Generalized Spatial Regression with Differential Penalization

Matthieu Wilhelm

Director: Dr. Laura Maria Sangalli

August 13, 2013

Master Project

Politenico di Milano

MOX, Laboratorio di Modellistica e di Calcolo Scientifico

Ecole polytechnique fédérale de Lausanne

Chaire de statistique mathématique, Prof. Victor M. Panaretos

ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Avant-propos

Ce travail a été mené au cours d'un séjour au Politecnico di Milano, Italie, entre les mois de février et de juin 2013. J'ai été supervisé par le Dr. Laura M. Sangalli, chercheuse associée au laboratoire de modélisation et de calcul scientifique (MOX) du département de mathématique Francesco Brioschi du Politecnico di Milano. Ce projet s'inscrit dans le cadre de mon master en ingénierie mathématique entammé à l'Ecole polytechnique fédérale de Lausanne en septembre 2011.

Foreword

This work has been done during a stay at the Politecnico di Milano, Italy, between February and June 2013. I have been supervised by the Dr. Laura Sangalli, associate professor at the modelling and scientific calculus laboratory (MOX) of the mathematical department "Francesco Brioschi" of the Politecnico di Milano. This thesis is a part of my master initiated in September 2011 at the Ecole polytechnique fédérale de Lausanne.

Résumé

Nous proposons une méthode noavatrice pour l'analyse de données spatialement distribuées sur des surfaces dont les bords sont irréguliers et qui sont la réalisation d'une variable exponentielle. On considère le contexte des modèles additifs généralisés et l'on étend le travail de Sangalli et al. (2013) à des distributions autres que gaussiennes. En particulier, on peut alors considérer les distributions de famille exponentielle (par exemple des données binomiales, de Poisson ou de distribution gamma), et ainsi disposer d'un modèle avec un fort potentiel d'application. Pour l'ajustement du modèle, on maximise une fonction de log-vraisemblance pénalisée. Le terme de pénalisation tient compte d'un opérateur différentiel appliqué à la fonction que l'on cherche à estimer. Le modèle permet aussi d'inclure des informations auxiliaires telles que des variables explicatives. Le modèle proposé utilise des méthodes issues du domaine du calcul scientifique et fait en particulier usage de la méthode des éléments finis, qui permet d'imposer des conditions au bord à la fonction recherchée. Finalement, on étend de manière théorique le modèle développé pour des données distribuées sur une surface non planaire.

Abstract

We propose a novel method for the analysis of spatially distributed data from an exponential family distribution, able to efficiently treat data occurring over irregularly shaped domains. We consider a generalized linear framework and extend the work of Sangalli et al. (2013) to distributions other than the Gaussian. In particular, we can handle all distributions within the exponential family, including binomial, Poisson and Gamma outcomes, hence leading to a very broad applicability of the proposed model. We maximize a penalized log-likelihood function. The roughness penalty term involves a suitable differential operator of the spatial field over the domain of interest. This maximization is done via a penalized iterative least square approach (see Wood (2006)). Covariate information can also be included in the model in a semi-parametric setting. The proposed models exploit advanced scientific computing techniques and specifically make use of the Finite Element Method, that provides a basis for piecewise polynomial surfaces and allows to impose boundary conditions on the space distribution of the probability. Finally, we extend theoretically the model to deal with data occurring on a two dimensional manifold.

Key words

Generalized additive models, spatial spline regression, finite element method, penalized regression

Thanks... I'm very grateful to Professor Victor Panaretos who gave me the chance to make a very interesting master thesis. I also would thank Dr. Kjell Konis who made me like computational statistics being my mentor with **R**.

Ringraziamenti... Vorrei ringraziare Laura M. Sangalli per avermi proposto un soggetto stimolante che ha rapresentato una degna fine del mio percorso di studi. Mi ha seguito con cura e attenzione durante questi mesi. Oltre all'aspetto scientifico, mi ha coinvolto nel gruppo di statistica del MOX e per questo gliene sono molto grato. Vorrei ringraziare Bree, per il suo aiuto nel risolvere certi problemi. La ringrazio soprattutto di avere condiviso con me tutte le pause gelato, momenti simpatici e allegri, sempre piacevoli a fine giornata. Vorrei ringraziare i miei colleghi del gruppo di statistica del MOX: la loro accoglienza e i pranzi del venerdì rimarranno dei bei ricordi.
In ultimo, vorrei ringraziare la comunità del Tender. Non ha prezzo arrivare la mattina in ufficio e sentire un atmosfera così cordiale e simpatica. In particolare, vorrei ringraziare Anna, Marianna, Ilaria, Franco, Mirko, Nur, Domenico, Anwar, Guido Abramo e Davide. Senza dubbio, è il migliore ufficio in cui sarò mai stato!

Remerciements... La fin des études est sans aucun doute, pour ceux qui ont la chance d'en faire, un rite de passage important. Cela marque le début de l'indépendance matérielle et donc la fin d'une époque, celle de l'éducation, de l'élévation (quand bien même ce terme demeure peu usité en ce sens). Je tiens donc à remercier mes parents, Luc et Luisa de m'avoir élevé, jusqu'à l'indépendance à laquelle je peux prétendre aujourd'hui. Ils m'ont donné l'amour et la sécurité qui m'ont permis d'être ce que je suis aujourd'hui. Ils m'ont transmis les valeurs qui me permettent d'aborder avec confiance la vie qui s'offre à moi. Plus qu'un remerciement, c'est un hommage que je souhaite leur rendre en cette occasion.
Je ne serais pas non plus ce que je suis sans mes deux frères, Pierre et Lionel. L'âge adulte nous aura permis de fonder notre unité. Je ne serai jamais seul, et je leur dois cette rassurante et intime conviction. Nous sommes le meilleur exemple que l'ensemble peut-être plus grand que la somme des parties.
Je ne peux pas évoquer ma famille proche sans évoquer Sophie, ma très chère belle-soeur. Si notre famille est si harmonieuse, c'est en partie grâce à elle.
Je souhaite rendre hommage à mes grands-parents et à mes nonni, François et Françoise Wilhelm, Angelo e Elisa Mozzi-Strick. Outre l'affection qu'ils nous ont toujours généreusement témoignée, par leur vie et leur histoire, ils ont été et sont toujours pour nous de véritables exemples. Cette dernière année, passée entre août et décembre à Neuchâtel et à Milan entre février et juin m'a permis de partager de beaux moments avec eux. Je suis conscient de la rare chance de pouvoir encore jouir de leur présence et je suis particulièrement heureux de pouvoir rentrer dans la vie active en les ayant encore à mes côtés. Mes grands-parents et mes nonni représentent une génération qui a fait des sacrifices conséquents pour que les générations suivantes, c'est-à-dire nous, puissent vivre mieux. C'est l'occasion pour moi de leur témoigner ma gratitude car ces peines et ces sacrifices n'ont pas été vains.
Je souhaite remercier mon parrain et ma marraine, Guy et Christiane Rambaud. Tout au long de ma vie, ils ont assumé avec conviction la charge qu'ils avaient accepté le jour de mon baptême. Ils sont et demeureront pour moi des références.
Je n'ai pas passé ces années à l'EPFL totalement seul. Mes compagnons d'études, mes compagnons de galère, m'ont été d'un grand soutient. Je veux les remercier pour tous les moments et toutes les discussions passées ensemble, pour tous les services qu'ils m'ont rendu et pour avoir rendu ces années agréable. Je veux en particulier remercier Reda Boumasmoud, Denis Devaud, Adrien Hitz et Thomas Lugrin. Je garderai un merveilleux souvenir de ma

volée, dans laquelle j'inclus abusivement Cyril, avec de beaux moments partagés, en particulier à Istanbul lors de notre voyage d'étude, lors de la préparation de celui-ci et lors des différents séjours à Zinal.
Pour leur longue et indéfectible amitié, je tiens à remercier Matthieu, Raphaëlle, Marina. Ils sont pour moi l'incarnation de ce que devrait être l'amitié: de la loyauté, de la chaleur, du respect et des bons moments à revendre!
Finalement, ma vie ne serait pas la même sans Marie à mes côtés. Elle sait combien cela me rend heureux.

Fais confiance au Seigneur, agis bien, habite la terre et reste fidèle, Psaumes (36.3)[1]

[1] *in* La Bible liturgique, Association episcopale liturgique des pays francophones (AELF), 1980

Contents

1 Introduction **11**

2 Model and Motivation **13**

2.1 Data and model 13

2.2 The framework: the exponential family 14

2.3 Penalized iterative reweighted least squares algorithm 15

2.4 Weighted penalized least square minimization problem 16

3 Finite Elements Formulation **19**

3.1 A very short introduction 19

3.2 Reformulation of the problem 19

3.3 From weak formulation to linear systems 20

3.4 Explicit form of the estimates 22

3.5 Generalization to a mesh independent of the data points 22

4 Justification of the PIRLS algorithm and penalized log-likelihood **25**

4.1 Functional derivation of the PIRLS algorithm 25

4.2 Derivation of the penality matrix 28

4.3 Finite dimensional derivation of the PIRLS algorithm 29

5 Smoothing parameter choice and variance estimation **31**

5.1 Choosing the smoothing parameter: a bias/variance trade-off 31

5.2 Choosing the smooth parameter: the generalized cross validation 32

5.3 Variance and scale parameter estimation 33

6 Simulations **35**

6.1 Meshes and data 36

6.2 Simulations without covariates 38

6.2.1 Binary responses 38

6.2.2 Gamma responses 44

6.3 Simulations with covariates 46

6.3.1 Binary responses with covariates 46

6.3.2 Gamma responses with covariates 46

6.4 Confidence intervals and distributional results 53

6.5 Comparison with other methods 55

7 Application: Pennsylvania lung cancer **57**

8 A first approach to non planar domains 61
8.1 A geometric framework 62
8.2 Finite element solution in the case of surface embedded in $\mathbb{R}^3$ 63

9 Conclusion and perspective 65

Bibliography 68

Notations

For a vector, bold lower case is used, for example $\boldsymbol{\mu}$. When a similar symbol in normal font and with an index i is used, e.g μ_i, it denotes the ith component of the vector $\boldsymbol{\mu}$. Bold upper case is used to denote a matrix. For example $\mathbf{X}$ denotes the design matrix. Normal lower case usually denotes functions. Maximum likelihood, or maximum penalized likelihood estimates are denoted with a hat, e.g., $\hat{\boldsymbol{\beta}}$. When a single variable function is applied to vector, it results in vector whose components are the evaluation of the function at the components. For example, $g(\boldsymbol{\mu}) = (g(\mu_1), \ldots, g(\mu_n))^t$.

Chapter 1

Introduction

Many datasets are considered as realizations of functions disturbed by random noise. These data, called functional data, suggest that we seek to estimate the underlying function that generated such a process. This type of analysis is relatively recent and is a very active field of research in statistics (Ramsay and Silverman, 2005).

In this context, we aim to estimate a function. This unusual since we generally aims to estimate parameters lying in a finite dimensional space. In the case of an estimation of a function, the estimation process is generally in two stages. First, a subspace of finite dimension of the functional space in which the function f (or is assumed to lie) the true function is chosen. Then, the problem of estimating becomes a parametric estimation. The choice of this finite dimensional subspace is then crucial and is usually subjective. As examples of finite dimensional spaces, we can mention the subspace spanned by the Fourier basis, usually used for periodic functions, or the subspace spanned by polynomials of a finite degree.

This project aims to extend the methodology developed in Sangalli et al. (2013). The finite dimensional space used is generated by the finite element method. The finite element method is widely used in the field of numerical approximation of partial differential equations (PDE's). It allows to make approximations which are piecewise polynomial surfaces. In addition, they are suitable for estimates over complex domains, which is common in the field of PDE's, and allow to take into account boundary conditions. These characteristics make them particularly suited to the context of functional data analysis (for an introduction to finite elements, see Quarteroni (2009)).

The model developed in Sangalli et al. (2013) uses a least square error formulation which can be also viewed as the log-likelihood of normally distributed data to approximate spatial functions. This is not well suited for spatial distributed binary data, for example. The purpose of this project is to extend this model to other distributions such as Bernoulli, Poisson or Gamma. We use a methodology inspired from the one used to extend linear models to generalized linear models. In addition, as it is commonly done in the area of functional data analysis, a penalty ensures that the estimated function is smooth enough.

Chapter 2

Model and Motivation

In this project, our aim is to extend the model developed in Sangalli et al. (2013) to other distributions than normal distribution. We propose a generalized additive model (GAM), also called as generalized semiparametric regression model, (see Hastie and Tibshirani (1986, 1990); Wood (2006); Bickel et al. (1993)).

Let $\{\mathbf{p}_i = (p_{1i}, p_{2i}); i = 1, \dots, n\}$ be a set of n points on a bounded regular domain $\Omega \subset \mathbb{R}^2$, z_i the corresponding value of the observed variable at the point $\mathbf{p}_i$ and let $\mathbf{x}_i = (x_{1i}, \dots, x_{qi})^t$ be a q-vector of covariates associated to the observation y_i. The goal is to deal with models of the type:

$$g\left(\mathbb{E}\left[Y_i \mid \mathbf{x}_i, \mathbf{p}_i\right]\right) = \mathbf{x}_i^t \boldsymbol{\beta} + f(\mathbf{p}_i), \qquad Y_i \mid \mathbf{x}_i, \mathbf{p}_i \sim \text{exponential family distribution} \tag{2.1}$$

where $g(.)$ is a link function (see section 2.2 for details about exponential family distribution). We aim to estimate both $\boldsymbol{\beta} \in \mathbb{R}^q$ and $f \in \mathcal{F}$, a function which belongs to a functional space $\mathcal{F}$ to be defined. Many different methods have been developed to deal with this kind of problems. Since the function f belongs to a functional space, it is usual to estimate the function f using $\mathcal{F}_K$, a finite dimensional approximation of the (hilbertian) functional space $\mathcal{F}$ of dimension $K \in \mathbb{N}$. Using a basis $\{\psi_1, \dots, \psi_K\}$ of $\mathcal{F}_K$, the problem (2.1) simplifies to a traditional generalized linear model (GLM).

The goal of this project is to find an efficient way to compute estimates of the vector $\boldsymbol{\beta}$ and of the function f. As for a traditional GLM, we will use an iterative algorithm to approximate the solution of the maximum likelihood, called Iterative Reweighted Least Squares (IRLS, see McCullagh and Nelder (1989, p. 40-43)). In this case, we will use particular case of this algorithm to take into account the penalization of the functions that must ensure a sufficient smoothness of the approximation of the function f.

2.1 Data and model

We wish to estimate the parameter $\boldsymbol{\beta}$ and the function f, according to model (2.1).

A roughness penalization of the function f is needed in order to avoid overfitting of the function. The iterative procedure aims to compute, for a given value of λ, the maximum likelihood estimates $\hat{\boldsymbol{\beta}} \in \mathbb{R}^q$ and $\hat{f} \in \mathcal{F}_K$ of the model, corresponding to the maximum of the penalized log-likelihood function:

$$\mathcal{L}_p(\boldsymbol{\beta}, f) = \sum_{i=1}^{n} l(y_i; \eta_i(\boldsymbol{\beta}, f)) - \lambda \int_\Omega (\Delta f)^2, \tag{2.2}$$

where, $l(\cdot)$ is the log-likelihood, $\eta_i = \mathbf{x}_i^t \boldsymbol{\beta} + f(\mathbf{p}_i)$ and λ is a positive constant. In the case where the distribution of Y_i is normal, the log-likelihood function is easy to optimize because of the quadratic form of the penalized log-likelihood function. Actually, this functional can be written as:

$$\mathcal{L}_{\text{Gaussian}}(\boldsymbol{\beta}, f) = \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}_n\|^2 - \lambda \int_{\Omega} (\Delta f)^2,$$

where $\mathbf{f}_n = (f(\mathbf{p}_1), \ldots, f(\mathbf{p}_n))^t$. In the Gaussian case, the optimum is simply the solution of a linear system which is obtained after discretization of the functional space $\mathcal{F}$ (Sangalli et al., 2013). In the more general case of a distribution belonging to the exponential family, the penalized log-likelihood function may not be a quadratic form. We use an iterative algorithm to find this optimum. One of the well-known methods used in the context of GLM is the Iterative Reweighted Least Square (IRLS, (McCullagh and Nelder, 1989)). The idea is to compute at each step a weighted least square estimation. In our case, due to the penalization term, we will use a slightly different form of this algorithm, called Penalized Iterative Reweighted Least Squares (PIRLS, see Wood (2006) or Wood (2011)). This allows us to use the procedure developed in Sangalli et al. (2013) at each step of the algorithm.

2.2 The framework: the exponential family

A random variable Y is said to belong to the exponential family if its distribution can be written as:

$$f_Y(y) = \exp\left\{(y\theta - b(\theta))/a(\phi) + c(\phi, y)\right\},$$

where $a(\cdot)$, $b(\cdot)$ and $c(\cdot)$ are arbitrary functions subject to some regularity constraints (McCullagh and Nelder, 1989). The parameter θ is called *canonical parameter* and ϕ is called *scale parameter*. In the context of GLM, we have $g(\mu) = \eta$, where μ is the mean, $g(\cdot)$ a link function and $\eta = \mathbf{x}^t\boldsymbol{\beta}$ is called the *natural parameter*. So η is modelled as a linear combination of the observation $\mathbf{x}$ and thus, influence the modelling of the distribution. Link functions are continuous, strictly monotonic functions and are therefore invertible. On one hand, we have the following properties, for general maximum likelihood estimates:

$$\begin{aligned}
\mathbb{E}_0\left[\left.\frac{\partial l}{\partial \theta}\right|_{\theta=\theta_0}\right] &= 0, \\
\operatorname{var}\left(\left.\frac{\partial l}{\partial \theta}\right|_{\theta=\theta_0}\right) &= \mathbb{E}_0\left[\left.\frac{\partial l}{\partial \theta}\right|^2_{\theta=\theta_0}\right], \\
\mathbb{E}\left[\left.\frac{\partial l}{\partial \theta}\right|^2_{\theta=\theta_0}\right] &= \mathbb{E}_0\left[\left.\frac{\partial^2 l}{\partial \theta^2}\right|_{\theta=\theta_0}\right],
\end{aligned} \tag{2.3}$$

where θ_0 is the maximum likelihood estimate and the subscript 0 is to emphasize that the expectation is taken with respect to the density function $f(y; \theta_0)$. On the other hand, simple calculations show that, for an exponential family distribution, we have:

$$\mathbb{E}\left[\left(\frac{\partial l}{\partial \theta}\right)\right] = \frac{\mathbb{E}[Y] - b'(\theta)}{a(\phi)}.$$

Thus, we have:

$$b'(\theta) = \mu.$$

Using the second and the third property of (2.3), we have then:

$$\operatorname{var}(Y) = b''(\theta)a(\phi).$$

Until now, we have made no restrictions about the function $a(\cdot)$. Actually, it is usual to consider distribution where $a(\phi)$ is of the form $a(\phi) = \phi/r$, which allows heteroscedastic models and covers all the cases of practical interest (Wood, 2011, p.63). More complex forms of $a(\cdot)$ won't be considered here. Usually r is set to 1. Thus in this case the variance can be written as:

$$\mathrm{var}(Y) = \frac{b''(\theta)\phi}{r}.$$

Since θ depends on the mean μ, we can always define a function $V(\cdot)$ such that

$$\mathrm{var}(Y) = V(\mu)\phi.$$

This function is called variance function. Under regularity conditions, and since we have $b'(\theta) = \mu$, we have:

$$V(\mu) = b''((b')^{-1}(\mu)),$$

where, $(b')^{-1}(\cdot)$ is the reciprocal function of the first derivative with respect to θ of the function $b(\cdot)$.

On one hand we have $b'(\theta) = \mu$ and the other hand, we have $g(\mu) = \eta = \mathbf{x}^t\boldsymbol{\beta}$. In the case where $\eta = \theta$, that is $b'(\cdot) = g^{-1}(\cdot)$, the link function is said to be *canonical.* Therefore, for every exponential distribution, there is a canonical link function. For the binomial distribution, the canonical link function is the logit function. Other link functions are available for binary responses such as the probit or the c-loglog. The choice of the link function is not discussed since it is not the purpose of this work.

2.3 Penalized iterative reweighted least squares algorithm

The goal of this algorithm is to approximate the optimum of the functional given in (2.2). To achieve this, we approximate at each step of the algorithm the penalized negative log-likelihood (up to a scale factor) with a quadratic form and we solve a simpler least square problem. The main problem is that the penalized log-likelihood optimisation problem is complex to solve since the link function g is usually not linear (except in the Gaussian case) and therefore, we use this iterative procedure. Let λ be a positive smoothing parameter. For a given vector $\boldsymbol{\mu}^j$, we define:

$$\boldsymbol{\eta}^j = g(\boldsymbol{\mu}^j),\ \mathbf{X} = \begin{bmatrix} \mathbf{x}_1^t \\ \vdots \\ \mathbf{x}_n^t \end{bmatrix},\ \ \mathbf{W}^j = \mathrm{diag}\left(\frac{1}{V(\boldsymbol{\mu}^j)(g'(\boldsymbol{\mu}^j))^2}\right), \mathbf{f}_n = (f(\mathbf{p_1}), \dots, f(\mathbf{p}_n))^t,$$

where $V(\cdot)$ is the variance function associated to the distribution of the observations, and g is the link function[1]. We can then approximate the solution of minimizing (2.2) as the minimum of (see section 4 for a rigorous justification):

$$\tilde{J}_\lambda(\boldsymbol{\beta}, f) = \|(\mathbf{W}^j)^{1/2}(\mathbf{z}^j - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}_n)\|^2 + \lambda \int_\Omega (\Delta f)^2,$$

where $\mathbf{z}^j = \boldsymbol{\eta}^j + g'(\boldsymbol{\mu}^j)(\mathbf{y} - \boldsymbol{\mu}^j)$ is called "pseudo-data", i.e., the first order approximation of $g(\mathbf{y})$, assuming that $\mathbf{y}$ is in the neighbourhood of $\boldsymbol{\mu}^j$.

Given $\boldsymbol{\mu}^j$, the algorithm is the following:

1. Compute $\boldsymbol{\eta}^j$, $\mathbf{z}^j$ and $\mathbf{W}^j$.

[1] When a univariate function, such as g here is applied to a vector, it is meant to apply element wise.

2. Solve the problem of finding $\tilde{\boldsymbol{\beta}}^j$ and $\tilde{f}^j$ that minimize $\|(\mathbf{W}^j)^{1/2}(\mathbf{z}^j - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}_n)\|^2 + \lambda \int_\Omega (\Delta f)^2$ (or a finite dimensional approximation of this functional)

3. Set $\boldsymbol{\mu}^{j+1} = g^{-1}(\mathbf{X}\tilde{\boldsymbol{\beta}}^j + \tilde{\mathbf{f}}_n^j)$

For the initialization, we set $\boldsymbol{\mu}^0 = \mathbf{y}$ except in the logistic case where $\boldsymbol{\mu}^0 = 1/2(\mathbf{y} + 1/2)$.

There are various convergence criteria to detect convergence of the process, here we use one based on a sufficiently small variation of the objective function between two iterations.

Such an iterative process is usual in the context of GAM and the particularity of our method is in finding an efficient way to solve the minimization problem at the step 2 of the algorithm.

