UNIVERSIDAD CARLOS III DE MADRID working papers

Working Paper 13-30
Statistics and Econometrics Series 26
Septiembre 2013

Departamento de Estadística
Universidad Carlos III de Madrid
Calle Madrid, 126
28903 Getafe (Spain)
Fax (34) 91 624-98-49

FAST ALGORITHM FOR SMOOTHING PARAMETER SELECTION IN MULTIDIMENSIONAL GENERALIZED P-SPLINES

María Xosé Rodríguez-Álvarez[1],Dae-Jin Lee[2], Thomas Kneib[3],
María Durbán[4], Paul Eilers[5]

Abstract:

A new computational algorithm for estimating the smoothing parameters of a multidimensional penalized spline generalized model with anisotropic penalty is presented. This new proposal is based on the mixed model representation of a multidimensional P-spline, in which the smoothing parameter for each covariate is expressed in terms of variance components. On the basis of penalized quasi-likelihood methods (PQL), closed-form expressions for the estimates of the variance components are obtained. This formulation leads to an efficient implementation that can considerably reduce the computational load. The proposed algorithm can be seen as a generalization of the algorithm by Schall (1991) - for variance components estimation - to deal with non-standard structures of the covariance matrix of the random effects. The practical performance of the proposed computational algorithm is evaluated by means of simulations, and comparisons with alternative methods are made on the basis of the mean square error criterion and the computing time. Finally, we illustrate our proposal with the analysis of two real datasets: a two dimensional example of historical records of monthly precipitation data in USA and a three dimensional one of mortality data from respiratory disease according to the age at death, the year of death and the month of death.

Keywords: Smoothing; P-splines; Tensor product; Anisotropic penalty; Mixed Models.

[1] Deptartment of Statistics and Operations Research. University of Vigo, Spain

[2] CSIRO Mathematics, Informatics and Statistics, Clayton, VIC, Australia

[3] Department of Economics, Georg August University Göttingen, Germany

[4] Department of Statistics, Universidad Carlos III de Madrid, Spain

[5] Erasmus Medical Center, Rotterdam, The Netherlands

Fast algorithm for smoothing parameter selection in multidimensional generalized P-splines

María Xosé Rodríguez-Álvarez[1] Dae-Jin Lee[2] Thomas Kneib[3]
María Durbán[4] Paul Eilers[5]

[1] Deptartment of Statistics and Operations Research. University of Vigo, Spain

[2] CSIRO Mathematics, Informatics and Statistics, Clayton, VIC, Australia

[3] Department of Economics, Georg August University Göttingen, Germany

[4] Department of Statistics, Universidad Carlos III de Madrid, Spain

[5] Erasmus Medical Center, Rotterdam, The Netherlands

July 29, 2013

Abstract

A new computational algorithm for estimating the smoothing parameters of a multidimensional penalized spline generalized model with anisotropic penalty is presented. This new proposal is based on the mixed model representation of a multidimensional P-spline, in which the smoothing parameter for each covariate is expressed in terms of variance components. On the basis of penalized quasi-likelihood methods (PQL), closed-form expressions for the estimates of the variance components are obtained. This formulation leads to an efficient implementation that can considerably reduce the computational load. The proposed algorithm can be seen as a generalization of the algorithm by Schall (1991) - for variance components estimation - to deal with non-standard structures of the covariance matrix of the random effects. The practical performance of the proposed computational algorithm is evaluated by means of simulations, and comparisons with alternative methods are made on the basis of the mean square error criterion and the computing time. Finally, we illustrate our proposal with the analysis of two real datasets: a two dimensional example of historical records of monthly precipitation data in USA and a three dimensional one of mortality data from respiratory disease according to the age at death, the year of death and the month of death.

KEY WORDS: Smoothing; P-splines; Tensor product; Anisotropic penalty; Mixed Models.

1 Introduction

Roughness penalty smoothing has become the most popular method for performing nonparametric regression. However, this methodology depends on a key step: the selection of the smoothing parameter which controls the trade of between fidelity to the data and smoothing.

There are two main approaches to smoothing parameter selection: the one based on the optimization of some criteria such as Akaike Information Criteria (AIC) or Generalized cross-validation (GCV) (see e.g Eilers and Marx 1996, Wood 2004; 2008), and the one in which the smooth function is treated as random, and the smoothing parameters estimated by maximum likelihood (ML), or restricted maximum likelihood (REML) (Fahrmeir et al. 2004, Ruppert et al. 2003, Wood 2011). When the model includes several smooth functions (additive model), the computational burden increases rapidly with the number of smoothing parameters to be chosen, and the minimization procedure can become unstable. Several algorithms have been developed to achieve numerical stability and improve the computational time. Most of these algorithms are in the framework of GCV, some are based on matrix factorizations (Wood 2004), or use full Newton method (Wood 2008) rather than iterative re-weighted least squares. More recently, Wood (2011) proposed a stable nested iteration method for REML or ML, that proved to outperform previous approaches in this contex.

