Recalling the meaning of Ψ matrix, this means that any function evaluates to 0 in the position corresponding to the missing value.

Comparing the right-hand side,

$$\left(\Psi_{n-1}^T Q_{n-1} z_{n-1}\right)_i = \left(\Psi_n^T Q_n z_n\right)_i \tag{A.9}$$

we get

$$\sum_{k,l} (\Psi_n)_{ki} (Q_n)_{kl} (z_n)_l = \sum_{k,l} (\Psi_{n-1})_{ki} (Q_{n-1})_{kl} (z_{n-1})_l \,. \tag{A.10}$$

Again, assuming we set $z_m = 0$, the differences come on lhs when $k = m$ or $l = m$. But the case $l = m$ is 0 because z_m is 0. The case $k = m$ is 0 because $(Q_n)_{ml}$ is 0.

In conclusion, for writing an equivalent system it is sufficient, like one could naturally guess, to:

- replace the missing value with 0;
- replace the covariates corresponding to the missing value with 0s;
- replace the row of Ψ matrix corresponding to the missing value with 0s.

Notice that it is straightforward to generalize this result in the case where the missing values are more than one.

Bibliography

[Azz+14] L. Azzimonti, F. Nobile, L. M. Sangalli, and P. Secchi. "Mixed Finite Elements for Spatial Regression with PDE Penalization". In: *SIAM/ASA Journal on Uncertainty Quantification* 2.1 (2014), pp. 305–335. DOI: 10.1137/130925426. URL: https://doi.org/10.1137/130925426 (cit. on p. 21).

[BHT89] A. Buja, T. Hastie, and R. Tibshirani. "Linear Smoothers and Additive Models". In: *The Annals of Statistics* 17 (June 1989). DOI: 10.1214/aos/1176347115.

[Ger+14] P. Gervasio, A. Quarteroni, R. Sacco, and F. Saleri. *Matematica Numerica*. Feb. 2014. ISBN: 978-88-470-5644-2 (cit. on p. 27).

[Isc] M. Ischia. URL: https://github.com/martinoischia/fdaPDE_martino. (accessed: 26.06.2023) (cit. on p. 33).

[Kim20] J. Kim. "Mixed-Effect Models in Spatial Regression with Partial Differential Equation Regularization". Politecnico di Milano, 2020 (cit. on pp. 16, 17, 23, 33, 34, 39, 40).

[Lil+20] E. Lila, L. Sangalli, E. Arnone, J. Ramsay, and L. Formaggia. "fdapde: statistical analysis of functional and spatial data, based on regression with PDE regularization". In: *URL: https://CRAN. R-project. org/package= fdaPDE* (2020).

[MS21] M. Massardi and S. Spaziani. *fdaPDE: an efficient iterative method for spatio-temporal regression with PDE regularization*. Politecnico di Milano, APSC course, 2021 (cit. on p. 25).

[PP13] M. Pollini and L. Ponti. *Regressione spazio-temporale su domini bidimensionali non planari*. Politecnico di Milano, NAPDE course, 2013 (cit. on p. 25).

[Qua23] A. Quarteroni. "Modellistica Numerica per Problemi Differenziali (Third. Ed.)" In: (Apr. 2023) (cit. on pp. 21, 22).

[Ram02] T. Ramsay. "Spline smoothing over difficult regions". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 64.2 (2002), pp. 307–319 (cit. on p. 34).

[San21] L. Sangalli. "Spatial Regression With Partial Differential Equation Regularisation". In: *International Statistical Review* (Mar. 2021). DOI: 10.1111/insr.12444 (cit. on pp. 15, 28, 45).

[SFF21] L. Sangalli, F. Ferraccioli, and L. Finos. "Some first inferential tools for spatial regression with differential regularization". In: *Journal of Multivariate Analysis* (2021).

[SRR13] L. Sangalli, J. Ramsay, and T. Ramsay. "Spatial spline regression models". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 75 (Sept. 2013). DOI: 10.1111/rssb.12009 (cit. on p. 15).

[WBH08] S. N. Wood, M. V. Bravington, and S. L. Hedley. "Soap film smoothing". In: *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* 70.5 (2008), pp. 931–955 (cit. on p. 34).

[Woo17] S. N. Wood. *Generalized additive models: an introduction with R*. CRC press, 2017.

[Woo19] S. Wood. *mgcv: mixed GAM computation vehicle with automatic smoothness estimation. R-package version 1.8–31*. 2019.