2.4 Weighted penalized least square minimization problem

Given the current (diagonal) weight matrix $\mathbf{W}$ and a design matrix $\mathbf{X}$, we want to minimize the following functional with respect to $(\boldsymbol{\beta}, f)$:

$$\tilde{J}_\lambda(\boldsymbol{\beta}, f) = \|\mathbf{W}^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}_n)\|^2 + \lambda \int_\Omega (\Delta f)^2. \tag{2.4}$$

We will add a tilde over variables which achieve the minimum of the functional $\tilde{J}_\lambda$. First, we need to recall a classical result known as the Lax-Milgram lemma, which will be used later:

Proposition 1
Let $\mathcal{F}$ be a Hilbert space, $G(\cdot,\cdot) : \mathcal{F} \times \mathcal{F} \to \mathbb{R}$ a continuous and coercive bilinear form, $F(\cdot) : \mathcal{F} \to \mathbb{R}$ a linear and continuous functional. Then there exists a unique solution to the problem:

$$\text{find } u \in \mathcal{F} \text{ such that: } G(u, v) = F(v), \quad \forall v \in \mathcal{F}.$$

Moreover, if $G(\cdot,\cdot)$ is symmetric, then u characterize the unique minimizer in $\mathcal{F}$ of the functional $J(\cdot) : \mathcal{F} \to \mathbb{R}$, defined as

$$J(v) = G(v, v) - 2F(v).$$

The Lax-Milgram lemma states an equivalence between two problems. We will use this fact to solve a equivalent problem instead of minimizing (2.4). Now we can show the following result:

Proposition 2
If $\mathbf{X}$ is a full-rank matrix, and if $\mathbf{W}_{ii} \neq 0,\ i = 1, \ldots, n$, the solution $(\tilde{\boldsymbol{\beta}}, \tilde{f})$ to the minimization problem (2.4) satisfies:

1. $\tilde{\boldsymbol{\beta}} = (X^tWX)^{-1}X^tW(\mathbf{z} - \tilde{\mathbf{f}}_n)$,

2. $\tilde{f}$ satisfies:

$$\mathbf{u}_n^t\mathbf{Q}\ \tilde{\mathbf{f}}_n + \lambda \int_\Omega (\Delta u)(\Delta \tilde{f}) = \mathbf{u}_n^t\mathbf{Q}\ \mathbf{z}, \qquad \forall u \in \mathcal{F}.$$

where $\mathbf{Q} = (\mathbf{I} - \mathbf{H})^t\mathbf{W}(\mathbf{I} - \mathbf{H})$ and $\mathbf{H} = \mathbf{X}(\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}$. Moreover, the solution $\left(\tilde{\boldsymbol{\beta}}, \tilde{f}\right)$ exists and is unique.

Proof. First of all, given $\overline{f}$, the unique minimizer of the functional $\tilde{J}_\lambda(\boldsymbol{\beta}, \overline{f})$ is given by:

$$\tilde{\boldsymbol{\beta}}(\overline{f}) = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}(\mathbf{z} - \overline{\mathbf{f}}_n). \tag{2.5}$$

To show that, we take the derivative of $\tilde{J}_\lambda(\boldsymbol{\beta}, f)$ with respect to $\boldsymbol{\beta}$:

$$\frac{\partial \tilde{J}_\lambda(\boldsymbol{\beta}, \overline{f})}{\partial \boldsymbol{\beta}} = -2\mathbf{X}^t\mathbf{W}(\mathbf{z} - \overline{\mathbf{f}_n}) + (\mathbf{X}^t\mathbf{W}\mathbf{X})\boldsymbol{\beta}.$$

Since $\mathbf{X}$ is assumed to be a full-rank matrix and since $\mathbf{W}$ is invertible ($\mathbf{W}_{ii}$ is assumed to be strictly positive, because $\mathbf{W}_{ii} \neq 0$ and $\mathbf{W}_{ii} \geq 0$ by construction), $\mathbf{X}^t\mathbf{W}\mathbf{X}$ is invertible. Finally the necessary condition $\partial \tilde{J}_\lambda(\tilde{\boldsymbol{\beta}}, \overline{f})/\partial \boldsymbol{\beta} = 0$ is satisfied if and only if $\tilde{\boldsymbol{\beta}}$ is given by (2.5). Since for fixed $\overline{f}$, $\tilde{J}_\lambda(\boldsymbol{\beta}, \overline{f})$ is clearly convex, $\tilde{\boldsymbol{\beta}}$ is a minimum. For vector and matrices derivation rules, see for example Felippa (2012).

Setting $\mathbf{H} = \mathbf{X}(\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}$, $\mathbf{Q} = (\mathbf{I} - \mathbf{H})^t\mathbf{W}(\mathbf{I} - \mathbf{H})$, and plugging $\tilde{\boldsymbol{\beta}}$ into the objective function, we have the following form of the functional:

$$\tilde{J}_\lambda(\overline{f}) = \mathbf{z}^t\mathbf{Q}\ \mathbf{z} - 2\overline{\mathbf{f}}_n\mathbf{Q}\ \mathbf{z} + \overline{\mathbf{f}}_n^t\mathbf{Q}\ \overline{\mathbf{f}}_n + \lambda \int_\Omega (\Delta \overline{f})^2. \tag{2.6}$$

Since we want to optimize this functional with respect to the function $\overline{f}$ only, we can rewrite the problem (2.6) as:

$$\text{find } \tilde{f} \in \mathcal{F} \text{ that minimizes: } \overline{J}_\lambda(\overline{f}) = \overline{\mathbf{f}}_n^t\mathbf{Q}\ \overline{\mathbf{f}}_n + \lambda \int_\Omega (\Delta \overline{f})^2 - 2\overline{\mathbf{f}}_n\mathbf{Q}\ \mathbf{z}, \tag{2.7}$$

which could be written as $G(\overline{f}, \overline{f}) - 2F(\overline{f})$, where G is a symmetric, coercive and continuous, bilinear form on $\mathcal{F} \times \mathcal{F}$ and F a linear and continuous functional on $\mathcal{F}$. Using the Lax-Milgram lemma and thanks to the symmetry of $G(\cdot, \cdot)$, the problem (2.7) is equivalent to the problem of

$$\text{find } \tilde{f} \in \mathcal{F} \text{ such that: } G(\tilde{f}, u) = F(u) \qquad \forall u \in \mathcal{F}.$$

The hypotheses of the Lax-Milgram lemma are fulfilled since the bilinear form G is coercive and continuous and since the linear form F is also continuous. The verification of these hypotheses are very simple and we do not address this issue. A rigorous proof of the equivalence of the problems and of the uniqueness of $\tilde{f}$, applied to this particular case can also be found in Azzimonti (2013). In our case, we can identify the bilinear form G as:

$$G(v, u) = \mathbf{u}_n^t\mathbf{Q}\ \mathbf{v}_n + \lambda \int_\Omega (\Delta u)(\Delta v),$$

where $\mathbf{u}_n = (u(\mathbf{p_1}), \ldots, u(\mathbf{p_n}))^t$, and the linear functional F as:

$$F(v) = \mathbf{v}_n^t\mathbf{Q}\ \mathbf{z}.$$

So finally the solution of minimizing (2.7) is given by:

$$\text{find } \tilde{f} \in \mathcal{F} \text{ such that: } \mathbf{u}_n^t\mathbf{Q}\ \tilde{\mathbf{f}}_n + \lambda \int_\Omega (\Delta u)(\Delta \tilde{f}) = \mathbf{u}_n^t\mathbf{Q}\ \mathbf{z} \qquad \forall u \in \mathcal{F}. \tag{2.8}$$

The uniqueness of the minimizer is also stated by the Lax-Milgram lemma. We conclude that $\tilde{f}$ exists and is unique hence $\tilde{\boldsymbol{\beta}}$ exists and is unique. □

Chapter 3

Finite Elements Formulation

3.1 A very short introduction

The Finite Elements Method (FEM) is a widely used method to compute numerical solutions of physical problems driven by partial differential equations. The idea is to find the best approximation belonging to a given finite dimensional space. First of all, we divide the domain, embedded in $\mathbb{R}^2$, into triangular subdomains. The set of triangles resulting from this process is called a mesh. The mesh generation is a very complicated procedure and a abundant literature is dedicated to this subject (see, for example Cheng et al. (2013)). Then we impose that finite element solution must be polynomial in x and y (in this case, polynomial of order 1 or 2) over all triangles and continuous across edges and vertices. For every node $\boldsymbol{\xi}_j$, we define a function ψ_j which is piecewise polynomial of the prescribed order and such that we have $\psi_i(\boldsymbol{\xi}_j) = \delta_{ij}, \quad \forall j = 1, \ldots, K$. The dimension of the finite element space is then the number of nodes, that is K. This kind of basis is referred to as *Lagrangian finite elements.* We will use a finite element basis to get a finite dimensional approximation $\mathcal{F}_K$ of the functional space $\mathcal{F}$. Then, we will compute the "best" approximation $\hat{f} \in \mathcal{F}_K$ of the function f. The meaning of the word "best" is the following: since we are looking for the solution of the problem (2.7) and that $G(u, v)$ is bilinear, symmetric and positive definite over $\mathcal{F} \times \mathcal{F}$, it can be proven that the error is orthogonal to the solution in the sense of the inner product $G(., .)$, that is the finite element solution is the projection of the true solution on $\mathcal{F}_K$. This property is called *Galerkin orthogonality.*

For more informations about FEM, see for example Quarteroni (2009).

3.2 Reformulation of the problem

We have seen that we have to solve at each step of the PIRLS algorithm a minimisation problem. It has been shown that such a problem can be reduced to solving a linear system (Sangalli et al., 2013). We describe this procedure in the following discussion.

First of all, we recall the minimization problem to be solved at each step of the algorithm:

$$\min_{(\boldsymbol{\beta}, f) \in \mathbb{R}^q \times \mathcal{F}} \tilde{J}_\lambda(\boldsymbol{\beta}, f) = \|\mathbf{W}^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}_n)\|^2 + \lambda \int_\Omega (\Delta f)^2,$$

where $\mathbf{W}$ is the diagonal weight matrix and $\mathbf{z}$ is the pseudo-data at the current step of the algorithm. The functional $\tilde{J}_\lambda(\boldsymbol{\beta}(f), f)$ can be expressed as $G(f, f) - 2F(f)$, as previously stated (see equation 2.7). The minimization problem is then equivalent to solve the problem $G(f, u) = F(u)$, $\forall u \in \mathcal{F}$. This equivalent problem is called *weak formulation.* We will describe the computations yielding to the solution of this problem.

First of all, given a function $\overline{f}$, the maximizer of $J_\lambda(\boldsymbol{\beta}, \overline{f})$ is given by

$$\tilde{\boldsymbol{\beta}} = (X^tWX)^{-1}X^tW(\mathbf{z} - \overline{\mathbf{f}}_n).$$

Moreover, $\tilde{f}$ must satisfy (2.7).

Now, we can recall a simple property of real hilbertian vector spaces. Let $\mathbf{a}, \mathbf{b} \in V$, where V is a real hilbertian vector space. We have the following property:

$$\mathbf{a} = \mathbf{b} \Leftrightarrow \langle \mathbf{a}, \mathbf{c} \rangle = \langle \mathbf{b}, \mathbf{c} \rangle, \quad \forall \mathbf{c} \in V. \tag{3.1}$$

Therefore, we can characterize any vector of $\mathcal{F}$ since $\mathcal{F}$ is a real vector space. Thanks to this characterization, we will reformulate the problem (2.8) substituting $\Delta\tilde{f}$ by another function $\tilde{g}$. Problem (2.8) then becomes:

$$\begin{cases} \mathbf{u}_n^t\mathbf{Q}\ \tilde{\mathbf{f}}_n + \lambda \displaystyle\int_\Omega (\Delta u)\tilde{g} = \mathbf{u}_n^t\mathbf{Q}\ \mathbf{z}, & \forall u \in \mathcal{F}, \\ \displaystyle\int_\Omega (\Delta \tilde{f})\ v = \int_\Omega \tilde{g}\ v, & \forall v \in \mathcal{F}. \end{cases} \tag{3.2}$$

This justification is only formal since the functional spaces used are Sobolev spaces. Actually all the functions considered are defined up to a set of null measure. We only want to give an intuition to the reader. For a more rigorous development, see Azzimonti (2013, chapter 2).

From the Green formula we get:

$$\int_\Omega (\Delta u)\tilde{g} = -\int_\Omega \nabla u \cdot \nabla \tilde{g} + \int_{\partial\Omega} \tilde{g}\frac{\partial \mathbf{u}}{\partial \mathbf{n}},$$

where $\mathbf{n}$ denotes the outward-pointing normal vector to the boundary. In our case, we consider functions whose normal derivative vanishes on the boundary, hence we have $\displaystyle\int_{\partial\Omega} \tilde{g}\frac{\partial u}{\partial \mathbf{n}} = 0$. We get then:

$$\begin{cases} \mathbf{u}_n^t\mathbf{Q}\ \tilde{\mathbf{f}}_n - \lambda \displaystyle\int_\Omega \nabla u \cdot \nabla \tilde{g} = \mathbf{u}_n^t\mathbf{Q}\ \mathbf{z} & \forall u \in \mathcal{F}, \\ -\displaystyle\int_\Omega \nabla \tilde{f} \cdot \nabla v = \int_\Omega \tilde{g}\ v & \forall v \in \mathcal{F}. \end{cases} \tag{3.3}$$

In this setting, we will find two different functions, $\tilde{f}$ and $\tilde{g}$. In a finite dimensional space, (3.3) results in a sparse linear problem.

These computations, apparently easy, are only formal. To justify this development, strong and subtle hypotheses are required. Such a rigorous approach is done in Azzimonti (2013).

3.3 From weak formulation to linear systems

We will show how the weak formulation problem leads to a linear problem using finite elements. Let $\hat{\mathcal{F}}$ be a K dimensional finite element subspace of $\mathcal{F}$. Then both equations of the problem (3.3) could be written in the form:

$$\tilde{G}(h, u) = \tilde{F}(u) \qquad \forall \tilde{u} \in \mathcal{F}_K, \tag{3.4}$$

where $\tilde{G}$ is a symmetric, bilinear form and $\tilde{F}$ is linear functional. Since $\mathcal{F}_K$ is finite dimensional, there exists a basis, $\{\psi_1, \ldots, \psi_K\}$ of the vector space $\mathcal{F}_K$. If (3.4) is satisfied for any element of the basis, thanks to the bilinearity of $\tilde{G}$ and to the linearity of $\tilde{F}$, then it will be satisfied for any element of $\mathcal{F}_K$. We can expand any function $h \in \mathcal{F}_K$ in the basis of $\mathcal{F}_K$ as:

$$h(\mathbf{x}) = \sum_{i=1}^{K} h_i \psi_i(\mathbf{x}), \quad h_i \in \mathbb{R},\ \forall i = 1, \dots, K,$$

which can also be written as:

$$\tilde{h} = \mathbf{h}^t \boldsymbol{\psi}, \quad \text{where } \mathbf{h} = (h_1, \dots, h_K)^t, \text{ and } \boldsymbol{\Psi} = (\psi_1(x), \dots, \psi_K(x))\,.$$

Finally, we get the following linear system:

$$\sum_{i=1}^{K} h_i\ \tilde{G}(\psi_i, \psi_j) = \tilde{F}(\psi_j), \qquad \forall j = 1, \dots, K.$$

Using this, we only have to compute the values $\tilde{G}(\psi_i, \psi_j)$ and $\tilde{F}(\psi_j), \quad i, j = 1, \dots, K$ and solve the system.

In our case, for the first equation, we have:

$$G_1(s,t) = \mathbf{s}_n^t \mathbf{Q}\ \mathbf{t}_n - \lambda \int_\Omega \nabla s \cdot \nabla t, \qquad F_1(s) = \mathbf{s}_n^t \mathbf{Q}\ \mathbf{z}.$$

and for the second one:

$$G_2(s,t) = \int_\Omega \nabla s \cdot \nabla t + \int_\Omega s\ t, \qquad F_2(s) = 0.$$

Instead of Ω, we will use $\Omega_\mathcal{T}$, a polygonal approximation of the domain, resulting in a regular triangulation of Ω.

Finally, this allows us to construct the linear system. We define the $K \times K$ matrices $\mathbf{R}_0$ and $\mathbf{R}_1$, called mass and stiffness matrix respectively, as:

$$(\mathbf{R}_0)_{ij} = \int_{\Omega_\mathcal{T}} \psi_i\ \psi_j, \qquad (\mathbf{R}_1)_{ij} = \int_{\Omega_\mathcal{T}} \nabla\psi_i \cdot \nabla\psi_j.$$

We introduce the order K block matrix $\mathbf{L}$ and the $K \times n$ block matrix $\mathbf{P}_1$, defined as:

$$\mathbf{L} = \left[\begin{array}{c|c} \mathbf{Q} & \mathbf{O}_{n\times(K-n)} \\ \hline \mathbf{O}_{(K-n)\times n} & \mathbf{O}_{(K-n)\times(K-n)} \end{array}\right] \text{ and } \mathbf{P}_1 = \left[\begin{array}{c} \mathbf{I}_n \\ \hline \mathbf{O}_{(K-n)\times n} \end{array}\right].$$

Finally, if $\tilde{\mathbf{f}}$ and $\tilde{\mathbf{g}}$ denote the vectors of coefficients of the expansion of the functions $\tilde{f}$ and $\tilde{g}$ in the basis $\boldsymbol{\psi}$, we can write the linear system corresponding to the problem (3.4) as:

$$\begin{bmatrix} -\mathbf{L} & \lambda\mathbf{R}_1 \\ \lambda\mathbf{R}_1 & \lambda\mathbf{R}_0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{f}} \\ \tilde{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} -\mathbf{L}\mathbf{P}_1\mathbf{z} \\ \mathbf{0} \end{bmatrix}. \tag{3.5}$$

For computational reasons, we have symmetrised the problem. In this context, $\tilde{g} = \tilde{\mathbf{g}}^t \boldsymbol{\psi}$ is an approximation of the function $\Delta\tilde{f}$. In order to get the vector $\tilde{\mathbf{f}}_n$ which is the vector of the values of $\tilde{f}$ evaluated at the points $\mathbf{p}_1, \dots, \mathbf{p}_n$, we need the values of the coefficients of the vector $\tilde{\mathbf{f}}$ corresponding to these points. We choose a triangulation such that the vertices of the triangles coincide with the locations of the observation. Hence thanks to the fact that, for a given node $\boldsymbol{\xi}_i$, we have $\psi_j(\boldsymbol{\xi}_i) = \delta_{ij}$, where δ_{ij} is the Kronecker symbol, we have $\tilde{f}(\mathbf{p}_i) = \tilde{\mathbf{f}}_i$. We can then estimate the value of the functional that we have minimized at the current estimates as:

$$\tilde{J}_\lambda(\tilde{\boldsymbol{\beta}}, \tilde{\mathbf{f}}, \tilde{\mathbf{g}}) = \left(\mathbf{z} - \mathbf{X}\tilde{\boldsymbol{\beta}} - \tilde{\mathbf{f}}_n\right)^t \mathbf{Q}\ \left(\mathbf{z} - \mathbf{X}\tilde{\boldsymbol{\beta}} - \tilde{\mathbf{f}}\right) + \lambda\ \tilde{\mathbf{g}}^t\ \mathbf{R}_1\ \tilde{\mathbf{g}}$$

3.4 Explicit form of the estimates

Thanks to (3.5), we can give the explicit forms of the estimates:

$$\tilde{\mathbf{f}} = \left(\mathbf{L} + \lambda \mathbf{R}_1 \mathbf{R}_0^{-1} \mathbf{R}_1\right)^{-1} \mathbf{L}\mathbf{P}_1 \mathbf{z}.$$

Since we have chosen a mesh whose internal nodes coincide with the data, writing, $\mathbf{P}_2 = [\mathbf{I}_n \ \mathbf{O}_{n\times K-n}]$, we have:

$$\tilde{\mathbf{f}}_n = \mathbf{P}_2 \tilde{\mathbf{f}} = \mathbf{P}_2 \left(\mathbf{L} + \lambda \mathbf{R}_1 \mathbf{R}_0^{-1} \mathbf{R}_1\right)^{-1} \mathbf{L}\mathbf{P}_1 \mathbf{z}.$$

Replacing that value in (2.5), we get:

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^t \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{W} (\mathbf{z} - \tilde{\mathbf{f}}_n) = \left((\mathbf{X}^t \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{W} - \mathbf{P}_2 \left(\mathbf{L} + \lambda \mathbf{R}_1 \mathbf{R}_0^{-1} \mathbf{R}_1\right)^{-1} \mathbf{L}\mathbf{P}_1 \right) \mathbf{z}.$$

We can then conclude that both estimates, $\tilde{\boldsymbol{\beta}}$ and $\tilde{\mathbf{f}}$ depend linearly on the pseudo-data $\mathbf{z}$. However, since the pseudo-data do not depend linearly of the data themselves, we cannot use this fact to give closed expressions for the mean and for the variance, as it is done in Sangalli et al. (2013).

We can finally define the hat matrix $\mathbf{R}$ for the generalized additive model (see Hastie and Tibshirani (1990, p. 156)) as the matrix such that, at convergence:

$$\hat{\boldsymbol{\eta}} = \mathbf{R}\mathbf{z}, \tag{3.6}$$

where $\hat{\boldsymbol{\eta}}$ is the natural parameter at the convergence and $\mathbf{z}$ is, in this context, the last pseudo-data, i.e., the pseudo data at the convergence. In this case, the hat matrix is given by:

$$\mathbf{R} = \left((\mathbf{X}^t \mathbf{W} \mathbf{X})^{-1} \mathbf{X}^t \mathbf{W} + (\mathbf{I} - \mathbf{X}) \mathbf{P}_2 \left(\mathbf{L} + \lambda \mathbf{R}_1 \mathbf{R}_0^{-1} \mathbf{R}_1\right)^{-1} \mathbf{L}\mathbf{P}_1 \right).$$

This hat matrix is important because it is the matrix which define the equivalent degree of freedom. See section 5.2 for more details.

3.5 Generalization to a mesh independent of the data points

We saw that it might be convenient to use a mesh whose internal nodes are exactly the location of the data points. Obviously, it is useless to refine the mesh where there is no data points and at the contrary, it seems reasonable to have a finer mesh where there is higher density of data points. In this case the finite element subspace becomes then dependent of the location points.

If we use a constrained mesh where the internal nodes coincide with the location of the data, setting that for $i = 1, \dots, n$, ψ_i is the basis function corresponding to the location $\mathbf{p}_i$, the unknown $\tilde{\mathbf{f}}_n$, defined as:

$$\tilde{\mathbf{f}}_n = (\tilde{f}(\mathbf{p}_1), \dots, \tilde{f}(\mathbf{p}_n))^t,$$

is also the n first components of the unknown vector $\tilde{\mathbf{f}}$. We can then easily solve the system (3.5).

On the contrary, if the location points do not coincide with nodes, we do not have the fact that $\tilde{\mathbf{f}}_i = \tilde{\mathbf{f}}(\mathbf{p}_i), \forall i = 1, \dots, n$. So we have to compute the value of $\mathbf{f}_n$ related to the vector $\tilde{\mathbf{f}}$. Let $\boldsymbol{\Psi}$ be the $n \times K$ matrix defined as:

$$(\boldsymbol{\Psi})_{ij} = \psi_j(\mathbf{p}_i).$$

Then $\tilde{\mathbf{f}}_n = \boldsymbol{\Psi}\tilde{\mathbf{f}}$. The system becomes (3.5):

$$\begin{bmatrix} -\boldsymbol{\Psi}^t\ \mathbf{Q}\ \boldsymbol{\Psi} & \lambda\mathbf{R}_1 \\ \lambda\mathbf{R}_1 & \lambda\mathbf{R}_0 \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{f}} \\ \tilde{\mathbf{g}} \end{bmatrix} = \begin{bmatrix} -\boldsymbol{\Psi}^t\mathbf{Q}\ \mathbf{z} \\ \mathbf{0} \end{bmatrix}. \tag{3.7}$$

Indeed, the penalization matrix is not influenced by this modification and the other blocks of the matrix (3.7) are identical to those of (3.5).