When it came to extending the aforementioned approaches to the estimation of multidimensional interaction surfaces, low-rank tensor product smoothers have become the general approach (Eilers and Marx 2003, Wood 2006b). Its popularity is primarily due to the flexibility that tensor product smoothers provide, mainly by the posibility of incorporating anisotropic penalizations. However, in this contex one is faced with the challenge of making estimation feasible from a computational point of view. Moreover, for the REML/ML-based estimation approaches one is also faced with the fact that estimation of the variance components can not be accomodated using standard mixed modelling software. Although estimation can be done by numerical maximization of the (restricted) log - likelihood (Fahrmeir et al. 2004, Wood 2006b; 2011), it has the drawback of being computationally demanding, specially for large datasets. Very recently, Wood et al.

(2013) and Lee et al. (2013) have proposed an alternative method for the estimation of a tensor-product smoother with anisotropic penalizations. Both approaches are based on the decomposition of the multidimensional smooth term in different terms that only depend on one smoothing parameter. Although both approaches have proved to be useful, the development of efficient and fast algorithms to deal with proper anisotropic penalizations are still challenging, in particular for more than two covariates.

This paper is devoted to present a new computational algorithm for estimating the smoothing parameters of a multidimensional tensor product penalized spline (P-spline) generalized model with anisotropic penalizations on the basis of the mixed model formulation. Following the ideas presented in Harville (1977) and Schall (1991), we derive closed-form expressions for the estimates of the variance components. The algorithm is, therefore, straightforward to implement in practice. Moreover, some characteristics of the derived expressions can be used to improve even further the computational time, thus rendering very good computing times.

The rest of this paper is organized as follows: in Section 2 a brief introduction to low-rank tensor product P-splines models and its representation as a mixed model is presented. For the sake of illustration, we primarily focus our attention on a two-dimensional (2D) genenalized P-spline. However, smoothing in more that two dimensions can be also accommodated. Once the needed background and notation has been introduced, we then describe in detail our approach in Section 3. In Section 4, we present some extesions of the proposed algorithm. Specifically, we describe the extension to the three dimensional (3D) case and to generalized additive mixed models (GAMM, Lin and Zhang 1999). A simulation study devoted to evaluate the practical performance of the proposed algorithm is discussed in Section 5. We illustrate our method in Section 6, using two real examples, and conclude with a discussion in Section 7. Some technical details have been added by way of appendices.

2 2D low-rank tensor product smoothers

Consider a bidimensional *generalized* regression problem in which observations on the ith of n independent units consists of a univariate response variable y_i and a 2D covariate vector $\mathbf{x}_i = (x_{i1}, x_{i2})^t$

$$g\left(E[y_i|\mathbf{x}_i]\right) = g\left(\mu_i\right) = \eta_i = f\left(x_{i1}, x_{i2}\right), \tag{1}$$

where f is a smooth and unknown function, and g is a monotonic link function. Here, we asume that y_i follows an exponential family distribution, where $\mathbb{V}\text{ar}(y_i|\boldsymbol{x_i}) = \phi\nu\left(\mu_i\right)$, with ν being the variance function that is determined by the exponential family the response variable belongs to, and ϕ is a dispersion parameter that may be known or unknown.

Within the P-spline framework of Eilers and Marx (1996; 2003), the unknown surface $f(x_1, x_2)$ can be approximated by the tensor product of two univariate low-rank spline bases, i.e.,

$$f(x_1, x_2) = \sum_{j=1}^{c_1}\sum_{k=1}^{c_2}\theta_{jk}B_{1j}\left(x_1\right)B_{2k}\left(x_2\right),$$

where B_j^1 and B_k^2 are the univariate basis functions of x_1 and x_2 respectively (as e.g B-splines (de Boor 2001) or thin plate regression splines (Wood 2003)), and θ_{jk} is a vector of regression coefficients. Let's denote $\boldsymbol{B}_1$ and $\boldsymbol{B}_2$ the marginal model matrices for the covariate values $\boldsymbol{x}_1 = \left(x_{11}, \ldots, x_{n1}\right)^t$ and $\boldsymbol{x_2} = \left(x_{11}, \ldots, x_{n2}\right)^t$ respectively. Then, in matrix notation, model (1) can be expressed as

$$g\left(\boldsymbol{\mu}\right) = \boldsymbol{\eta} = \boldsymbol{B\theta}, \tag{2}$$

where $\boldsymbol{B} = \boldsymbol{B}^2\square\boldsymbol{B}^1$ is the full regression matrix (with $\square$ denoting the 'row-wise' kronecker product), $\boldsymbol{\mu} = \left(\mu_1, \ldots, \mu_n\right)^t$, $\boldsymbol{\eta} = \left(\eta_1, \ldots, \eta_n\right)^t$, and $\boldsymbol{\theta} = \left(\theta_{11}, \ldots, \theta_{c_11}, \ldots, \theta_{c_1c_2}\right)^t$.