The computation of the estimates then becomes:

$$\tilde{\mathbf{f}} = \left(\boldsymbol{\Psi}^t\ \mathbf{Q}\ \boldsymbol{\Psi} + \lambda\mathbf{R}_1\mathbf{R}_0^{-1}\mathbf{R}_1\right)^{-1}\boldsymbol{\Psi}^t\mathbf{Q}\mathbf{z},$$

and

$$\tilde{\boldsymbol{\beta}} = (\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W}(\mathbf{z} - \tilde{\mathbf{f}}_n) = \left((\mathbf{X}^t\mathbf{W}\mathbf{X})^{-1}\mathbf{X}^t\mathbf{W} - \boldsymbol{\Psi}\left(\boldsymbol{\Psi}^t\ \mathbf{Q}\ \boldsymbol{\Psi} + \lambda\mathbf{R}_1\mathbf{R}_0^{-1}\mathbf{R}_1\right)^{-1}\boldsymbol{\Psi}^t\mathbf{Q}\right)\mathbf{z}.$$

As in the case of data constrained mesh, we can easily define a matrix $\mathbf{R}$ such that $\hat{\boldsymbol{\eta}} = R\mathbf{z}$, where $\mathbf{z}$ is the pseudo-data at convergence.

Chapter 4

Justification of the PIRLS algorithm and penalized log-likelihood

Since our model can be seen as a generalized additive model, we will use methods developed in this context (Hastie and Tibshirani, 1986, 1990; Wood, 2006). In particular, the PIRLS algorithm is suitable for our model. PIRLS aims to compute the maximum penalized likelihood estimates $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\hat{f} \in \mathcal{F}_K$. Let $m(.,.)$ be a bilinear, symmetric and positive definite form on $\mathcal{F} \times \mathcal{F}$ that represents the penalization. In our case, we are seeking to find the maximizer of the following penalized log-likelihood:

$$\mathcal{L}_p(\boldsymbol{\beta}, f) = \mathcal{L}(\boldsymbol{\beta}, f) - \lambda \; m\left(f, f\right), \tag{4.1}$$

where, $\mathcal{L}(\boldsymbol{\beta}, f)$ is the log-likelihood of the model. Until now, we do not have made precise assumptions about the functional space $\mathcal{F}$ in which lies the function f. A necessary condition to ensure the existence of a solution to the problem of minimizing (4.1) is that $f \in H^2(\Omega)$. Moreover, suitable boundary conditions are required to ensure well-posedness of the problem. For more precise references about the functional framework, we refer to Azzimonti (2013). In particular, the use of a mixed finite elements approach for a 4 order minimization problem is discussed.

The problem of maximizing the penalized log-likelihood is non-trivial since we deal with functional spaces. As already mentioned, solving this problem is a two step process. The two steps are the discretization and the optimization. If we first discretize and then optimize, we seek an optimum in a finite dimensional space, which is a well-known problem. We can also first optimize in the functional space and then find a finite dimensional approximation of this optimum. These approaches are well-known in the context of optimal control theory. In this work, we develop both approaches. Obviously, the so-called optimize-discretize approach is more complex since the optimization is done in a functional space. In our case, this allows us to reformulate the problem in order to use a similar approach to the one used in Sangalli et al. (2013). Without that, we would not have obtained the problem of minimizing (2.4).

4.1 Functional derivation of the PIRLS algorithm

We know that the maximum values $(\hat{\boldsymbol{\beta}}, \hat{f})$ of the functional (4.1) satisfy the following equations:

$$\begin{cases} \dfrac{\partial \mathcal{L}(\hat{\boldsymbol{\beta}}, \hat{f})}{\partial \beta_j} & = & 0, \quad \forall j = 1, \ldots, q, \\ \lim\limits_{t \to 0} \dfrac{1}{t} \left[\mathcal{L}(\hat{\boldsymbol{\beta}}, \hat{f} + tu) - \mathcal{L}(\hat{\boldsymbol{\beta}}, \hat{f}) \right] - \lambda \; m(u, \hat{f}) & = & 0 \quad \forall u \in \mathcal{F}. \end{cases} \tag{4.2}$$

Actually, since $\hat{f} \in \mathcal{F}$, and $\mathcal{F}$ is infinite dimensional, we would have to consider the maximization problem in an infinite dimensional space. We aim to compute the Gâteaux derivative of the penalized log-likelihood to get the variational formulation of the minimization problem seen in (2.4). The Gâteaux derivative is defined for functions in locally convex topological vector spaces, such as Banach spaces and is a generalization of the concept of directional derivative. Actually, the functional spaces used in our context are Sobolev spaces. The log-likelihood function $\mathcal{L}$ depends on $\boldsymbol{\beta}$ and f because the natural parameter θ depends on $\boldsymbol{\beta}$ and f. Given any function $u \in \mathcal{F}$, and any real number t, we then have:

$$\begin{aligned}
&\mathcal{L}(\boldsymbol{\beta}, f+tu) - \mathcal{L}(\boldsymbol{\beta}, f) = \\
&= \sum_{i=1}^{n} \frac{r_i}{\phi} \left[(y_i\theta_i(\boldsymbol{\beta}, f+tu) - b(\theta_i(\boldsymbol{\beta}, f+tu)) - y_i\theta_i(\boldsymbol{\beta}, f) + b(\theta_i(\boldsymbol{\beta}, f))) \right] \\
&= \sum_{i=1}^{n} \frac{r_i}{\phi} (y_i\theta_i(\boldsymbol{\beta}, f+tu) - y_i\theta_i(\boldsymbol{\beta}, f) - (b(\theta_i(\boldsymbol{\beta}, f+tu)) - b(\theta_i(\boldsymbol{\beta}, f)))\,.
\end{aligned}$$

Dividing this expression by t and taking the limit as t tends to 0 gives the Gâteaux derivative. We have then:

$$\begin{aligned}
&\frac{\partial \mathcal{L}(\boldsymbol{\beta}, u)}{\partial f} = \lim_{t\to 0} \frac{1}{t} \left[\mathcal{L}(\boldsymbol{\beta}, f+tu) - \mathcal{L}(\boldsymbol{\beta}, f) \right] \\
&= \lim_{t\to 0} \frac{1}{t} \left(\sum_{i=1}^{n} \frac{r_i}{\phi} \left[y_i\theta_i(\boldsymbol{\beta}, f+tu) - y_i\theta_i(\boldsymbol{\beta}, f) - (b(\theta_i(\boldsymbol{\beta}, f+tu)) - b(\theta_i(\boldsymbol{\beta}, f)) \right] \right) \\
&= \sum_{i=1}^{n} \frac{r_i}{\phi} \left[y_i \frac{\partial \theta_i(\boldsymbol{\beta}, u)}{\partial f} - b'(\theta_i(\boldsymbol{\beta}, f)) \frac{\partial \theta_i(\boldsymbol{\beta}, u)}{\partial f} \right].
\end{aligned}$$

where $b'(\cdot)$ stands for $\frac{\partial b}{\partial \theta}$. We have then to compute the value of $\dfrac{\partial \theta_i(\boldsymbol{\beta}, u)}{\partial f}$. Recalling that we have $\mathbb{E}\left[Y_i\right] = \mu_i = b'(\theta_i)$ and taking the derivative, we get:

$$\frac{\partial \mu}{\partial \theta} = b''(\theta) \Rightarrow \frac{\partial \theta}{\partial \mu} = \frac{1}{b''(\theta)}$$

and hence:

$$\frac{\partial \theta_i(\boldsymbol{\beta}, u)}{\partial f} = \frac{1}{b''(\theta_i)} \frac{\partial \mu_i(\boldsymbol{\beta}, u)}{\partial f}.$$

We can conclude that we have:

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}, u)}{\partial f} = 0 \Leftrightarrow \sum_{i=1}^{n} \frac{r_i}{\phi} \frac{(y_i - b'(\theta_i))}{b''(\mu_i)} \frac{\partial \mu_i(\boldsymbol{\beta}, u)}{\partial f} = \sum_{i=1}^{n} \frac{(y_i - b'(\theta_i))}{\mathrm{var}(Y_i)} \frac{\partial \mu_i(\boldsymbol{\beta}, u)}{\partial f} = 0.$$

Since we have $\mathrm{var}(Y_i) = V(\mu_i)\phi$ and $b'(\theta_i) = \mu_i$, we have:

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta}, u)}{\partial f} = 0 \Leftrightarrow \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)} \frac{\partial \mu_i(\boldsymbol{\beta}, u)}{\partial f} = 0.$$

On the other hand, we have to compute the derivative of the functional $\mathcal{L}_p$ with respect to β. Hence we have:

$$\frac{\partial \mathcal{L}_p(\boldsymbol{\beta})}{\partial \beta_j} = \sum_{i=1}^{n} \frac{r_i}{\phi} \left(y_i \frac{\partial \theta_i}{\partial \beta_j} - b'(\theta_i) \frac{\partial \theta_i}{\partial \beta_j} \right).$$

Since we have

$$\frac{\partial \theta_i}{\partial \beta_j} = \frac{\partial \theta_i}{\partial \mu_i} \frac{\partial \mu_i}{\partial \beta_j} = \frac{1}{b''(\theta_i)} \frac{\partial \mu_i}{\partial \beta_j},$$

We finally get:

$$\frac{\partial \mathcal{L}(\boldsymbol{\beta})}{\partial \beta_j} = 0 \Leftrightarrow \sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)} \frac{\partial \mu_i}{\partial \beta_j} = 0.$$

Putting all these results together and assuming that $V(\mu_i)$ is fixed, the solution of (4.2) is equivalent to finding $\boldsymbol{\mu}$ that satisfies

$$\left\{ \begin{array}{rcl} \displaystyle\sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)} \frac{\partial \mu_i}{\partial \beta_j} & = & 0, \quad \forall j = 1, \ldots, q, \\ \displaystyle\sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)} \frac{\partial \mu_i(\boldsymbol{\beta}, u)}{\partial f} + \lambda \, m(f, u) & = & 0 \quad \forall u \in \mathcal{F}, \end{array} \right.$$

which is itself equivalent to find the solution of minimizing the functional $\mathcal{S}$:

$$\mathcal{S} = \| \mathbf{V}^{-1/2} \, (\mathbf{y} - \boldsymbol{\mu}) \, \|^2 + \frac{\lambda}{2} \, m(f, f),$$

where $\mathbf{V} = \mathrm{diag}(V(\mu_1), \ldots, V(\mu_n))$, but considering this fixed and hence suggesting an iterative computation.

We then develop a first order approximation of $\boldsymbol{\mu}$ in the neighbourhood of the current value $(\boldsymbol{\beta}^0, f^0)$. We have:

$$\boldsymbol{\mu}(\boldsymbol{\beta}, \mathbf{f}) = \underbrace{g^{-1}(\mathbf{X}\boldsymbol{\beta}^0 + \mathbf{f}_n^0)}_{=\boldsymbol{\mu}^0} + \frac{\partial \boldsymbol{\mu}(\boldsymbol{\beta}, f)}{\partial \boldsymbol{\beta}} (\boldsymbol{\beta} - \boldsymbol{\beta}^0) + \frac{\partial \boldsymbol{\mu}(\boldsymbol{\beta}, f)}{\partial f}.$$

We have the to compute the partial derivatives of $\boldsymbol{\mu}$ with respect to both variables $\boldsymbol{\beta}$ and f. Since we have

$$g(\mu_i^0) = \mathbf{x}_i^t \boldsymbol{\beta}^0 + [\mathbf{f}_n^0]_i \Rightarrow \frac{\partial}{\partial \beta_j} g(\mu_i^0) = g'(\mu_i^0) \frac{\partial \mu_i^0}{\partial \beta_j} = \frac{\partial}{\partial \beta_j} \mathbf{x}_i^t \boldsymbol{\beta}^0 = \mathbf{X}_{ij} \Rightarrow \mathbf{J}_{ij} = \frac{\mathbf{X}_{ij}}{g'(\mu_i^0)},$$

we get then:

$$\frac{\boldsymbol{\mu}(\boldsymbol{\beta}, f)}{\partial \boldsymbol{\beta}} = (\mathbf{G}^0)^{-1} \mathbf{X},$$

where $\mathbf{G}^0 = \mathrm{diag}(g'(\boldsymbol{\mu}^0))$ and $\mathbf{X}$ is the design matrix.

Since $\mu(\boldsymbol{\beta}, f) = g^{-1}(\boldsymbol{\eta}(\beta, f)$, $\boldsymbol{\eta} = (\mathbf{X}\boldsymbol{\beta} + \mathbf{f}_n)$, for the Gâteaux derivative of $\mu(\beta, f)$, in the direction f, we have for ith component:

$$\begin{array}{rl} & \displaystyle\lim_{t \to 0} \frac{\mu_i(\boldsymbol{\beta}^0, f^0 + tf) - \mu_i(\boldsymbol{\beta}^0, f^0)}{t} \\ = & \displaystyle\lim_{t \to 0} \frac{g^{-1}(\eta_i(\boldsymbol{\beta}^0, f^0 + tf)) - g^{-1}(\eta_i(\boldsymbol{\beta}^0, f^0))}{\eta_i(\boldsymbol{\beta}^0, f^0 + tf) - \eta_i(\boldsymbol{\beta}^0, f^0)} \frac{\eta_i(\boldsymbol{\beta}^0, f^0 + tf) - \eta_i(\boldsymbol{\beta}^0, f^0)}{t} \\ = & \displaystyle\frac{1}{g'(\mu_i^0)} [\mathbf{f}_n]_i, \end{array}$$

and thus we finally have a first order approximation of $\mathcal{S}$ in the neighbourhood of the current value $(\boldsymbol{\beta}^0, f^0)$:

$$\mathcal{S} \approx \| \mathbf{V}^{-1/2} (\mathbf{G}^0)^{-1} (\mathbf{G}^0 (\mathbf{y} - \boldsymbol{\mu}_0) - \boldsymbol{\eta}^0 - \mathbf{f}_n) \|^2 + \frac{\lambda}{2} \, m(f, f),$$

where $\boldsymbol{\eta}^0 = \mathbf{X}\boldsymbol{\beta}^0$. Setting $\mathbf{z}^0 = \boldsymbol{\eta}^0 + \mathbf{G}^0(\mathbf{y} - \boldsymbol{\mu}_0)$, and $\mathbf{W}^0$ the diagonal matrix such that $(\mathbf{W}^0)_{ii} = \dfrac{1}{V(\mu_i^0)g'(\mu_i^0)^2}$, we can write $\mathcal{S}$ as:

$$\mathcal{S} \approx \|(\mathbf{W}^0)^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta} - \mathbf{f}_n)\|^2 + \frac{\lambda}{2}\, m(f, f).$$

Since $\mathbf{W}^0$ is positive definite, $\mathcal{S}$ is approximately a quadratic form whose minimum exists and is unique.

The likelihood of an exponential family distribution is strictly concave if a canonical parameter is used. Since the penalization term is concave too (it is minus a positive term), the maximum of the penalized log-likelihood is unique, when it exists. Therefore, if the convergence of the PIRLS is reached, it always results in the maximum penalized log-likelihood estimate.

4.2 Derivation of the penality matrix

If we consider the problem of minimizing the functional (4.1), it leads to an infinite dimensional minimization problem. To avoid this problem, we discretize the penalization operator, in our case the integral of the square of the Laplacian.

We consider any function $h \in \mathcal{F}_K$, where $\mathcal{F}_K$ is K-dimensional finite element space with basis functions $\{\psi_1(x), \dots, \psi_K(x)\}$. We consider $\Omega_\mathcal{T}$, a triangulation of the domain Ω. Our penalization of the function is then written as:

$$p(h) = \int_{\Omega_\mathcal{T}} (\Delta h)^2 = \int_{\Omega_\mathcal{T}} \Delta h\ \Delta h.$$

Setting $g = \Delta h$ we can write:

$$p(h) = \int_{\Omega_\mathcal{T}} \Delta h\ g, \text{ such that } g = \Delta h.$$

Using the characterization of (3.1), we can write:

$$p(h) = \int_{\Omega_\mathcal{T}} \Delta h\ g, \text{ such that } \int_{\Omega_\mathcal{T}} g\ v = \int_{\Omega_\mathcal{T}} \Delta h\ v \quad \forall v \in \mathcal{F}_K.$$

Using the Green formula and thanks to the homogeneous Neumann boundary conditions, we obtain:

$$p(h) = -\int_{\Omega_\mathcal{T}} \nabla h \cdot \nabla g \text{ such that } \int_{\Omega_\mathcal{T}} g\ v = -\int_{\Omega_\mathcal{T}} \nabla f \cdot \nabla v \quad \forall v \in \mathcal{F}_K.$$

Since $\int_{\Omega_\mathcal{T}} g\ v = -\int_{\Omega_\mathcal{T}} \nabla h \cdot \nabla v$ must be satisfied for all $v \in \mathcal{F}_K$, it is sufficient to satisfies this condition for any element of the basis. We define the $K \times K$ matrices $\mathbf{R}_0$ and $\mathbf{R}_1$ as in section (3.3). Writing h and g as $\mathbf{h}^t\boldsymbol{\psi}$ and $\mathbf{g}^t\boldsymbol{\psi}$, we get:

$$p(h) = -\mathbf{h}^t\mathbf{R}_1\ \mathbf{g} \text{ such that } \mathbf{R}_1\ \mathbf{h} + \mathbf{R}_0\ \mathbf{g} = 0.$$

Finally, we can write the penalization of the function h as:

$$p(h) = \mathbf{h}^t\ \mathbf{R}_1^t\ \mathbf{R}_0^{-1}\mathbf{R}_1\ \mathbf{h}.$$

We can conclude that the penalization matrix can be written as:

$$\tilde{S} = \mathbf{R}_1^t\ \mathbf{R}_0^{-1}\mathbf{R}_1 \tag{4.3}$$

4.3 Finite dimensional derivation of the PIRLS algorithm

We are looking for the minimum of the penalized log-likelihood function. Let be $\boldsymbol{\beta}^* = (\boldsymbol{\beta}, \mathbf{f})^t$, where $\boldsymbol{\beta} \in \mathbb{R}^q$ and $\mathbf{f}^t$ is the vector containing the values of coefficients defining the function $\hat{f} \in \mathcal{F}_K$. For sake of simplicity let $N = q + K$ be the dimension of the vector $\boldsymbol{\beta}^*$. We can write the finite dimensional approximation of the penalized log-likelihood as:

$$l_p(\boldsymbol{\beta}^*) = l(\boldsymbol{\beta}^*) - \lambda \, {\boldsymbol{\beta}^*}^t S \, \boldsymbol{\beta}^*,$$

where S is a $N \times N$ penalization matrix defined as:

$$\mathbf{S} = \left[\begin{array}{c|c} \mathbf{O}_{q\times q} & \mathbf{O}_{q\times K} \\ \hline \mathbf{O}_{K\times q} & \tilde{\mathbf{S}} \end{array} \right].$$

Since we consider realizations of an exponential family distribution, we can write the density of y_i, with natural parameter θ_i, which depends of $\boldsymbol{\beta}^*$, and scale parameter ϕ, as:

$$f_{\theta_i}(y_i) = \exp\left\{(y_i\theta_i - b(\theta_i))/a_i(\phi) + c(\phi, y_i)\right\}.$$

The functions $b(\cdot)$ and $c(\cdot)$, are arbitrary defined and we set that a has the form $a_i(\phi) = \frac{r_i}{\phi}$ and, in this case $r_i = 1$. This choice is explained in the section 2.2. The log-likelihood of the model is then:

$$l(\boldsymbol{\beta}^*) = \sum_{i=1}^{n} r_i \left[(y_i\theta_i - b(\theta_i))/\phi + c(\phi, y_i)\right].$$

Hence, the derivative of $l(\boldsymbol{\beta}^*)$ with respect to $\boldsymbol{\beta}_j^*$ is:

$$\frac{\partial l(\boldsymbol{\beta}^*)}{\partial \beta_j^*} = \frac{1}{\phi}\sum_{i=1}^{n} r_i \left(y_i \frac{\partial \theta_i}{\partial \beta_j^*} - b'(\theta_i)\frac{\partial \theta_i}{\partial \beta_j^*} \right).$$

Using the chain rule, we have:

$$\frac{\partial \theta_i}{\partial \beta_j^*} = \frac{\partial \theta_i}{\partial \mu_i}\frac{\partial \mu_i}{\partial \beta_j^*}.$$

Since we have $\mu_i = b'(\theta_i)$, for any distribution belonging to the exponential family, we have:

$$\frac{\partial \mu_i}{\partial \theta_i} = b''(\theta_i) \Rightarrow \frac{\partial \theta_i}{\partial \mu_i} = \frac{1}{b''(\theta_i)}.$$

We have then:

$$\frac{\partial l(\boldsymbol{\beta}^*)}{\partial \beta_j^*} = \frac{1}{\phi}\sum_{i=1}^{n} \frac{(y_i - b'(\theta_i))}{b''(\theta_i)/r_i}\frac{\partial \mu_i}{\partial \beta_j}.$$

Since we have $\text{var}(y_i) = b''(\theta_i)\phi = V(\mu_i)\phi$, we get:

$$\frac{\partial l(\boldsymbol{\beta}^*)}{\partial \beta_j^*} = \frac{1}{\phi}\sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)}\frac{\partial \mu_i}{\partial \beta_j}.$$

Hence the estimate have to satisfy the following equations:

$$\frac{\partial l_p(\boldsymbol{\beta}^*)}{\partial \beta_j^*} = 0 \quad \forall j \Leftrightarrow \frac{1}{\phi}\sum_{i=1}^{n} \frac{(y_i - \mu_i)}{V(\mu_i)}\frac{\partial \mu_i}{\partial \beta_j} - \lambda \, [\mathbf{S}\boldsymbol{\beta}]_i = 0, \quad \forall j = 1, \ldots, N. \tag{4.4}$$

where $[\mathbf{S}\boldsymbol{\beta}]_i$ is the ith component of the vector $\mathbf{S}\boldsymbol{\beta}$. Let $\mathcal{S}$ be defined as:

$$\mathcal{S} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{V(\mu_i)} + \frac{\lambda}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^*$$

If we consider $V(\mu_i)$ as fixed, the solution of (4.4) is equivalent to minimizing the functional $\mathcal{S}$ with respect to $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_n)^t$. This suggests then the following iterative procedure. Given a starting value $\boldsymbol{\mu}^0 = g^{-1}(\boldsymbol{\beta}^0)$, we compute the value of $V(\boldsymbol{\mu}^0)$. We then have to minimize the value of $\mathcal{S}$:

$$\mathcal{S} = \sum_{i=1}^{n} \frac{(y_i - \mu_i)^2}{V(\mu_i)} + \lambda\, \frac{1}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^* = \|\mathbf{V}^{-1/2}\,(\mathbf{y} - \boldsymbol{\mu}(\boldsymbol{\beta}^*))\,\|^2 + \lambda\, \frac{1}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^*,$$

where $\mathbf{V} = \mathrm{diag}(V(\mu_1^0), \ldots, V(\mu_n^0))$. Since $\boldsymbol{\beta}^0$ is supposed to be in the vicinity of the true value $\boldsymbol{\beta}^*$, using a first order Taylor approximation, we can write:

$$\boldsymbol{\mu}(\boldsymbol{\beta}^0) \approx \boldsymbol{\mu}_0 + \mathbf{D}\mu\,(\boldsymbol{\beta}^* - \boldsymbol{\beta}_0)\,,$$

where $(\mathbf{D}\mu)_{ij} = \dfrac{\partial \mu_i}{\partial \beta_j}(\boldsymbol{\beta}^0)$ and so we have:

$$\mathcal{S} \approx \|\mathbf{V}^{-1/2}\,(\mathbf{y} - \boldsymbol{\mu}_0 - \mathbf{D}\mu\,(\boldsymbol{\beta}^* - \boldsymbol{\beta}_0))\,\|^2 + \lambda\, \frac{1}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^*.$$

Since we have

$$g(\mu_i) = \mathbf{x}_i^t\boldsymbol{\beta}^* \Rightarrow \frac{\partial}{\partial \beta_j} g(\mu_i) = g'(\mu_i)\frac{\partial \mu_i}{\partial \beta_j} = \frac{\partial}{\partial \beta_j}\mathbf{x}_i^t\boldsymbol{\beta}^* = \mathbf{X}_{ij} \Rightarrow (\mathbf{D}\mu)_{ij} = \frac{\mathbf{X}_{ij}}{g'(\mu_i^0)}$$

Defining $\mathbf{G}^0 = \mathrm{diag}(g'(\mu_1^0), \ldots, g'(\mu_n^0))$, we note that we have $\mathbf{D}\mu = (\mathbf{G}^0)^{-1}\ \mathbf{X}$. We can then finally write:

$$\mathcal{S} \approx \|\mathbf{V}^{-1/2}(\mathbf{G}^0)^{-1}(\mathbf{G}^0(\mathbf{y} - \boldsymbol{\mu}_0) - \boldsymbol{\eta}^0 - \mathbf{X}\boldsymbol{\beta}^*)\|^2 + \lambda\, \frac{1}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^*,$$

where $\boldsymbol{\eta}^0 = \mathbf{X}\boldsymbol{\beta}^0$. Setting $\mathbf{z}^0 = \boldsymbol{\eta}^0 + \mathbf{G}^0(\mathbf{y} - \boldsymbol{\mu}_0)$, and $\mathbf{W}^0$ the diagonal matrix such that $(\mathbf{W}^0)_{ii} = \dfrac{1}{V(\mu_i)g'(\mu_i)^2}$, we can write $\mathcal{S}$ as:

$$\mathcal{S} \approx \|(\mathbf{W}^0)^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}^*)\|^2 + \lambda\, \frac{1}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^*. \tag{4.5}$$

Since $\mathbf{W}^0$ is positive definite, $\mathcal{S}$ is approximately a quadratic form whose minimum exists and is unique.

The algorithm aims to minimize this approximation to get the value $\boldsymbol{\beta}^1$ of the parameter and then iterate the procedure until convergence. The solution of minimization problem (4.5) is the solution of a linear system which, under mild conditions, admits a unique solution (see proposition 2).

Actually, the value of

$$-\frac{1}{2\phi}\|(\mathbf{W}^0)^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}^*)\|^2 + \lambda\, \frac{1}{2}\, \boldsymbol{\beta}^{*t}\ \mathbf{S}\ \boldsymbol{\beta}^*,$$

evaluated after convergence of the algorithm can be viewed as a quadratic approximation of the penalized log-likelihood function. For a detailed development of this argument, see Wood (2011, p.66).

An alternative of the log-likelihood as measure of goodness-of-fit is given by the *deviance*. The deviance is the difference between the log-likelihood of the estimated model and of the full model, i.e., the model with as many variables as observations. Usually, maximizing the log-likelihood or minimizing the deviance leads to the same estimates. Actually, we can notice that the term $\|(\mathbf{W}^0)^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta}^*)\|^2$ is an approximation of the deviance of the model.

Chapter 5

Smoothing parameter choice and variance estimation

5.1 Choosing the smoothing parameter: a bias/variance trade-off

The choice of the smoothing parameter is crucial for the estimation. This choice express the classical bias/variance trade-off. Let $\mathbf{S}_\lambda$ be a smoothing matrix, i.e., a matrix such that for a given observation $\mathbf{y}$, we have that $\hat{\mathbf{y}}$ is given by $\mathbf{S}_\lambda \mathbf{y} = \hat{\mathbf{y}}$. In our case, since we consider generalized additive models, the smoothing matrix is the matrix such that applied to the last pseudo-data, it gives the estimation of the natural parameter η (Hastie and Tibshirani, 1990, p.156).

To illustrate the bias/variance trade-off, we will take the simplest case. Let's consider $y_1, \ldots, y_n \in \mathbb{R}$ observations at points $x_1, \ldots, x_n \in \mathbb{R}$ and assume that we consider them as realizations of a sufficiently smooth function g with Gaussian noise:

$$y_i = g(x_i) + \varepsilon_i, \quad \varepsilon \sim \mathcal{N}(0, \sigma^2).$$

We would like to ensure a sufficiently smoothness of the estimated function $\hat{g}$. To do this, we will consider the following problem:

$$\text{find } g \in C^2 \text{ that minimize } \sum_{i=1}^{n} (y_i - g(x_i))^2 + \lambda \int (g''(x))^2 dx$$

The term $\lambda \int (g''(x))^2 dx$ represent a roughness penalty. We aim to estimate the function g using a finite dimensional space with basis $(\psi_1(x), \ldots, \psi_K(x))$. In this case, defining $\mathbf{P}$ and $\mathbf{X}$ as

$$(\mathbf{P})_{ij} = \int \psi_i''(x) \psi_j''(x) dx \text{ and } (\mathbf{X})_{ij} = \psi_j(x_i),$$

the estimation problem becomes:

$$\text{find } \hat{\boldsymbol{\beta}} \in \mathbb{R}^K \text{ such that } \hat{\boldsymbol{\beta}} = \operatorname{argmin} \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + \lambda \, \boldsymbol{\beta}^t \mathbf{P} \boldsymbol{\beta}$$

and $\hat{g}$ is given by:

$$\hat{g}(x) = \sum_{i=1}^{K} \beta_i \psi_i(x).$$

Setting $\mathbf{g} = (g(x_1), \ldots, g(x_n))$ and $\hat{\mathbf{g}} = (\hat{g}(x_1), \ldots, \hat{g}(x_n))$, we can decompose the mean square error in the bias and in the variance components as:

$$\mathbb{E}[\|\mathbf{g} - \hat{\mathbf{g}}\|^2] = \text{var}(\hat{\mathbf{g}}) + (\mathbb{E}[\hat{\mathbf{g}}] - \mathbf{g})^2.$$

In our case, we can write this as:

$$\mathbb{E}[\|\mathbf{g} - \hat{\mathbf{g}}\|^2] = \text{tr}(\mathbf{S}_\lambda \mathbf{S}_\lambda^t)\sigma^2 + (\mathbf{g} - \mathbf{S}_\lambda \mathbf{g})^t(\mathbf{g} - \mathbf{S}_\lambda \mathbf{g}).$$

Finally, we can notice that if λ increase, the variance decrease but the bias increase and at the contrary, if λ decrease, the variance increase but the bias decrease. The optimal λ depends of the unknown function g.

In the more general context of generalized additive models and if the function to be estimated is a function of several variables, the same kind of reasoning applies.

5.2 Choosing the smooth parameter: the generalized cross validation

To choose the best parameter, we use a prediction error criterion. The basic idea of the prediction error is to fit the model using all the observations except one and to compute the error between the observed value which was not used to fit the model and the predicted value of the model for this observation. Doing this for all the observations, and summing these errors we obtain, for a given model, a total prediction error criterion. Since, in this case, the model vary only with the value of the smoothing parameter λ, we the choose the value of λ that minimizes this total prediction error criterion. This kind of procedure is called *cross validation.* We will use a slightly different version of the cross validation called *generalized cross validation* (GCV) (Wahba, 1990). For more details about cross validation and generalized cross validation we refer to Wood (2011, p. 172-187). But the GCV is not optimal in the kind of context we consider since independence is required. We could use other kind of criterion as AIC or BIC but no available criterion is optimal.

One of the main ingredient of the GCV procedure is defined as the *equivalent degree of freedom.* Since we fit a function which is defined by K degree of freedom, one expects that it should increase the degree of freedom of the entire model. We know that in the case of a linear model, if $\mathbf{R}$ is the hat matrix (or influence matrix, see Wood (2011, p. 16-17)), one can notice that number of identifiable parameter is given by $\text{tr}(\mathbf{R})$. Analogously, we define the *equivalent degree of freedom* for generalized additive model as the trace of the hat matrix (at convergence of the PIRLS) $\mathbf{R}$, as defined in the section 3.4. This measure of the degree of freedom is usual in the context of functional data analysis but other measure are possible (Ramsay and Silverman, 2005).

An alternative of the log-likelihood as measure of goodness of fit is given by the *deviance.* Actually the deviance is the difference between the log-likelihood of the estimated model and of the full model, i.e., the model with as many variables as observations. Usually, maximizing the log-likelihood or minimizing the deviance leads to the same estimates. We have seen at the end of the section 4.3, that the value of $\|(\mathbf{W}^0)^{1/2}(\mathbf{z} - \mathbf{X}\hat{\boldsymbol{\beta}} + \hat{\mathbf{f}}_n)\|^2$ can be seen as a scaled approximation of the deviance, denoted by $D(\boldsymbol{\beta})$ of the model. Finally, for the hat matrix $\mathbf{R}$, the value of the GCV criterion is given by (see Wood (2011, p.181)):

$$\text{GCV}(\lambda) = \frac{nD(\hat{\beta}(\lambda))}{[n - \gamma\ \text{tr}(\mathbf{R}(\lambda))]^2}, \tag{5.1}$$

where γ is a constant usually set to 1. In some case, the GCV optimum leads to overfitting so it is sometimes usefull, as suggested in Wood (2011), to give more weight to the equivalent degree of freedom setting $\gamma \geq 1$. In our simulations, such a tuning constant was necessary to ensure that the convexity of the GCV function. Not using such a constant can leads to misestimation of smoothing parameters and then to misestimation of the other parameters.

A lot of aspects of smoothing parameter choice could be discussed. First, the parameter estimation could be done as a step of the PIRLS algorithm, leading to an update of the value of λ at each iteration of the PIRLS algorithm. Alternatively, the update of a new smoothing parameter could be done at the convergence of the algorithm. These two different approaches as refereed as *performance iteration* and *outer iteration* respectively. These two methods are extensively discussed in Wood (2011, p.179-189). In our case, we simply tested a grid of values and takes the best one, according to the value of the GCV criterion, which is a kind of outer iteration.

Estimations are generally stable with respect to the smoothing parameter, that is a small change in the smoothing parameter won't induce a great change in the estimations of the parameters. Sometimes a tuning constant is necessary to ensure the convexity of the GCV function. In practice, it is useful to take a look at the GCV function to eventually detect such a phenomenon. Not using such a constant can leads to the choice of smooth parameter close to zero. In such cases, the estimated parameters can be totally false. This constant increase the weight of the equivalent degree of freedom, which means that overfitting is more penalized that if wouldn't use such a constant. Adding this constant does not influence a lot the minimum and tends to only increase the convexity of the GCV function.

5.3 Variance and scale parameter estimation

In some cases of the exponential family, the distribution depends on two parameters, the mean and another parameter usually called *scale parameter*. In the Gaussian case, that is in a traditional linear model, we estimate the variance σ^2 as, which is unbiased:

$$\hat{\sigma}^2 = \frac{1}{n - \mathrm{tr}(\mathbf{R})} \|\mathbf{y} - \hat{\mathbf{y}}\|^2,$$

which is unbiased and where $\mathbf{R}$ is the hat matrix, $\hat{\mathbf{y}}$ the fitted values and $\mathbf{y}$ the observed values. Analogously, we can define a scale parameter ϕ, which is the variance in the Gaussian case, that can fully characterize the distribution.

If Y belongs to the exponential family (with canonical link function) with distribution f_Y, we have seen in the section 2.2 that we can implicitly define ϕ as:

$$\mathrm{var}(Y) = V(\mu)\phi.$$

So a natural estimator of ϕ is given by:

$$\hat{\phi} = \frac{\|\mathbf{V}^{-1}(\mathbf{y} - \hat{\boldsymbol{\mu}})\|^2}{n - \mathrm{tr}(\mathbf{R})}, \tag{5.2}$$

where $\mathbf{V} = \mathrm{diag}(V(\hat{\mu}_1), \ldots, V(\hat{\mu}_n))$, $\hat{\boldsymbol{\mu}}$ is the estimated mean at the convergence and $\mathbf{R}$ the hat matrix.

Chapter 6

Simulations

The simulations have been done on a C-shaped domain, which has been used in various publications (Ramsay, 2002; Wood et al., 2008; Sangalli et al., 2013), with various test functions. We have randomly generated data points according to a uniform distribution on the C-shaped domain shown in Figure 6.1. The number of points, denoted by n, is 200 for all the simulations on the C-shaped domain. For each case, we did 100 repetitions. We have tested different types of responses, such as binary or gamma.

We use only in one case an automatic selection of the smoothing parameter, in the case of a gamma responses with covariates. This case is the most complex since the scale parameter must also be estimated. We did not use the automatic selection of the smoothing parameter mainly because the simulations are long. The way we implemented smoothing parameter selection is far from optimal. We choose it among a grid of value and choose the parameter which minimize the GCV criterion.

All of the codes are written in **R** (R Core Team, 2012), excepted the one used for the mesh generation, which is a matlab code (MATLAB, 2013). In the future, we could probably use only **R** code, since the package `geometry` is being developed and should be able to make the needed routines available in **R**.

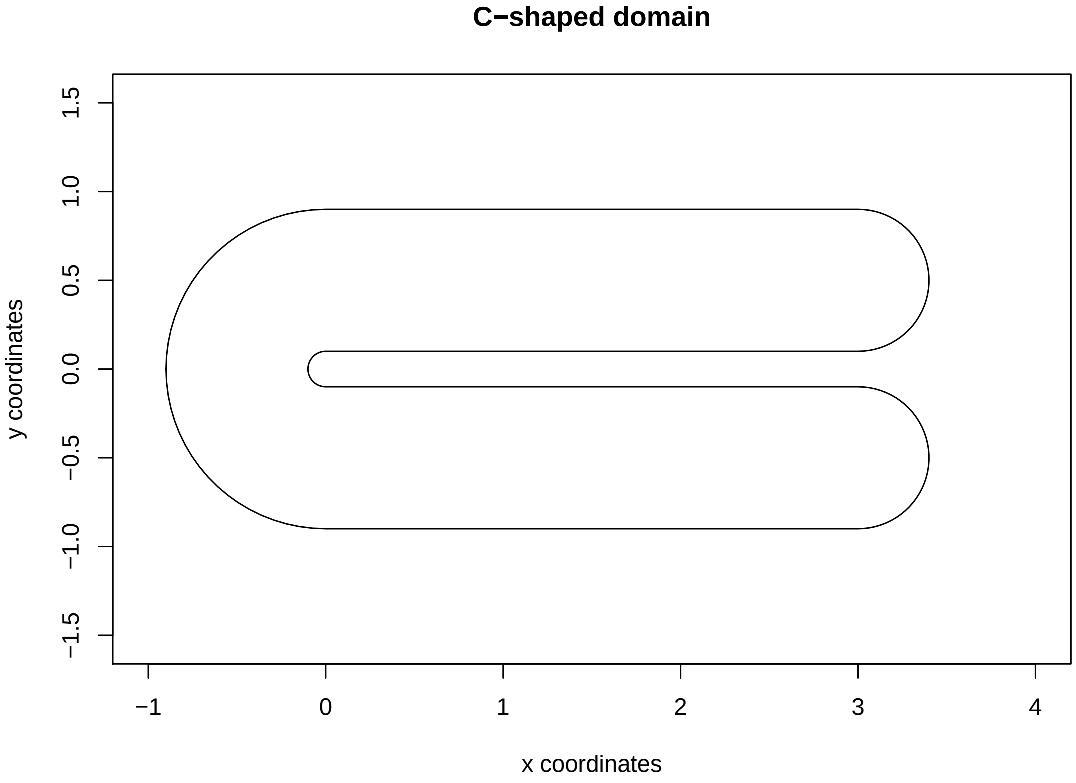


Figure 6.1: Plot of the C-shaped domain used in the simulations.

6.1 Meshes and data

It has been mentioned (section 3.5) that we have used two different types of meshes. The first is referred to as a “data constrained” mesh. It is actually a Delaunay constrained mesh with nodes at the location of the data points and with some other nodes on the boundaries (for more information about meshes and Delaunay triangulation, see Cheng et al. (2013)). In all the simulations, we add the same nodes on the boundaries whatever is the kind of mesh. In the case of data constrained meshes, the vertices of the triangulation are exactly the data points and the 108 points on the boundaries. Since we usually have 200 data points, the dimension of the finite element space is of 308 in this situation. The computational cost of the estimations with data constrained mesh is quite small. Moreover, it makes sense that the precision of the estimation of the function increase with the density of data points.

When we do not use a data constrained, we use a uniform mesh, that is a constrained Delaunay mesh with nodes in a lattice over the domain. In this case, we have more nodes and so we increase the dimension of the finite element space and hence, the size of the system to be solved. Furthermore, we have to construct the matrix $\boldsymbol{\Psi}$, as defined in the section 3.5. We can see in Figure 6.2 the two different types of meshes that we have used.

We have used two different types of responses in our simulations, realizations of a Bernoulli distribution, i.e., binary responses and realizations of a gamma distribution.

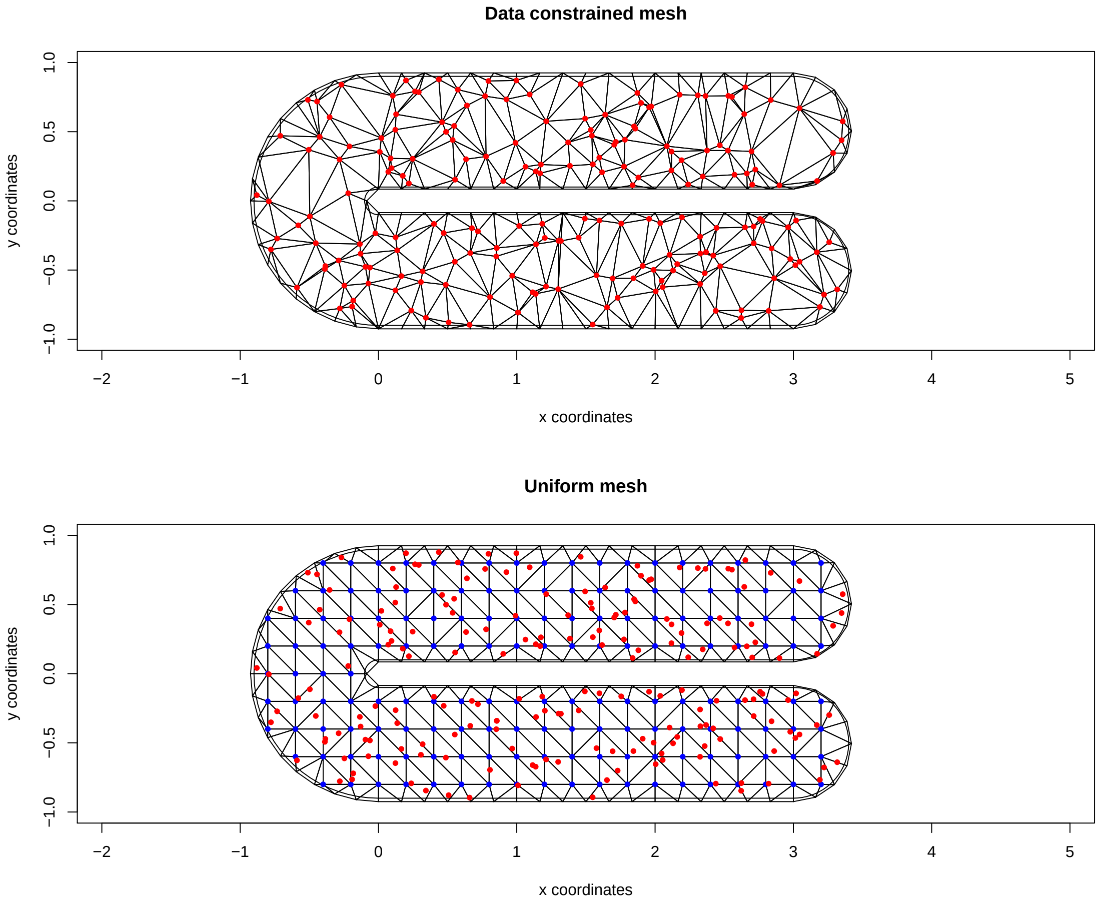


Figure 6.2: Plot of the two different types of meshes. Red bullets are the data points, in both Figures. Blue bullets are the internal nodes of the meshes.

6.2 Simulations without covariates

First, we did simulations not including any covariates. We have then generated data of the type:

$$\mathbb{E}[y_i|\mathbf{p}_i] = g^{-1}(f(\mathbf{p}_i)), \qquad i = 1, \ldots, n$$

where $g(\cdot)$ denotes a link function and $\mathbf{p}_i$ is the location of the data y_i. Here, the function used is the one considered in Sangalli et al. (2013); Wood et al. (2008), which is shown on the Figure 6.3 and is denoted as f_0. We also used another test function denoted f_Γ, defined as $f_\Gamma = f_0 + 100$. We used f_Γ since only positive values were allowed in some cases.

The simulation without covariates allows to compare both estimates of the test function and of the spatial distribution of the mean, which is actually estimated. In the case of covariates, since they are not defined for every point of domain, it is not possible to make this type of comparisons.

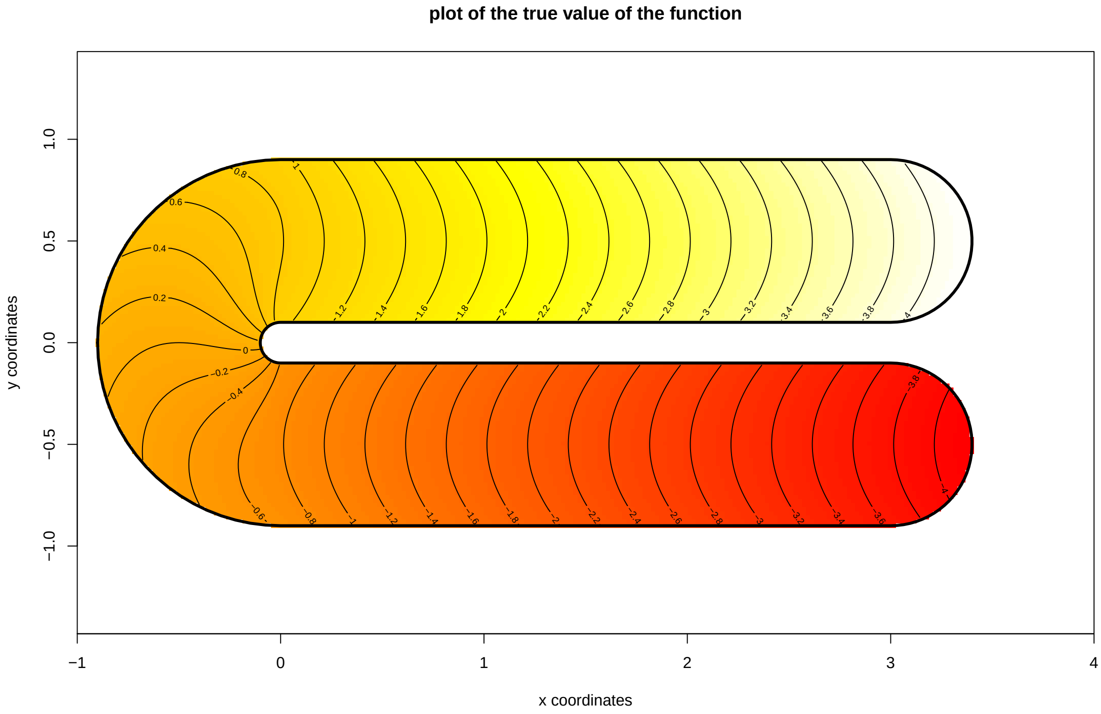


Figure 6.3: Plot of the test function f_0 defined over the C-shaped domain.

6.2.1 Binary responses

To deal with binary responses, we have to choose an appropriate link function. There are several possibilities of link functions in the case of binary responses, such as those referred to as logit, probit or c-loglog. We used the logit function, defined as:

$$g_{\text{logit}}(x) = \log\left(\frac{x}{1-x}\right).$$

It is obvious that links function, in the binary case, have to map the interval $]0,1[$ onto $\mathbb{R}$ and to be bijective. Most of the link functions are symmetric but not all. The choice of the link function won't be discussed here. When a link function is applied to the test function, it returns

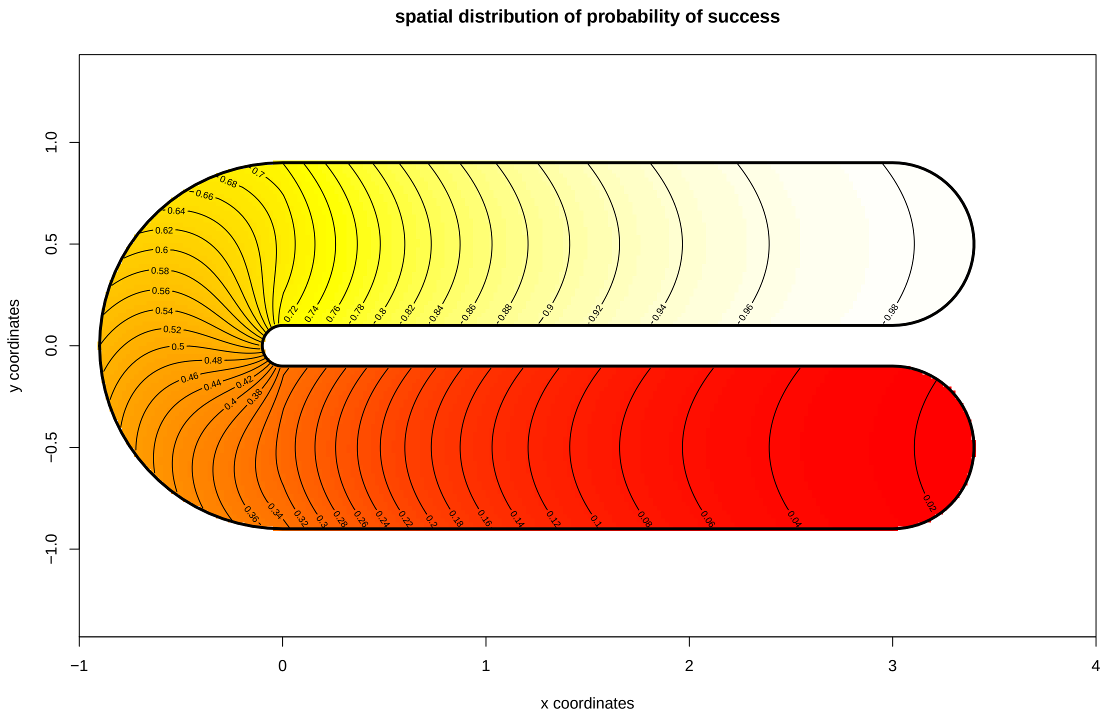


Figure 6.4: Plot of the spatial distribution of the true probability of success over the C-shaped domain.

the probability of success. The spatial distribution of the probability of success corresponding to the test function is shown on the Figure 6.4.

If we denote by p_i the probability of success of the observation associated to the point $\mathbf{p}_i$, we generated the response y_i according the following model:

$$y_i \sim \text{Bernoulli}(p_i), \qquad p_i = g_{\text{logit}}^{-1}(f(\mathbf{p}_i)), \quad i = 1, \dots, 200$$

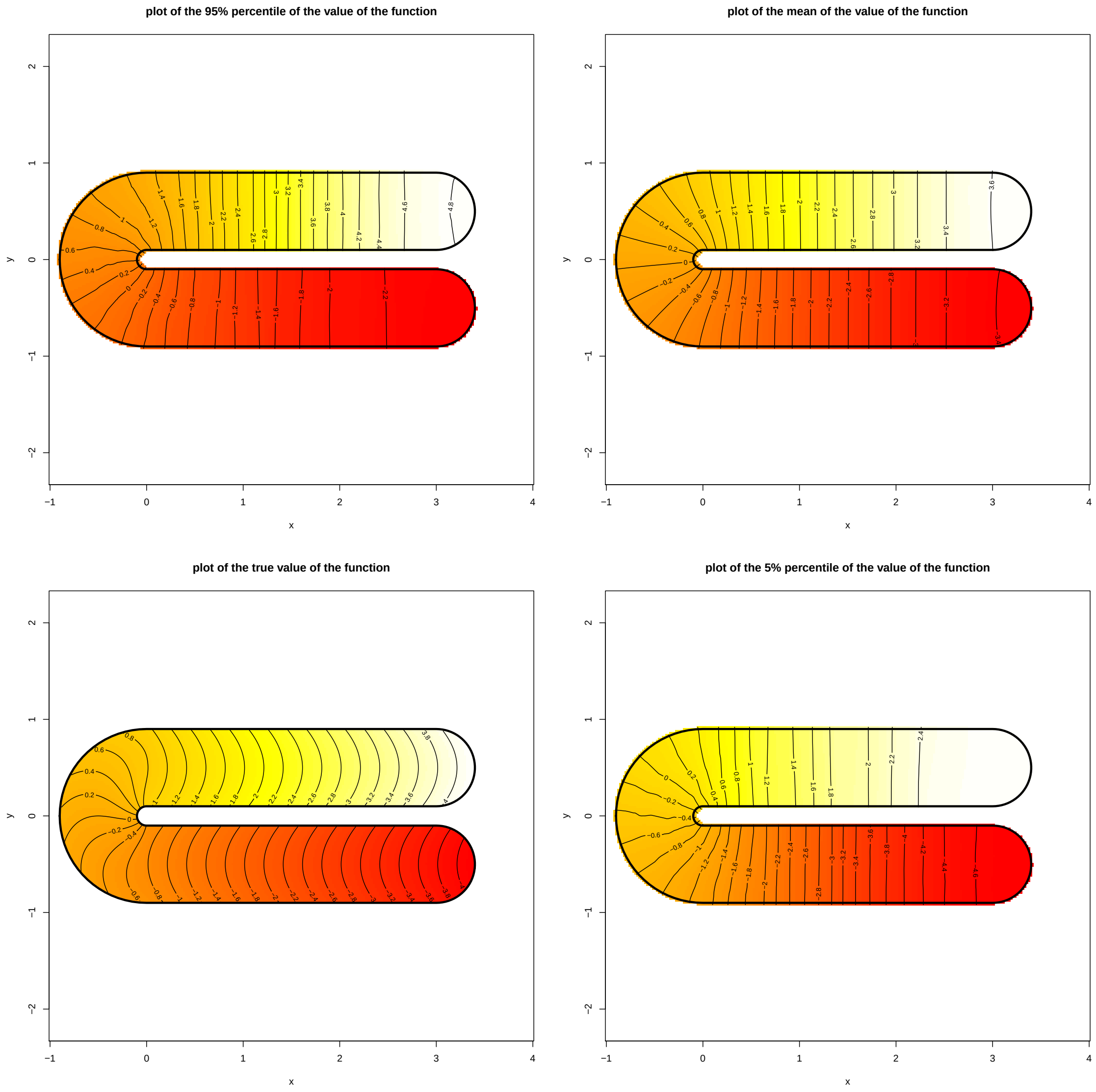
where f_0 is the test function showed in Figure 6.3, and $\mathbf{p}_i$ is the location of the observation y_i.

The summary of these simulations is presented in Figures 6.5 and Figure 6.6. First we can see that the estimator has a reasonable variance, as the spatial distribution of the standard error and the percentiles shows, in Figures 6.7 and 6.8. The choice of the smoothing parameter has been made before the simulation and is the same for all the simulations, that is $\lambda = 1.2$. This choice is not optimal for all the simulations but is reasonable.

It is interesting to notice the effect of the link function on the spatial distribution of the standard errors. Actually, the pattern of the spatial distribution of the sample standard error on the estimated function is not the same as on the estimated probability, as we can see in Figures 6.7 and 6.8. In both case, the standard error is reasonable but the spatial distributions are not similar, mainly because of the behaviour of the link function.

We can see in Figure 6.9 that the estimator seems to be unbiased almost everywhere on the domain. There are two small zones where it fails to be unbiased. It is probably due to the fact that the spatial variation of the true probability in these zones is too strong to be captured by our penalized smooth function. We can actually see in Figure 6.4 how important is the spatial variation of the true probability in these zones. We can also see in Figure 6.6 that the mean estimation have a symmetric behaviour, as we could expect.

Figure 6.9 shows that the difference between unconstrained and constrained meshes seems not to be significant.

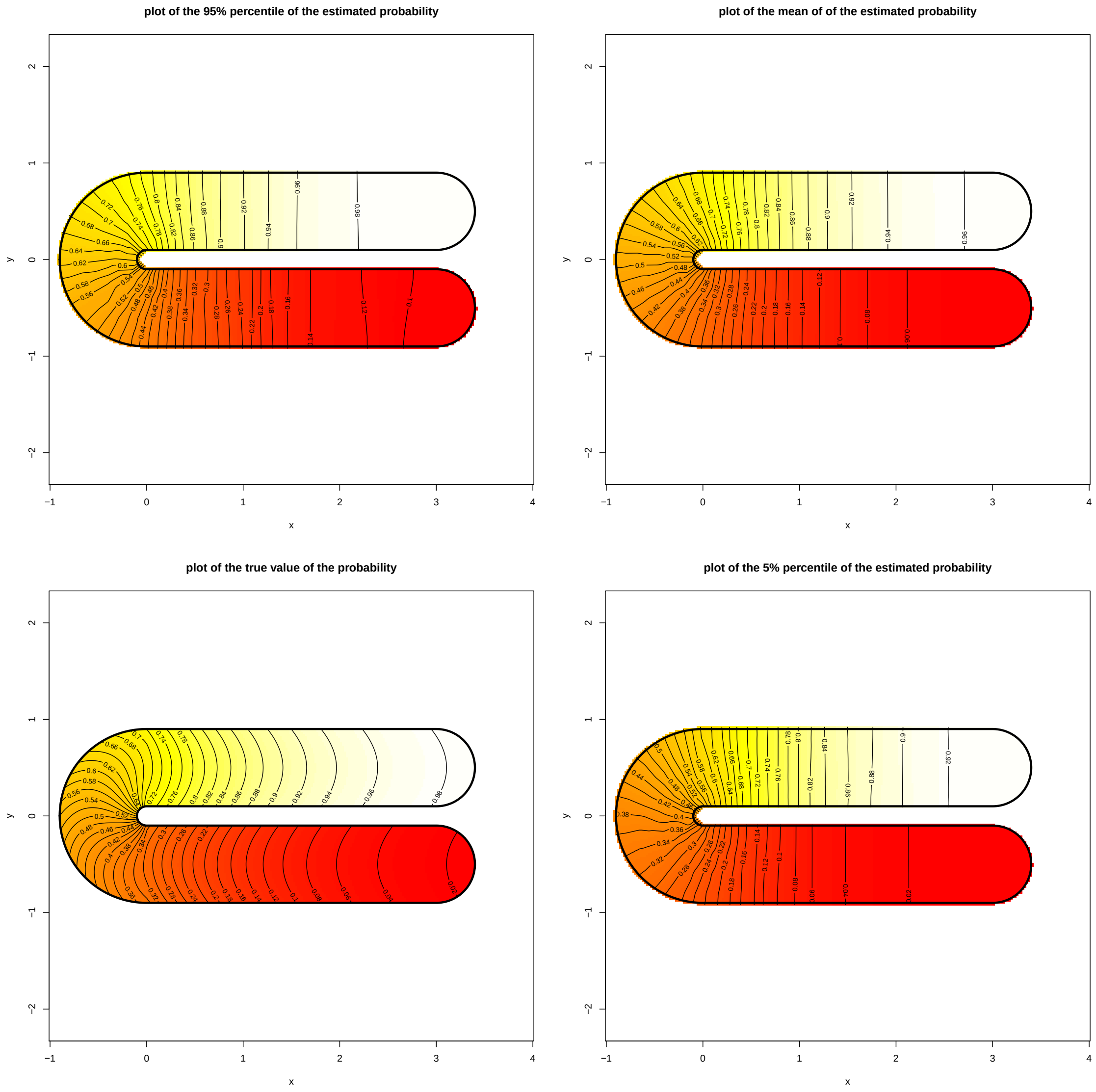

plot of the 95% percentile of the value of the function

plot of the mean of the value of the function

plot of the true value of the function

plot of the 5% percentile of the value of the function

Figure 6.5: Binary responses, without covariates. Mean, 95% and 5% of percentile of the estimated f. The smoothing parameter λ was set to 1.2.


plot of the 95% percentile of the estimated probability

plot of the mean of of the estimated probability

plot of the true value of the probability

plot of the 5% percentile of the estimated probability

Figure 6.6: Binary responses, without covariates. Mean, 95% and 5% of percentile of estimated probability. The smoothing parameter λ was set to 1.2.

We can conclude that our model have good performance in the logistic case without covariates.

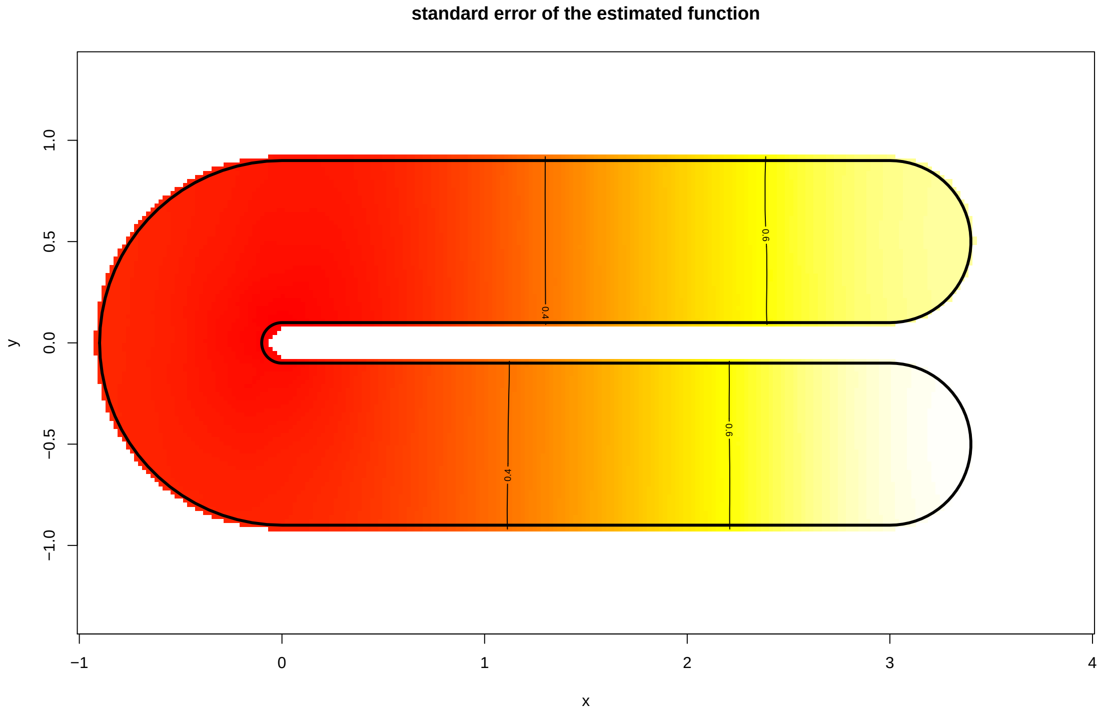


Figure 6.7: Binary responses, without covariates. Empirical standard error of the estimated f. The smoothing parameter λ was set to 1.2.

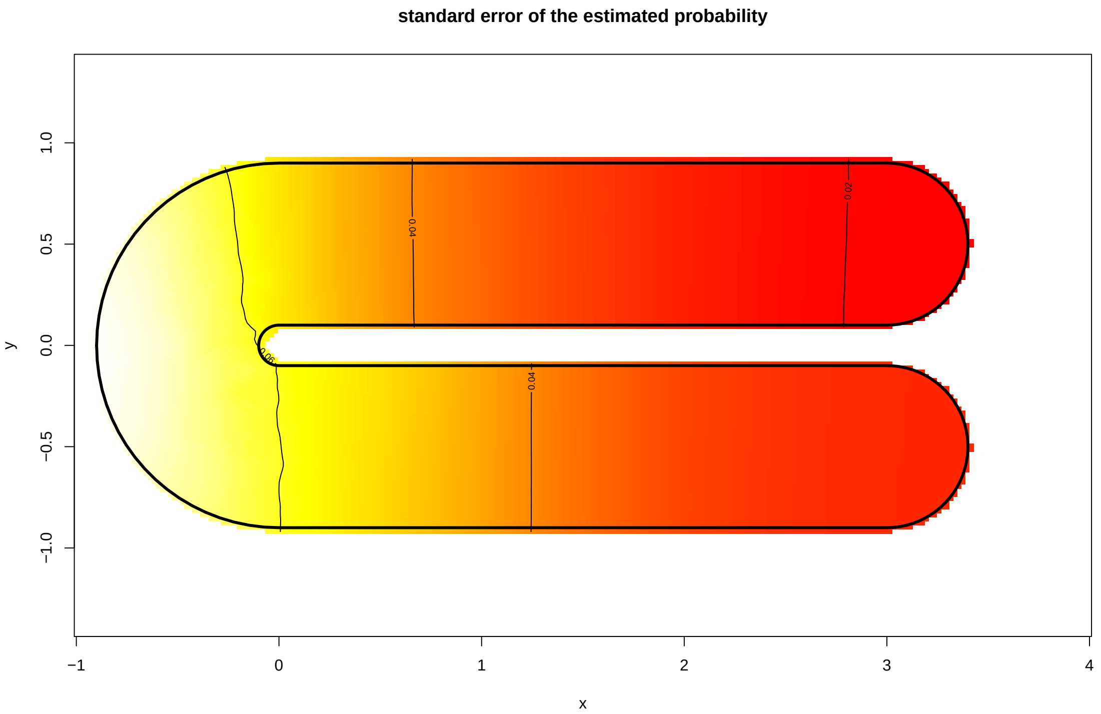
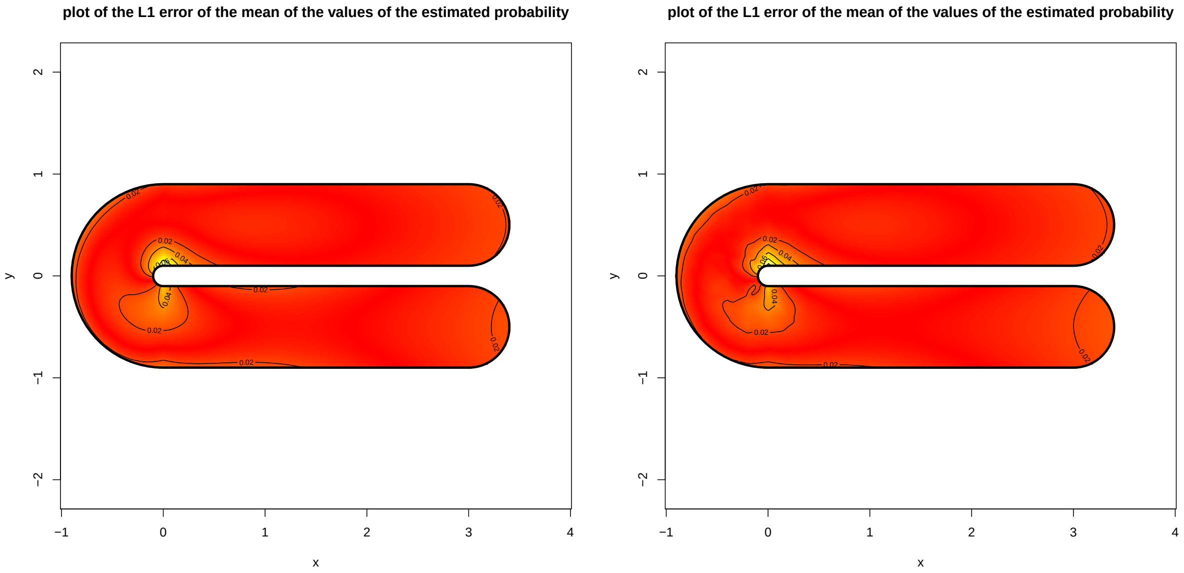


Figure 6.8: Binary responses, without covariates. Empirical standard error of the estimated probability. The smoothing parameter λ was set to 1.2.



(a) With data constrained mesh

(b) With a uniform mesh

Figure 6.9: L_1 error between true probability and mean estimated probability with two type of meshes. The smoothing parameter λ was set to 1.2.

6.2.2 Gamma responses

We present here the results with realizations of a gamma distribution. A random variable Y following a gamma distribution with shape parameter k and scale parameter θ, has distribution given by:

$$f_Y(x; k, \theta) = x^{k-1} \frac{e^{-x/\theta}}{\Gamma(k)\theta^k}.$$

We then have:

$$\mathbb{E}[Y] = k\theta \quad \text{and} \quad \text{var}(Y) = k\theta^2.$$

We choose as test function the already mentioned f_Γ. We then set that θ is constant over the domain and k vary over the domain as:

$$k = k(x) = f_\Gamma(x), \Rightarrow \mathbb{E}[Y](x) = k(x)\theta = f_\Gamma(x)\theta.$$

Since the variance depends on the shape parameter too, the realizations of our simulations are not homoscedastics that is we have a space varying variance.

We start by show how the mean is evaluated, with and without estimation of the scale parameter in Figure 6.10. We can see on the Figure 6.10 that the mean seems to be well estimated in both case. As in the case with binary response, the estimated mean is not able to fully capture the variation of the mean in the radial direction, due to the uniform penalization.

Adding the estimation of the scale parameter does not influence the estimation, as we we can see on the Figure 6.10 and the standard error of the estimator seems to be reasonable.

The performance of our model is quite good, even when an estimation of the scale parameter is needed.

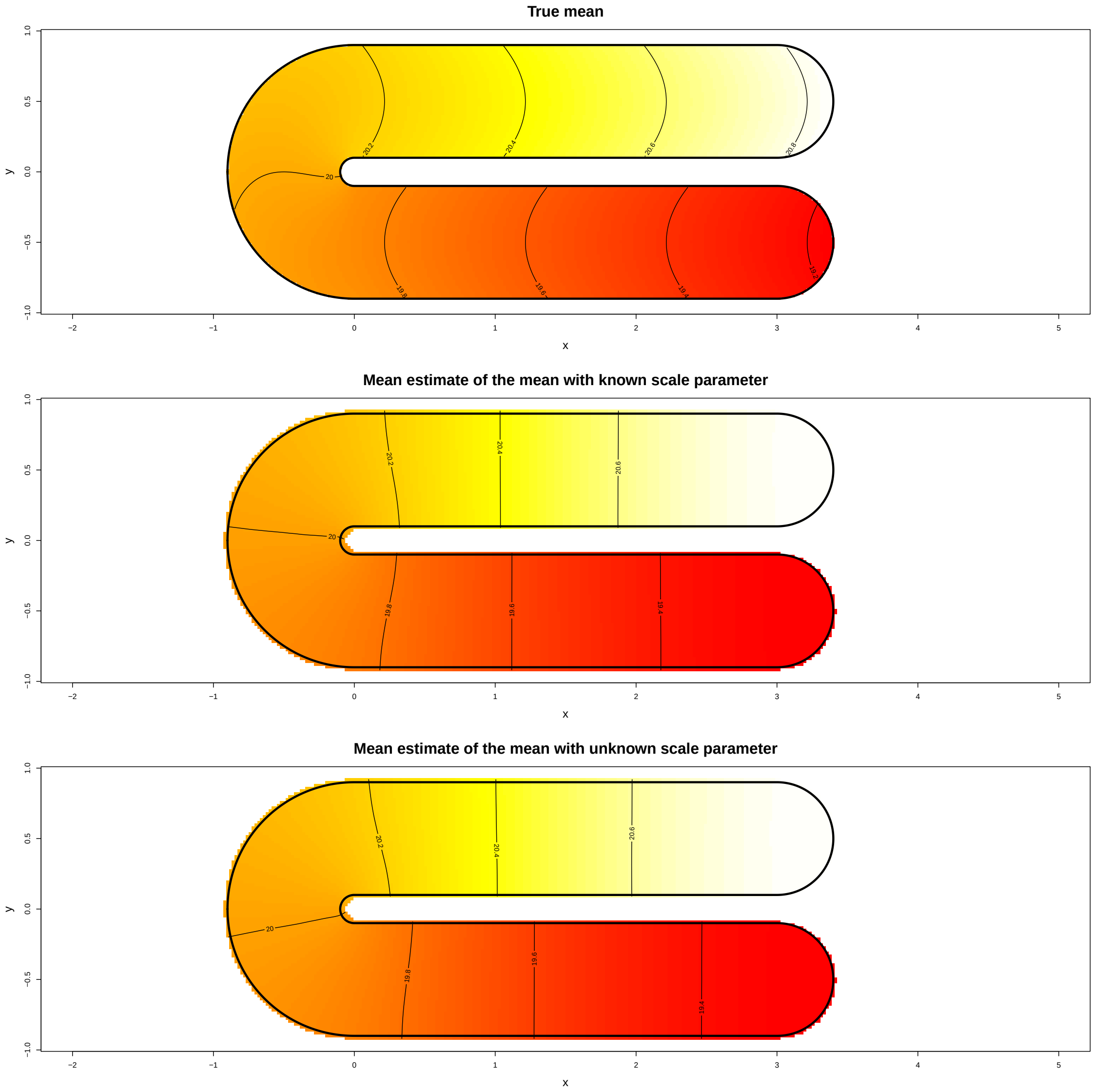


Figure 6.10: Gamma responses, without covariates. Comparison between mean estimates with and without scale parameter estimation. Smoothing parameter λ was set to 9000.

6.3 Simulations with covariates

As our model allows the use of covariates, we made simulations in this case too. We used two kind of responses, binary and gamma.

6.3.1 Binary responses with covariates

First, for $i = 1, \ldots, n$ we generated realizations of two random variables $x_{1i} \sim \mathcal{N}(3, 1)$ and $x_{2i} \sim \mathcal{N}(7, 2)$, defining $\mathbf{x}_i = (x_{1i}, x_{2i})^t$ which are the covariates. $\mathbf{X}$ is the $n \times 2$ matrix whose ith line is $\mathbf{x}_i$. We set $\beta_1 = -\frac{1}{2}$ and $\beta_2 = \frac{1}{5}$, defining $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = (\beta_1, \beta_2)^t$. For the spatial dependency, we use the same test function f_0 as defined in section 6.2.1. So finally, for each data point $\mathbf{p}_i$, we generate an observation according to the following model:

$$y_i \sim \text{Bernoulli}(p_i), \qquad p_i = g_{\text{logit}}^{-1}(\mathbf{x}_i^t \boldsymbol{\beta} + f(\mathbf{p}_i)).$$

As already mentioned, we cannot compare estimation of probabilities but only the underlying function since covariates are not defined for every point of the domain. First, we can see in Figure 6.11 that the accuracy of the estimation seems not to be affected by including covariates. The standard error of the estimation seems to be reasonable too, as it can be seen in Figure 6.12. Since it is not possible to compare the estimated probability and the true probability except on data points, we have separated the domain in regions and we compare the estimated and the true probability for all the simulations but region by region. We can see in Figure 6.13 that the errors in terms of probability seems to have a similar pattern as in the case without covariates.

The estimation of the coefficients seems to be unbiased or at least to have small bias as we can see in Figure 6.14. Finally, we can conclude that our model shows good performance in the logistic case, with covariates.

6.3.2 Gamma responses with covariates

In order to assess our model in the case of a Gamma response, we used a similar setting as in the binary case. Since the results are similar to those of binary case, we do not show all the plots but the essentials ones only. In these simulations, we use an automatic selection of the smoothing parameter λ.

First, for $i = 1, \ldots, n$ we generated realizations of two random variables $x_{1i} \overset{\text{iid}}{\sim} \Gamma(9, 1/3)$ and $x_{2i} \overset{\text{iid}}{\sim} \Gamma(49/2, 2/7)$ [1] defining $\mathbf{x}_i = (x_{1i}, x_{2i})^t$. $\mathbf{X}$ is the $n \times 2$ matrix whose ith line is $\mathbf{x}_i$. We set $\beta_1 = -\frac{1}{2}$ and $\beta_2 = \frac{1}{5}$, defining $\boldsymbol{\beta}$ as $\boldsymbol{\beta} = (\beta_1, \beta_2)^t$. For the spatial dependency, we use the a test function denoted by f_{Γ_1} and defined as

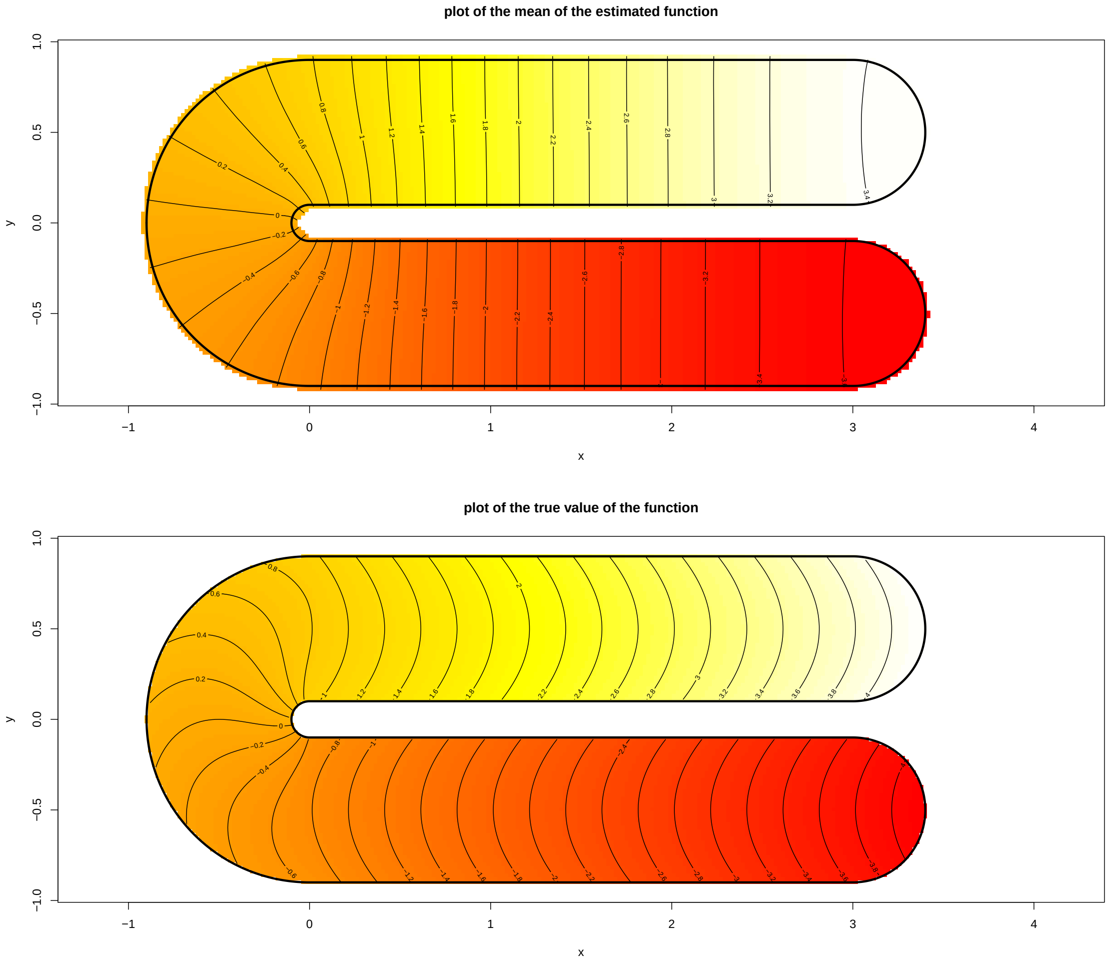
$$f_{\Gamma_1} = -\frac{1}{10}(f_0 + 10).$$

For each data point $\mathbf{p}_i$, we generate an observation according to the following model:

$$y_i \sim \Gamma(k = \mu_i^2/\nu^2, \theta = \nu/\mu_i), \qquad \mu_i = g_{\text{recip}}^{-1}(\mathbf{x}_i^t \boldsymbol{\beta} + f_{\Gamma_1}(\mathbf{p}_i)),\ \nu = 0.8,$$

where g_{recip} denotes the canonical link function associated to the gamma distribution, i.e., the reciprocal ($g_{\text{recip}}(x) = -1/x = g_{\text{recip}}^{-1}(x)$). We used a gamma distribution to generate the covariates to ensure the positivity of them. There is a non-null probability that $\mu_i < 0$ then

[1] for a distribution $\Gamma(k, \theta)$, k is the scale parameter and θ the shape parameter


plot of the mean of the estimated function

plot of the true value of the function

Figure 6.11: Binary responses, with covariates. Mean of the estimated f. The smoothing parameter λ was set to 1.2

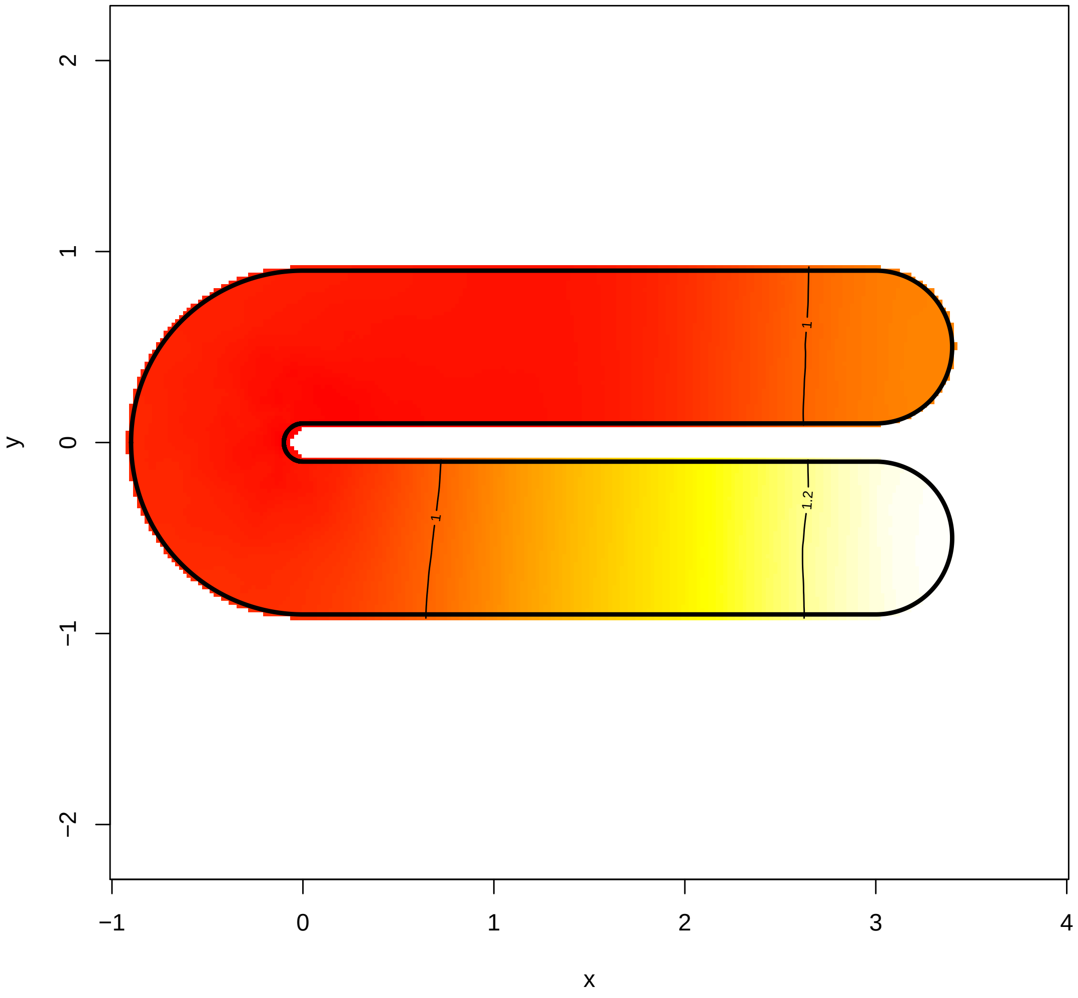


Figure 6.12: Binary responses, without covariates. Standard error of the estimated f. The smoothing parameter λ was set to 1.2

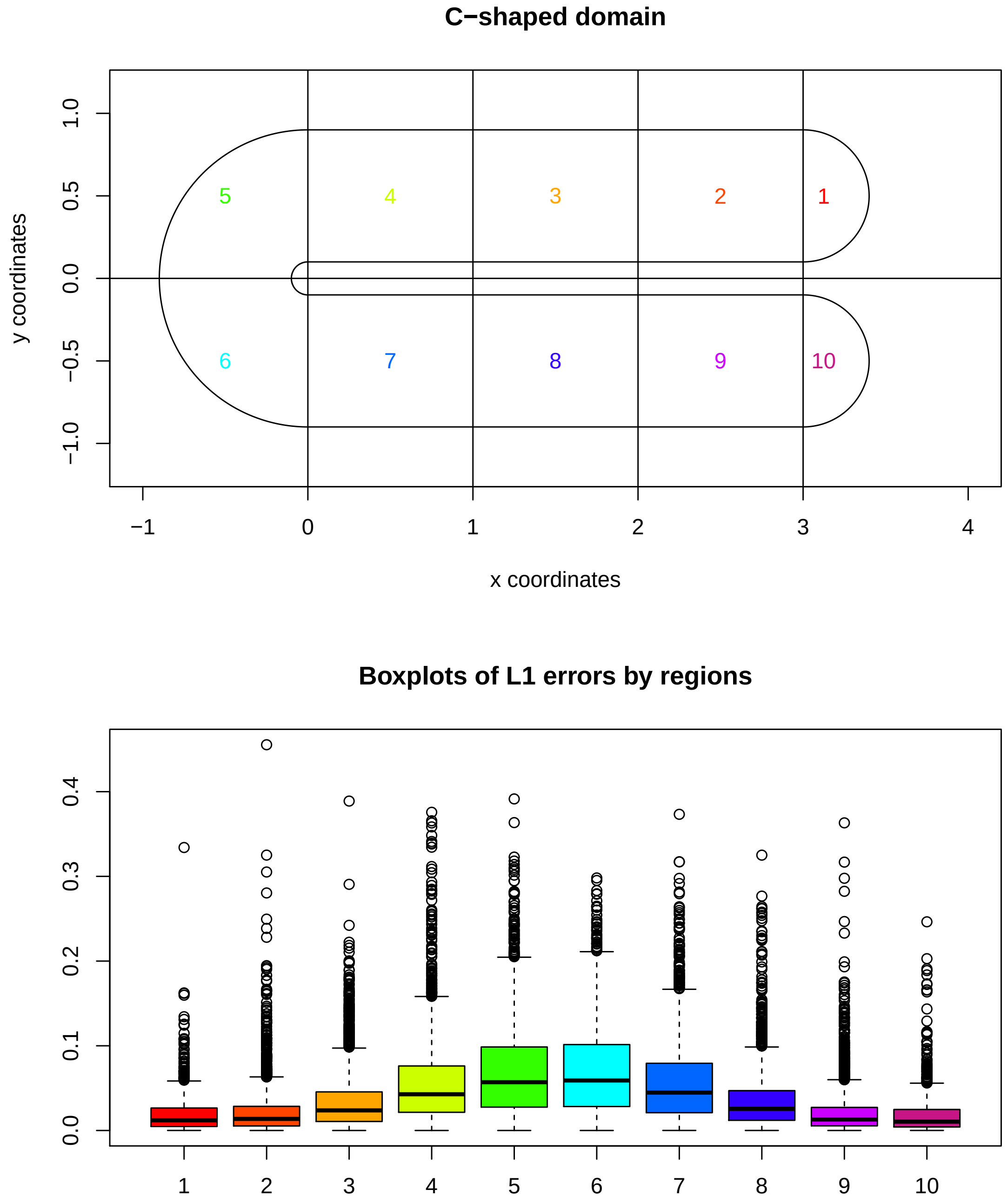


Figure 6.13: Binary responses, with covariates. Top: subdivision of the C-shaped domain. Bottom: for the 20000 points of all the simulations, L_1 error between true and estimated probability per region. The smoothing parameter λ was set to 1.2.

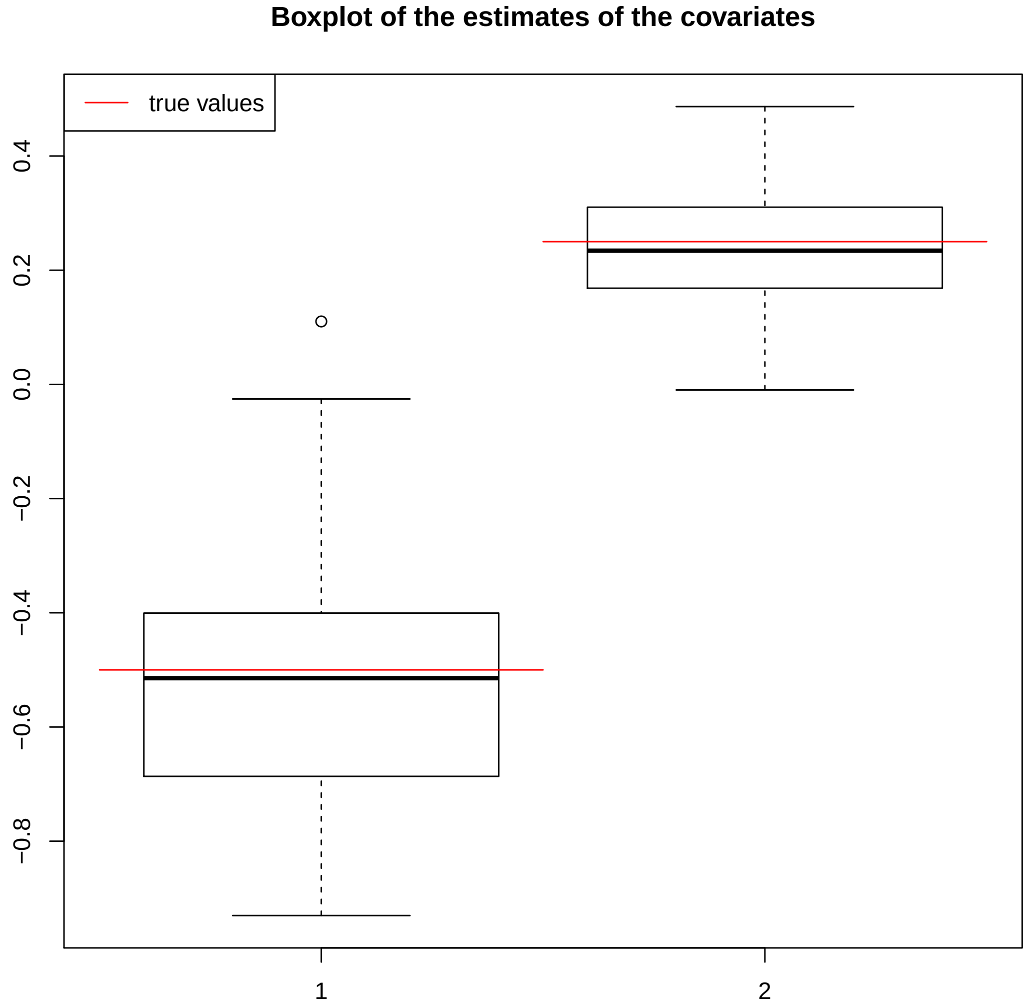


Figure 6.14: Boxplots of the coefficient estimates in the case of binary responses. The meshes were constrained by the location points and the smoothing parameter λ was set to 1.2.

covariates are generated following the model until $\mu_i > 0$. We used the canonical link function which is natural choice. We have only implemented canonical link functions but other link could be useful and easily implemented.

First, we show the estimated function. We can see in Figure 6.15 that the mean of the function is close to the real function but seems not to be unbiased everywhere. the estimations at the extremes of the C-shaped domain are more biased than in the middle. Moreover, the bias is not symmetric, since the underlying function is not either. We can see the errors in L_1 norm between the true and the estimated mean by regions on the Figure 6.16. We see that the errors are not uniformly distributed over the domain. Again, we can see how important the link function influence the estimated mean.

Since the parameter was automatically chosen, we had to use a tuning constant γ that penalizes the equivalent degree of freedom (see section 5.2), set in this case to 1.8.

Finally, we can see on the Figure 6.17 that the parameters estimation are quite good.

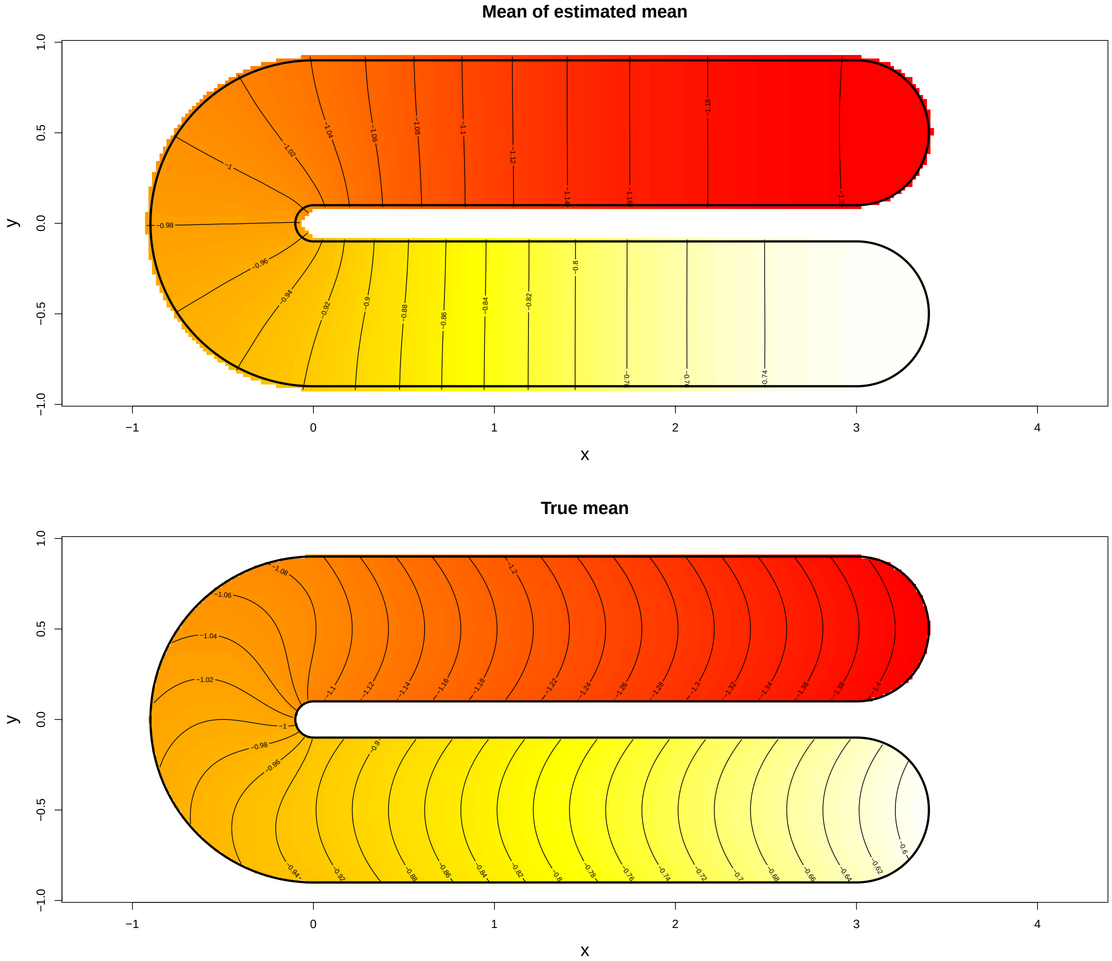


Figure 6.15: Mean of the estimated spatial function and of true function for gamma distributed data, with covariates. The meshes were constrained by the location points and the smoothing parameter λ was chosen by GCV selection.

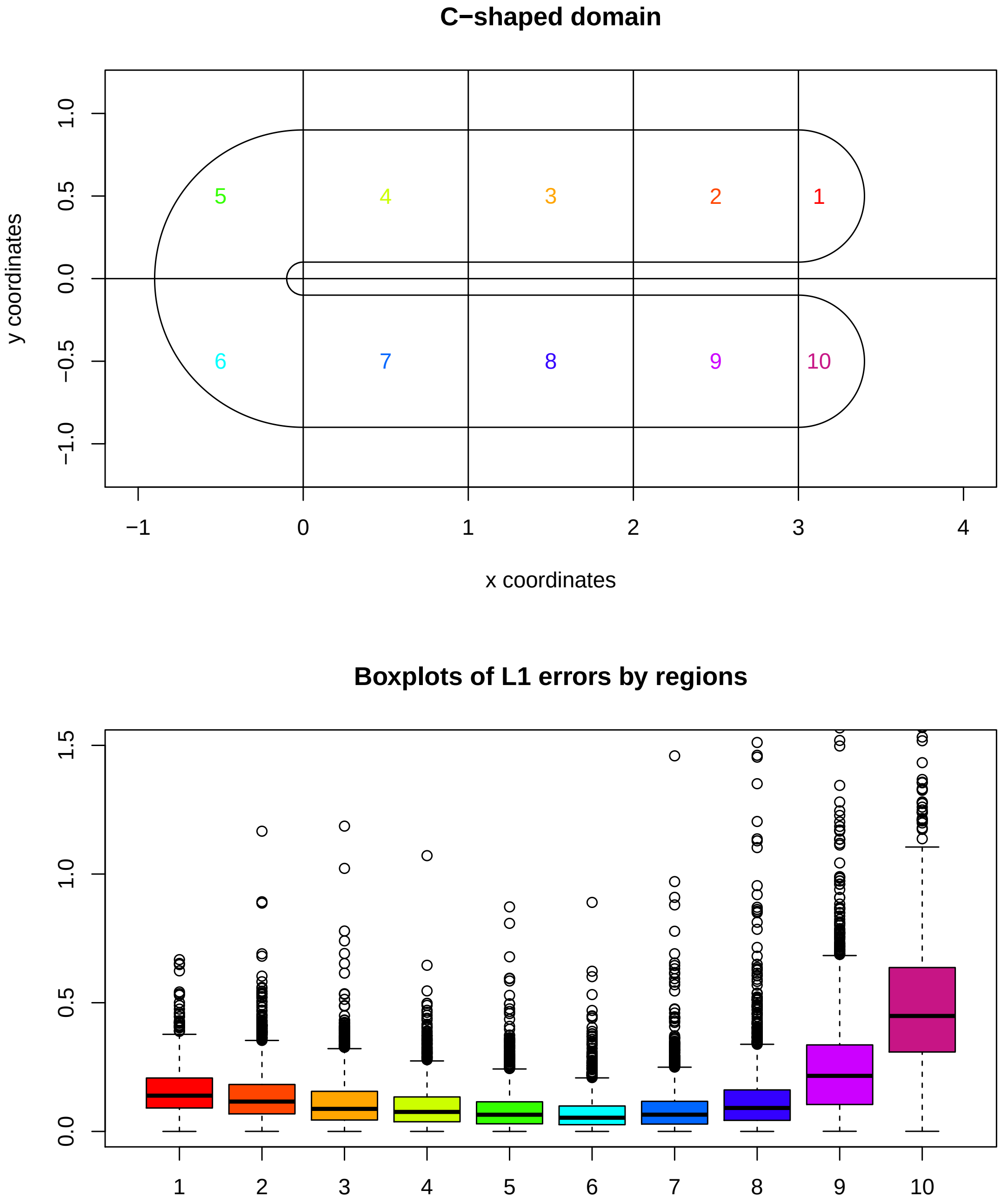


Figure 6.16: Gamma responses with covariates. Top: subdivision of the C-shaped domain. Bottom: for the 20000 points of all the simulations, L_1 error between true and estimated mean per region. The smoothing parameter λ was chosen using the GCV criterion.

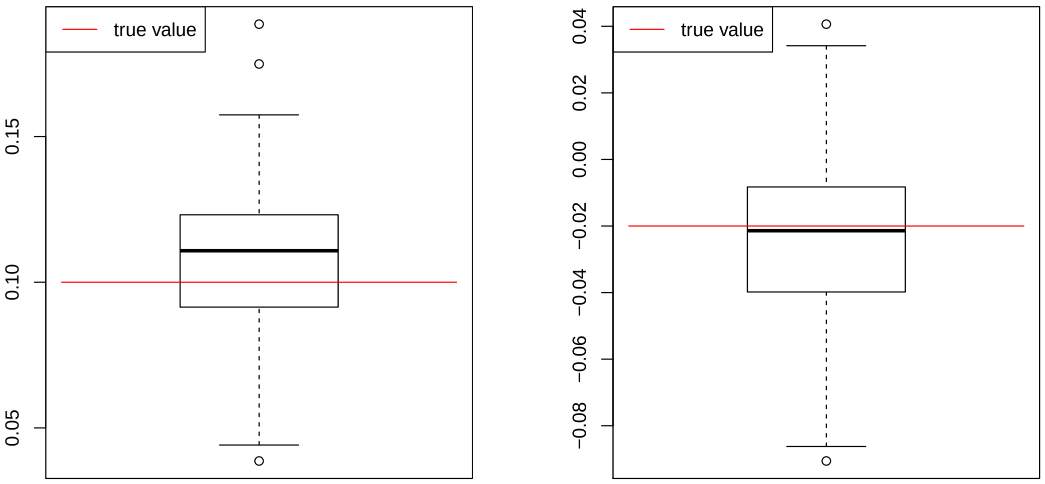


Figure 6.17: Boxplots of the coefficient estimates in the case of gamma responses. The meshes were constrained by the location points and the smoothing parameter λ was chosen using the GCV criterion.

6.4 Confidence intervals and distributional results

In this section (and only in this section), we denote by $\boldsymbol{\beta}$ the total vector of parameter containing the parametric part, that is $(\boldsymbol{\beta}, \mathbf{f})^t$ in the notation used until now. So we do not make any difference between the parametric part and the functional part.

Since the parameter estimates can be written as a linear transformation of the pseudo-data at convergence (see section 3.4), we can assume that asymptotically, the distribution of the estimates are normally distributed. We investigate this assumption only via a simulation since a theoretical result is not available (yet?). We assume that

$$\mathbb{E}[\hat{\boldsymbol{\beta}}] = \boldsymbol{\beta}$$

and that

$$\mathrm{var}(\hat{\boldsymbol{\beta}}) = \mathbf{V}_e = \left(\mathbf{X}^t\mathbf{W}\mathbf{X} + \lambda\mathbf{S}\right)^{-1} \mathbf{X}^t\mathbf{W}\mathbf{X} \left(\mathbf{X}^t\mathbf{W}\mathbf{X} + \lambda\mathbf{S}\right)^{-1} \phi.$$

We compare a normal distribution with such mean and variance to the empirical distribution of a sample.

We test the normality of the estimates. Using a same framework as the one presented in the section 6.3.1, we simulate 100 repetitions of a given simulation that is, we generated 200 points and a design matrix for these points. So, for $i = 1, \ldots, 200$, we have,

$$p_i = g_{\text{logit}}^{-1}(\mathbf{x}_i^t\beta + f(\mathbf{p}_i)), \quad \mathbf{x}_i^t \overset{\text{i.i.d.}}{\sim} \mathcal{N}\left(\begin{pmatrix} 3 \\ 7 \end{pmatrix}, \begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}\right).$$

Then, we have simulated 100 repetitions for every point, that is we generated

$$y_i^k \overset{\text{i.i.d.}}{\sim} \text{Bernoulli}(p_i), \quad \forall i = 1, \ldots, 200, \quad \forall k = 1, \ldots, 100.$$

Finally, we choose at random 4 component (i.e., 4 points in the domain) of the estimated function to examine the distribution of the estimates. First we examine the normality of the empirical distribution of these estimates using a QQ-plot. We can see in Figure 6.18 that approximately the hypothesis of normality seems to be reasonable. We then examine how good are the estimates of the mean and the variance for the components we choose.

We can see on the Figure 6.19 that the estimates are usually close to the true value of the function. So the estimation of the mean seems to be good. So even if it seems not to be always unbiased, the estimation of the mean seems to be reasonable.

We then examine the performance of the estimation of the variance. First, we compute the variance-covariance matrix $\mathbf{V}_e$. We need to compute the weight matrix $\mathbf{W}$ for the true value of the mean, which is known. We can see in Figure 6.20 that the variance estimate are very good. We can then conclude that the normality seems to be a good approximation. The main problem is the lack of theoretical arguments to assess this approximation.

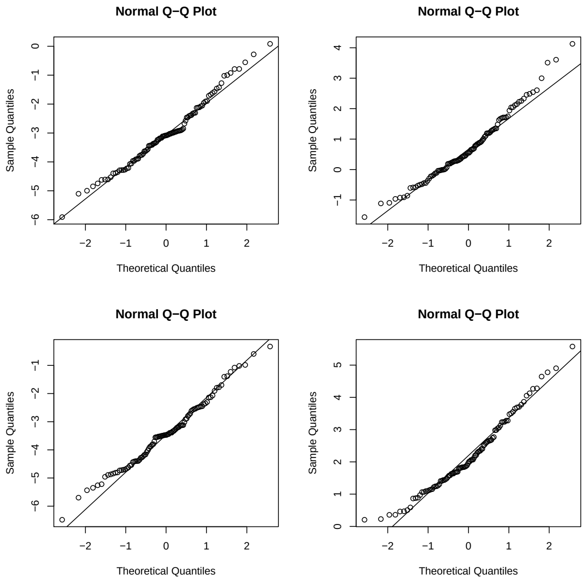


Figure 6.18: QQ-plots for 4 components of the estimates of the function.

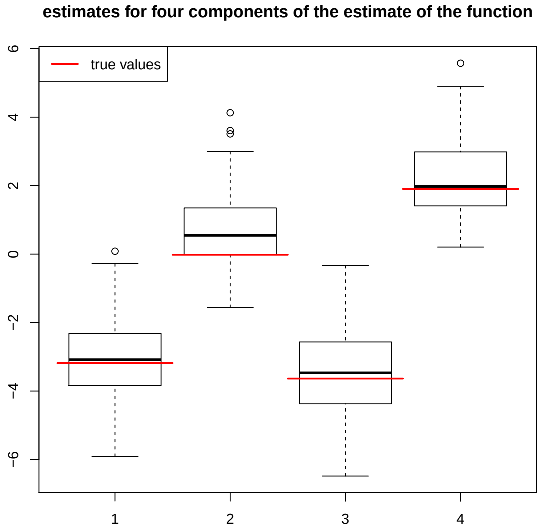


Figure 6.19: Boxplots of the estimates of 4 components of the function.

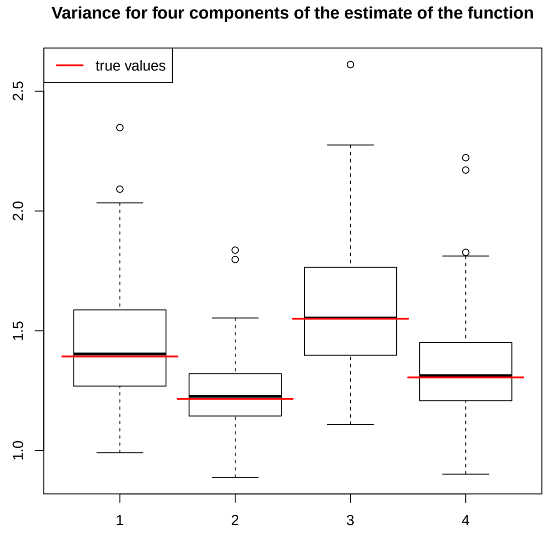


Figure 6.20: Boxplots of the variance of the estimates of 4 components of the function. The true value is the one according to the asymptotic normal distribution.

An alternative is to consider a bayesian approach. This is natural since the penalization represent a prior belief of the parameter to be estimated. This alternative is developed in details in Wood (2006).

It is important to mention the recent work of Marra and Wood (2012) about coverage properties of the confidence intervals for the GAM components. This explore new methods to derive confidence intervals for the context we are studying.

6.5 Comparison with other methods

The challenge of our method is to choose a suitable finite dimensional space to approximate the functional space $\mathcal{F}$, where f lies. Several possibilities exist. The mainly used in this context are the *thin plates regressions splines* (TPRS) (Duchon, 1977). A more recent possibility is the so called *soap film smoothing* (Wood et al., 2008). Actually, all of these methods can be seen as a particular case of a GAM, since only the finite dimensional space used is different. For the GAM estimation with TPRS and soap film, we used the **R** package `mgcv` (Wood, 2000, 2003, 2004, 2006; Wood et al., 2008; Wood, 2011) in wich all the methods to fit a GAM are available. We compare only the Rooted Mean Square Errors (RMSE) of the three methods, for each replications.

We used the a similar experimental design as in section 6.3.1, i.e., binary responses with covariates and automatic selection of the smoothing parameter.

First, we can see in Figure 6.21 that FEM and soap film are clearly better than TPRS. We can also see that soap film smoothing is a bit better than FEM. Since in the case of Gaussian distribution, soap film smoothing method and FEM splines have similar results (Sangalli et al., 2013), we suppose that the difference between the RMSE is mainly due to the smoothing parameter choice. The way we implemented the smoothing parameter choice is far from optimal on the contrary of the implementation in the package `mgcv`. That could explain this small difference between the RMSE of both methods.

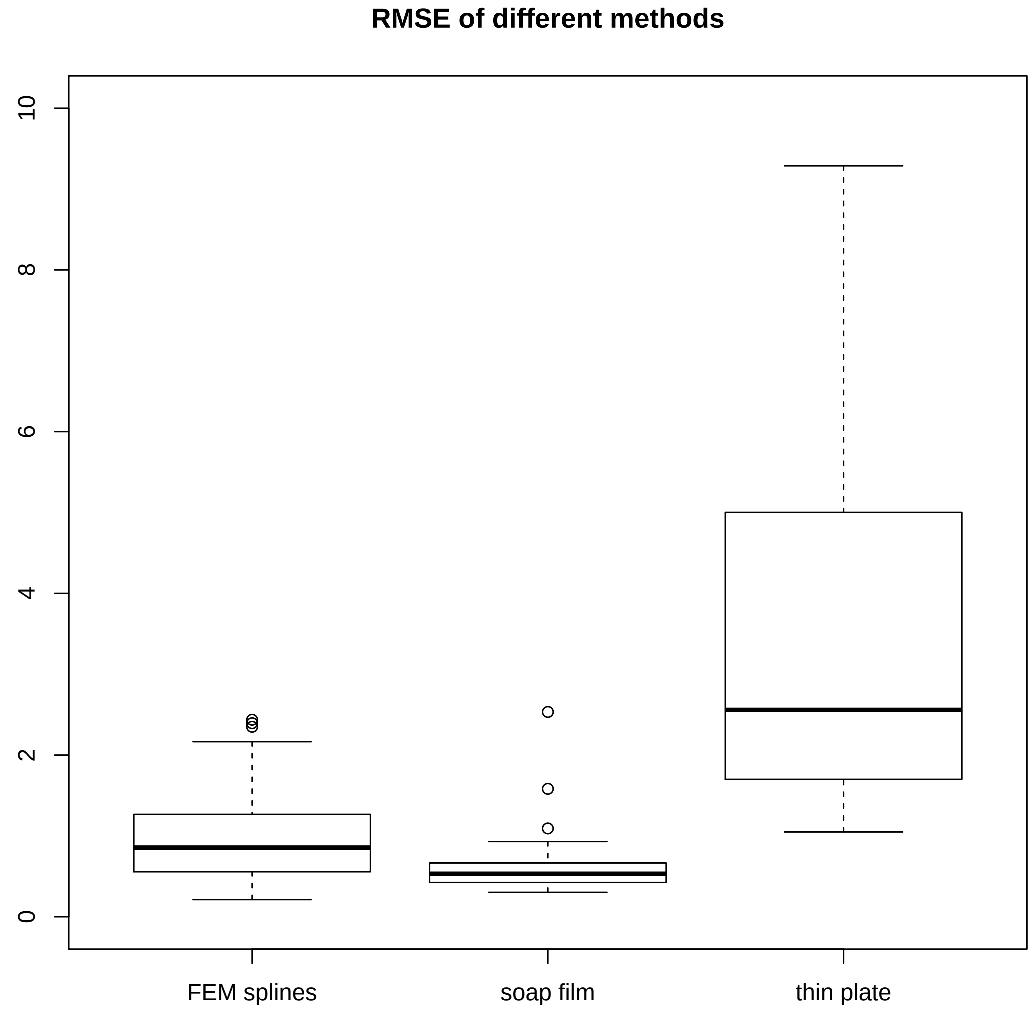


Figure 6.21: Boxplots of the RMSE in the case of binary responses with binary covariates. The meshes were constrained by the location points and the smoothing parameter λ was chosen using the GCV criterion.

Chapter 7

Application: Pennsylvania lung cancer

In order to illustrate our method, we use a classical data set available in the **R** package `SpatialEpi` (Chen et al., 2013). We used the **R** package `maps` (Becker et al., 2013) to get a the map of the Pennsylvania.

The data contains the number of lung cancer for every county of the Pennsylvania, per category of age, gender and race (white people/non white people), for the year 2002. Moreover, the total population and smoking rates in each county is also supplied. We modelled the number of cases of lung cancer per county, denoted N_j as the following:

$$\log((\mathbb{E}[N_j])) = \texttt{smoking rates}_j\ \beta_1 + \texttt{population}_j\ \beta_2 + f(\mathbf{p}_j),$$

where $\texttt{smoking rates}_j$ is the smoking rate (not in percent), $\texttt{population}_j$ is the population of the county j in thousands and $\mathbf{p}_j$ is the centroid of the county j (GPS coordinates). We can see in Figure 7.1 the mesh of the Pennsylvania corresponding to our data.

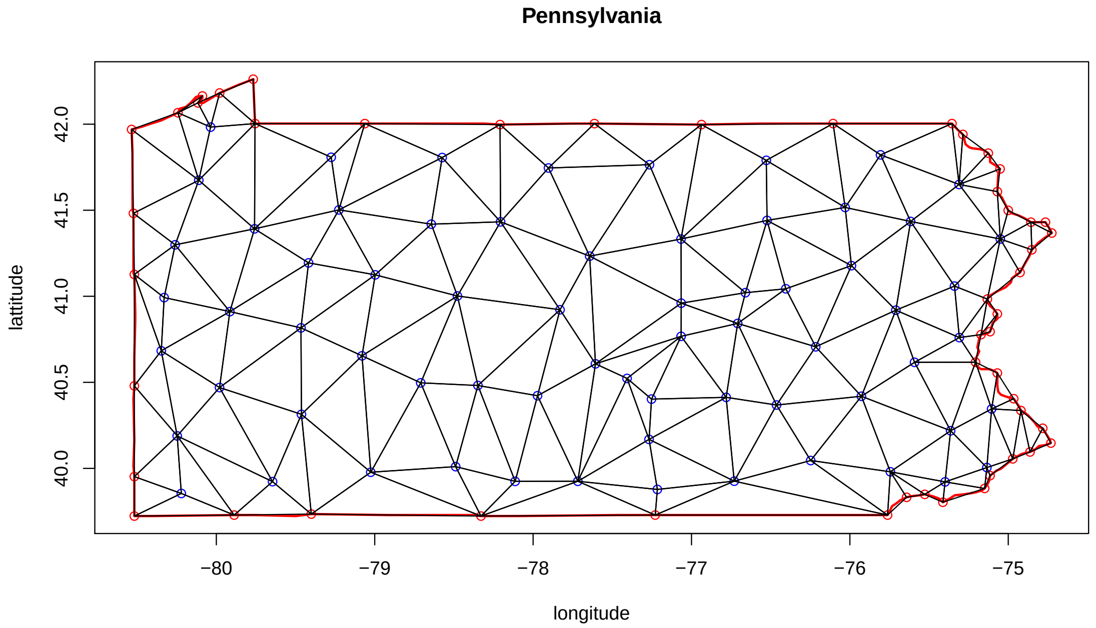


Figure 7.1: Map and mesh of the Pennsylvania State. The true boundaries are in red. We used 50 nodes on the boudaries and there is 67 internal observations, coinciding with internal nodes.

First, we have to note that since the smoking rates are in a very small interval, they will probably have the same effect for every county and not be significant. Actually, classical hypotheses tests are not available since we consider penalized likelihood. In these cases, they are not so justified as in classical contexts.

We can see a boxplot of the estimation for each covariate and for the spatial component for each observation in Figure 7.2. We can see that the greater component is due to the spatial component. The estimated coefficients are:

$\hat{\beta}_1$	$\hat{\beta}_2$
-2.146	0.001883

It is a bit surprising that the estimates β_1 is negative. But, since there is small variations in the variable `smoking rates`, the estimation of β_1 acts almost as an intercept. This could explain why the estimate is negative. Moreover, the spatial part clearly dominates the part due to the smoking rates, as it can be seen on the Figure 7.2. Finally, we can see in Figure 7.3 the spatial contribution to the log of the mean of the intensity of the Poisson process. We can compare this with a map of the Pennsylvania shown in Figure 7.4. We can see that the more rural is

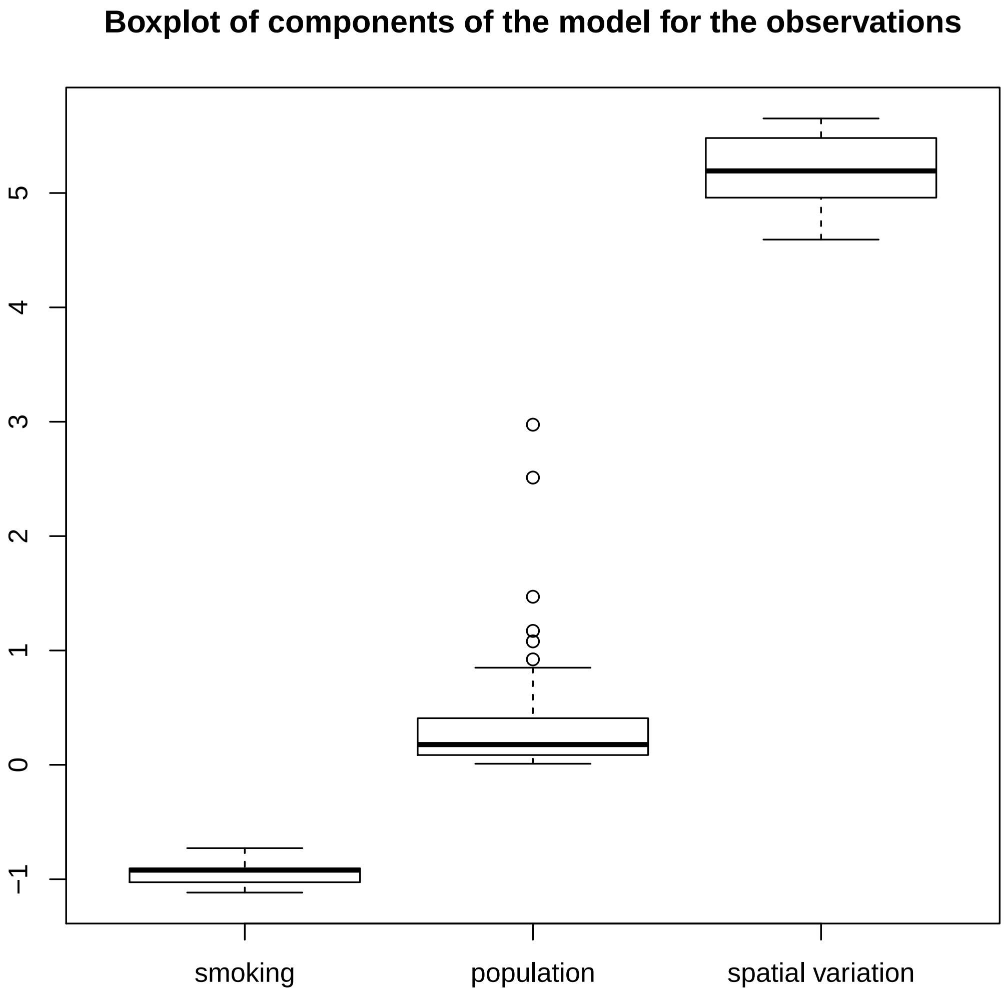


Figure 7.2: Boxplots of the different components of the estimation of the log of the mean for the counties of the Pennsylvania.

the location the higher is the spatial contribution to the intensity. Since we use the population as covariate, this is not only due to density of the population. We can the presume that urban environment seems to increase the number of lung cancer. Actually the zone with the lower spatial contribution is a zone with natural parks and the zone with a higher contributions are the urban zones of Philadelphia and Pittsburgh, the biggest urban agglomerations of the Pennsylvania. We can then assume that a natural environment decrease the expected number of lung cancers.

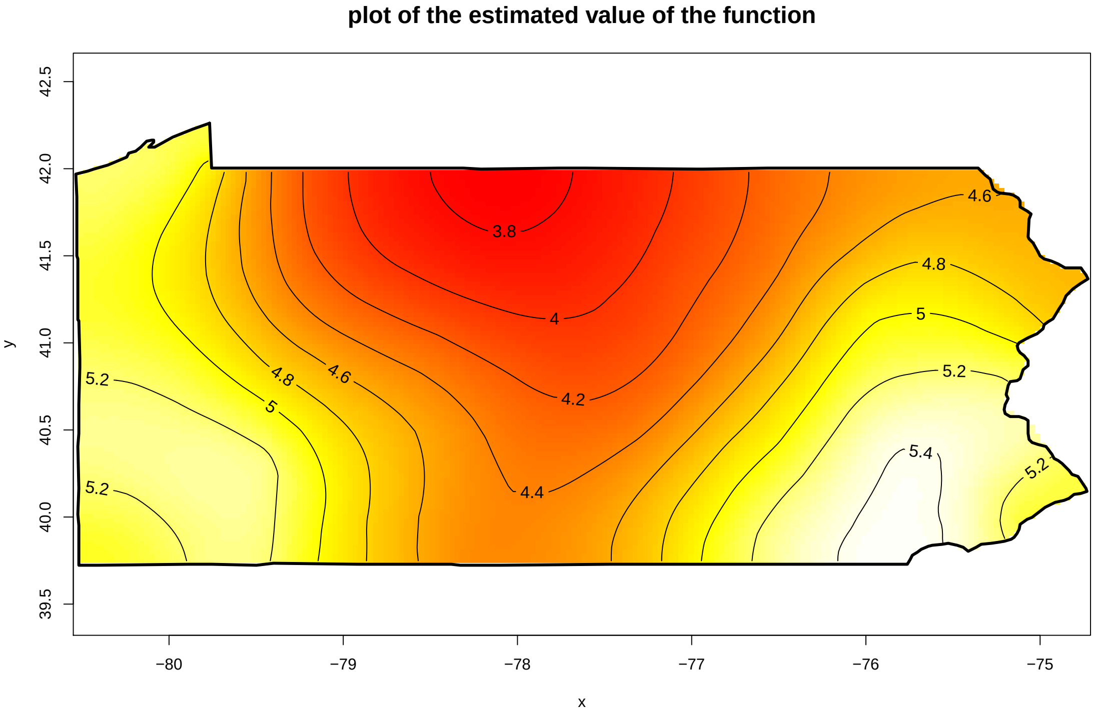


Figure 7.3: Spatial variation of the spatial component of the mean, with FEM splines. Smoothing parameter is chosen according to GCV criterion.

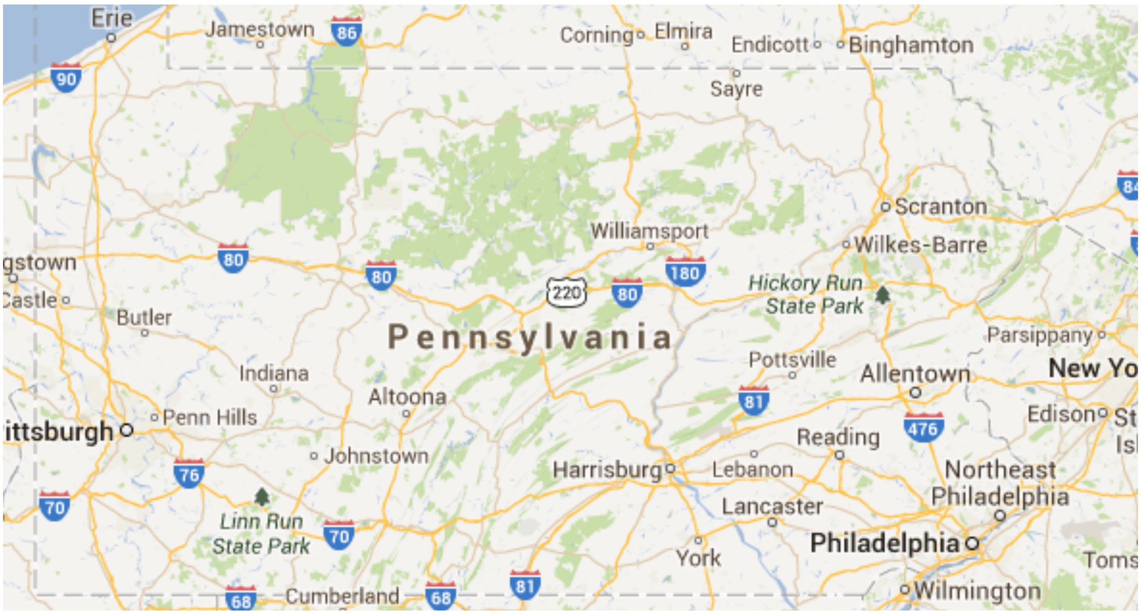


Figure 7.4: Map of the Pennsylvania, from Google map.

Chapter 8

A first approach to non planar domains

As it is done in Ettinger et al. (2012), we extend the model developed until now to surface embedded in $\mathbb{R}^3$. Let's precise the framework, the same as in Ettinger et al. (2012). We consider a surface $\Sigma \in \mathbb{R}^3$ defined by a uniformly regular continuously differentiable map X (not to be confused with a design matrix, usually denoted by $\mathbf{X}$):

$$\begin{aligned} X:\ & \Omega \subset \mathbb{R}^2 \to \Sigma \subset \mathbb{R}^3, \\ & \mathbf{u} = (u_1, u_2) \mapsto \mathbf{s} = (s_1, s_2, s_3), \end{aligned}$$

where Ω is an open and bounded set of $\mathbb{R}^2$ and the boundary $\partial\Omega$ is piecewise C^∞. We will denote by $\mathbf{u} = (u_1, u_2)$ a point of Ω and by $\mathbf{s} = (s_1, s_2, s_3)$ a point of Σ.

For a data point $\mathbf{p}_i, i = 1, \dots, n$, we consider realizations of an exponential family distribution y_i (see section 2.2) with natural parameter η_i. Moreover η_i depends on covariates $\mathbf{x}_i^t$ and on the value $f(\mathbf{p}_i)$ of a function f, where $f : \Sigma \to \mathbb{R}$. Since f is defined on Σ, the framework defined in the previous sections does not hold. We have to modify the concept of the penalization and to find a suitable way to adapt the methodology already described.

In order to impose a smoothness to the function f defined on Σ, we have first to think about the meaning of smoothness when we deal with functions defined on a riemannian manifold. Actually, we have to consider a kind of space-varying Laplacian take accounts the variation of the curvature of the manifold, which is the traditional Laplacian but defined on the tangent plan of the manifold. This generalization of the Laplace operator, called Laplace-Beltrami operator and denoted by Δ_Σ, is suitable to be the penalization term.

So in this framework the penalized log-likelihood of the model can the be written as:

$$\mathcal{L}_\Sigma(\boldsymbol{\beta}, f) = \sum_{i=1}^{n} l(y_i; \eta_i(\boldsymbol{\beta}, f)) - \lambda \int_\Sigma (\Delta_\Sigma f)^2 d\Sigma. \tag{8.1}$$

First, we have to notice that, in the case where $\Sigma = \Omega$, i.e., when X is the identity, the penalized log-likelihood (8.1) coincides with the penalized log-likelihood in a planar domain (2.2). Hence, this framework is clearly a generalization of the methodology developed in the previous sections.

In the following, we will assume that the map X is bijective and conformal. A conformal map is a map that preserves angles. This property is important for the development of the methodology. In particular, a regular triangulation of the surface Σ should be also an approximative triangulation of the domain Ω.

8.1 A geometric framework

We define the Jacobian matrix, denoted by $\nabla X(\mathbf{u})$, as the matrix with components

$$(\nabla X(\mathbf{u}))_{ij} = \frac{\partial u_j}{\partial x_i}(\mathbf{u}).$$

We then define the columns of the jacobian matrix $\nabla X(\mathbf{u})$ as X_1 and X_2, such that $\nabla X(\mathbf{u}) = (X_1, X_2)$. We can then define the metric tensor of the map X, defined for every $\mathbf{u} \in \Omega$ as:

$$\mathbf{G}(\mathbf{u}) = \nabla X(\mathbf{u})^t \nabla X(\mathbf{u}) = \begin{pmatrix} \|X_1(\mathbf{u})\|^2 & \langle X_1(\mathbf{u}), X_2(\mathbf{u})\rangle \\ \langle X_1(\mathbf{u}), X_2(\mathbf{u})\rangle & \|X_2(\mathbf{u})\|^2 \end{pmatrix}.$$

The metric tensor gives a lot of geometrical information, in particular about length and angles of mapped vectors. We can already notice that the metric tensor con be viewed as a symetric, positive definite, bilinear form and then can be considered as an inner product defined on the tangential plan at point $X(\mathbf{u})$ of the surface Σ. Actually, as any inner product, it also induces a metric on the riemannian manifold defined by the map X. In the same way as in the vectorial space $\mathbb{R}^n$, we can define an angle ϑ between two tangent vectors $\mathbf{l}_1$ and $\mathbf{l}_2$ to the manifold at a point $X(\mathbf{u})$ as:

$$\cos(\vartheta) = \frac{\langle \mathbf{l}_1, \mathbf{l}_2\rangle_{G(\mathbf{u})}}{\|\mathbf{l}_1\|_{G(\mathbf{u})}\ |\mathbf{l}_2\|_{G(\mathbf{u})}},$$

where

$$\begin{aligned} \langle \mathbf{l}_1, \mathbf{l}_2\rangle_{G(\mathbf{u})} &= \mathbf{l}_1^t G(\mathbf{u}) \mathbf{l}_2, \\ \|\mathbf{l}_1\|_{G(\mathbf{u})} &= \mathbf{l}_1^t G(\mathbf{u}) \mathbf{l}_1. \end{aligned}$$

Now, thanks to the following proposition, we can characterize a map that preserves angle (called sometimes conformal map):

Proposition 3
Let be Ω a open and bounded subset of $\mathbb{R}^2$ and a map X,$X : \Omega \subset \mathbb{R}^2 \to \Sigma \subset \mathbf{R}^3$ such that $X \in C^1(\Omega)$. Let G be the metric tensor of X. Then, the map X preserves the angles if and only if the metric tensor G is of the form:

$$G(\mathbf{u}) = r(\mathbf{u})\mathbf{I}_2, \quad \forall \mathbf{u} \in \Omega,$$

where r is a scalar continuous function defined on Ω.

We do not give a proof of this classical result of geometry (see Troyanov (2009)).

Since we have set that X is a uniformly regular map, it exists a positive constant $\alpha \in \mathbb{R}$ such that:

$$\mathcal{W}(\mathbf{u}) = \|X_1 \wedge X_2\| = \sqrt{\|\mathbf{X}_1^2\| \|\mathbf{X}_2\|^2 - \langle \mathbf{X}_1, \mathbf{X}_2\rangle^2} \geq \alpha > 0, \quad \forall \mathbf{u} \in \Omega.$$

This is equivalent to impose that the tangents vectors exists and are not collinear for every point surface Σ.

We can now define the various differential operators needed. Since the function f we consider is defined on the manifold Σ, we have to consider different kind of differential operators, suitable to be used in this situation. Let $f \circ X \in C^2(\Omega)$. First, we define the Σ-gradient of f, denoted $\nabla_\Sigma f$ as:

$$\nabla_\Sigma f(\mathbf{s}) = \nabla X(\mathbf{u}) \mathbf{G}^{-1}(\mathbf{u}) \nabla f(X(\mathbf{u})), \forall \mathbf{s} \in \Sigma,$$

where $\mathbf{u} = X^{-1}(\mathbf{s})$. We can define the Σ-divergence for a tangential vector field Y as:

$$\mathrm{div}_\Sigma Y(\mathbf{s}) = \sum_{i=1}^{2} \frac{\partial}{\partial u_i} \mathcal{W}(\mathbf{u}) Y^i(X(\mathbf{u})),$$

where Y^i denotes the ith direction of the tangential vector field. Analogously of the traditional Laplace operator, we can define the Σ-Laplace operator, also called Laplace-Beltrami operator, as:

$$\begin{aligned} \Delta_\Sigma f(\mathbf{s}) &= \mathrm{div}_\Sigma(\nabla_\Sigma f(X(\mathbf{u})) \\ &= \frac{1}{\mathcal{W}(\mathbf{u})} \sum_{i,j=1}^{2} \frac{\partial}{\partial u_i}(k_{ij} \frac{\partial}{\partial u_j} \nabla f(X(\mathbf{u}))) = \frac{1}{\mathcal{W}(\mathbf{u})} \mathrm{div}(\mathbf{K} \nabla f(X(\mathbf{u}))), \end{aligned}$$

where the matrix $\mathbf{K}$ is defined as $\mathbf{K}(\mathbf{u}) = \mathcal{W}(\mathbf{u})\mathbf{G}^{-1}(\mathbf{u})$, whose components are denoted by $k_{ij}(\mathbf{u})$.

Since we assume that X is a conformal map, the metric tensor $\mathbf{G}(\mathbf{u})$ is a multiple of the identity and hence is also a multiple of the identity (see proposition 3). Then, we get that $\mathbf{K}(\mathbf{u}) = \mathbf{I}_2$ and hence, we get:

$$\Delta_\Sigma f(\mathbf{s}) = \frac{1}{\mathcal{W}(\mathbf{u})} \Delta(f \circ X)(\mathbf{u}), \forall \mathbf{s} \in \Sigma,$$

where, $\mathbf{u} = X^{-1}(\mathbf{s})$. We can reformulate the penalized log-likelihood in terms of the function $f \circ X \in C^2(\Omega)$ as:

$$\mathcal{L}_\Omega(\boldsymbol{\beta}, f \circ X) = \sum_{i=1}^{n} l(y_i; \eta_i(\boldsymbol{\beta}, f \circ X)) - \lambda \int_\Omega \left(\frac{1}{\mathcal{W}^{1/2}(\mathbf{u})} \Delta(f \circ X) \right)^2 d\Omega. \tag{8.2}$$

We have then showed our previous problem simplifies to the problem of finding a function $f \circ X$ that maximize this penalized log-likelihood. Since $f \circ X$ is a function defined on Ω, the problem of finding f that maximize (8.1) is of the same form as the problem (2.2). So we can apply the methodology developed in the previous sections.

We notice that this extension strongly depends on the assumption of the existence of a conformal map. Two facts must then be highlighted. The first is that we need the map to be bijective. For a same surface, it exists infinitely different parametrisations. So for a given surface, a suitable map must be used. The suitability, in this context, is the requirement of conformity and invertibility. The conformity has two advantages. The first is that the mesh used on the surface Σ is not deformed by the map. In practice, since we work with a mesh of the surface, without a conformal map, the image on Ω of the mesh could not be a suitable mesh for the domain Ω. The second is that the penalization simplifies to a traditional Laplacian operator (up to a factor $1/\mathcal{W}(\mathbf{u})$).

8.2 Finite element solution in the case of surface embedded in $\mathbb{R}^3$

From now, we assume that we have a bijective and conformal map $X : \Omega \subset \mathbb{R}^2 \to \Sigma \subset \mathbb{R}^3$. We use the same notation as in section 8.1. We have to minimize the penalized log-likelihood (8.2). Following a similar derivation as in section 4.1, we identify the symmetric, bilinear $m(\cdot, \cdot)$ form which represent the penalization as:

$$m(s, t) = \int_\Omega \frac{1}{\mathcal{W}(\mathbf{u})} \Delta(s \circ X) \Delta(t \circ X) d\Omega.$$

For a given smoothing parameter λ and a given estimation $\boldsymbol{\mu}$, this leads to minimize at each step of the PIRLS algorithm the following functional:

$$\tilde{J}_\Omega(\boldsymbol{\beta}, f \circ X) = \|\mathbf{W}^{1/2}(\mathbf{z} - \mathbf{X}\boldsymbol{\beta} - f(X(\mathbf{u}))_n)\|^2 + \lambda \int_\Omega \frac{1}{\mathcal{W}(\mathbf{u})}(\Delta(f \circ X))^2 d\Omega,$$

where

$$f(X(\mathbf{u}))_n = ((f \circ X)(\mathbf{u}_1), \dots, (f \circ X)(\mathbf{u}_n))^t, \text{ and } \mathbf{u}_i = X^{-1}(\mathbf{p}_i), \quad \forall i = 1, \dots, n,$$

and

$$\mathbf{z} = \boldsymbol{\eta} + \mathbf{G}^{-1}(\mathbf{y} - \boldsymbol{\mu}), \quad \mathbf{G} = \text{diag}(g(\boldsymbol{\mu})), \text{ and } \mathbf{W} = \text{diag}\left(\frac{1}{V(\mu_1)g'(\mu_1)^2}, \dots, \frac{1}{V(\mu_n)g'(\mu_n)^2}\right)^t.$$

Now, setting $h = f \circ X$, and $\mathbf{h}_n = f(X(\mathbf{u}))_n$, the problem then becomes similar to (2.4). The derivation of the finite element formulation of the problem is then straightforward, following a similar methodology as in the section 3.3.

Chapter 9

Conclusion and perspective

This work extends the methodology developed in Sangalli et al. (2013) to other distribution than the Gaussian. It is broadly applicable and hence very interesting. The model seems to have very good inferential properties. Moreover, it is computationally very efficient despite the iterative method used in the estimation process. From a theoretical point of view, the use of a finite element space is very useful for different reasons. First, it can deal with complex geometries, such as we have seen. Moreover, it can take account of different boundary conditions (even if we do not adress this issue in this work, see (Azzimonti, 2013) for more details), can be extended to surfaces embedded $\mathbb{R}^3$ and can deal with different kind of penalization terms (Ettinger et al., 2012; Azzimonti, 2013).

Future work could be done in two directions. First, in a practical direction. According to the author, it is essential to create a **R** package that implement all the methods developed in this work and in the field of spline regression using finite element spaces. A patch to the package `mgcv` that implement the finite element spaces could be a smart (and easier) alternative to the creation of a **R** package. In this way, all the routines developed in the package `mgcv` such as (multiple) smoothing parameter estimation could also be available.

The second direction of a future work is more theoretical. First, demonstration of consistence of the estimators developed in the field of spatial spline regression with finite element spaces would be a great contribution. But extension of the model developed in this thesis to other kind of penalization could also be done.

Bibliography

L. Azzimonti. *Blood flow velocity field estimation via spatial regression with PDE penalization.* PhD thesis, Politecnico di Milano, 2013.

Richard A. Becker, Allan R. Wilks, R. Brownrigg, and Thomas P. Minka. *maps: Draw Geographical Maps*, 2013. URL `http://CRAN.R-project.org/package=maps`. R package version 2.3-2.

P. J. Bickel, C. A.J. Klaassen, R. Ya'acov, and J. A. Wellner. *Efficient and Adaptative Estimation for Semiparametric Models.* Springer-Verlag New York, 1993.

C. Chen, A. Y. Kim, M. Ross, J. Wakefield, and E. S. Venkatraman. *SpatialEpi: Performs various spatial epidemiological analyses*, 2013. URL `http://CRAN.R-project.org/package=SpatialEpi`. R package version 1.1.

S-W Cheng, T. K. Dey, and J. C. Shewchuk. *Delaunay Mesh Generation.* Computer and Information Series. Chapman & Hall/CRC, 2013.

J. Duchon. Splines minimizing rotation-invariant semi-norms in sobolev spaces. In Walter Schempp and Karl Zeller, editors, *Constructive Theory of Functions of Several Variables*, volume 571 of *Lecture Notes in Mathematics.* Springer Berlin Heidelberg, 1977.

B. Ettinger, S. Perotto, and L. M. Sangalli. Spatial regression models over two-dimensional manifolds. TechRep. MOX 54/2012, Dipartimento di Matematica, Politecnico di Milano, 2012.

C. Felippa. Introduction to finite element methods online book, appendix f, 2012.

T. J. Hastie and R. J. Tibshirani. Generalized additive models. *Statistical Sciences*, 23(4), 1986.

T. J. Hastie and R. J. Tibshirani. *Generalized Additive Models.* Monographs on Statistics and Applied Probability Series. Chapman & Hall, CRC Press, 1990.

G. Marra and S. N. Wood. Coverage properties of confidence intervals for generalized additive models components. *Scandinavian Journal of Statistics*, 939, 2012.

MATLAB. *version 8.1.0 (R2013a).* The MathWorks Inc., Natick, Massachusetts, 2013.

P. McCullagh and J. A. Nelder. *Generalized Linear Models.* Monographs on Statistics and Applied Probability. Chapman & Hall, second edition edition, 1989.

A. Quarteroni. *Numerical Models for Differential Problems.* MS&A (Series). Springer-Verlag Milan, 2009.

R Core Team. *R: A Language and Environment for Statistical Computing.* R Foundation for Statistical Computing, Vienna, Austria, 2012. URL `http://www.R-project.org/`. ISBN 3-900051-07-0.

J. O. Ramsay and B. W. Silverman. *Functional Data Analysis.* Springer Series in Statistics. Springer, 2005.

T. O. Ramsay. Spline smoothing over difficult regions. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64(2), 2002.

L. M. Sangalli, J. O. Ramsay, and T. O. Ramsay. Spatial spline regression models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 75, 2013.

M. Troyanov. *Cours de géométrie.* Enseignement des mathématiques. Presses polytechniques et universitaires romandes, 2009.

G. Wahba. *Spline Models for Observational Data.* CBMS-NSF Regional Conference Series in Applied Mathematics. Society for Industrial and Applied Mathematics, 1990.

S. N. Wood. Modelling and smoothing parameter estimation with multiple quadratic penalties. *Journal of the Royal Statistical Society (B)*, 62(2):413–428, 2000.

S. N. Wood. Thin-plate regression splines. *Journal of the Royal Statistical Society (B)*, 65(1): 95–114, 2003.

S. N. Wood. Stable and efficient multiple smoothing parameter estimation for generalized additive models. *Journal of the American Statistical Association*, 99(467):673–686, 2004.

S. N. Wood. *Generalized additive models: an introduction with application in R.* Springer Series in statistics. Springer, 2006.

S. N. Wood. Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 73(1), 2011.

S. N. Wood, M. V. Bravington, and S. L. Hedley. Soap film smoothing. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70(5), 2008.