In the context of P-splines, smoothness is achieved by imposing a penalty on the regression coefficients $\boldsymbol{\theta}$ in the form $\boldsymbol{\theta}^t\breve{\boldsymbol{P}}\boldsymbol{\theta}$, where $\breve{\boldsymbol{P}}$ is the penalty matrix. For P-spline smoothing in more than one dimension, one is faced with the decision to either assume the same amount of smoothing for all the covariates (an isotropic penalization), or to allow different smoothness on each covariate (an anisotropic penalization). Whereas the isotropy could be justified when modelling, for instance, a smooth function of latitude and longitude, this is not always the case when the covariates, e.g. x_1 and x_2, are measured in different units.

In this paper we assume an anisotropic penalization, i.e., a different amount of smoothing for x_1 and x_2. Acccordingly, the penalty matrix is then given by (see, e.g., Eilers et al. 2006)

$$\breve{\boldsymbol{P}} = \lambda_1\boldsymbol{I}_{c_2}\otimes\breve{\boldsymbol{P}}_1 + \lambda_2\breve{\boldsymbol{P}}_2\otimes\boldsymbol{I}_{c_1}, \tag{3}$$

where $\otimes$ denotes the kronecker product, $\boldsymbol{I}_k$ is an identity matrix of dimension k, λ_d is a smoothing parameter that controls the amount of smoothing along the covariate x_d, and

$\breve{\boldsymbol{P}}_d$ are $c_d \times c_d$ positive semidefinite matrices of rank $(c_d - q_d)$ whose elements depend on the chosen spline basis. For instance, in the case of B-splines, these penalty matrices can be expressed as $\breve{\boldsymbol{P}}_d = \boldsymbol{D}_d^t \boldsymbol{D}_d$, where $\boldsymbol{D}_d$ is a matrix that forms differences of order q_d $(d = 1, 2)$ (Eilers and Marx 1996).

2.1 Mixed model representation

To estimate model (2) subject to the penalization defined in (3), we adopt here the equivalence between P-splines and generalized linear mixed models (GLMMs) (Lin and Zhang 1999, Currie and Durban 2002, Wand 2003). Under this approach, the design matrix $\boldsymbol{B}$ and the vector of regression coefficients $\boldsymbol{\theta}$ in (2) are reformulated in such a way that

$$g(\boldsymbol{\mu}) = \boldsymbol{\eta} = \boldsymbol{B}\boldsymbol{\theta} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\alpha}, \quad \text{with} \quad \boldsymbol{\alpha} \sim N(\boldsymbol{0}, \boldsymbol{G}), \tag{4}$$

where $\boldsymbol{X}$ and $\boldsymbol{Z}$ are the model matrices, and $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$ are the fixed and random effects coefficients of the generalized linear mixed model respectively. The random effects have covariance matrix $\boldsymbol{G}$, which depends on two variance components τ_1^2 and τ_2^2.

To obtain the mixed model representation (4), we follow the proposal by Lee (2010), Lee and Durbán (2011). Their approach is based on the Singular Value Decomposition (SVD) of the marginal penalties $\breve{\boldsymbol{P}}_d$ involved in (3), for $d = 1, 2$. Let $\breve{\boldsymbol{P}}_d = \boldsymbol{U}_d \boldsymbol{\Sigma}_{\boldsymbol{d}} \boldsymbol{U}_d^t$, where $\boldsymbol{U}_d$ is the matrix of eigenvectors and $\boldsymbol{\Sigma}_d$ is the diagonal matrix of eigenvalues. Let's also denote $\boldsymbol{U}_{ds}$ the sub-matrix of $\boldsymbol{U}_d$ containing the eigenvectors corresponding to the $(c_d - q_d)$ non-zero eigenvalues. Then, the mixed model matrices for model (4) are

$$\begin{aligned} \boldsymbol{X} &= [\boldsymbol{X}_2 \Box \boldsymbol{X}_1], \\ \boldsymbol{Z} &= [\boldsymbol{Z}_2 \Box \boldsymbol{X}_1 | \boldsymbol{X}_2 \Box \boldsymbol{Z}_1 | \boldsymbol{Z}_2 \Box \boldsymbol{Z}_1], \end{aligned} \tag{5}$$

where $\boldsymbol{X}_d = \left[\boldsymbol{1}_n | \boldsymbol{x}_d | \dots | \boldsymbol{x}_d^{(q_d - 1)}\right]$ and $\boldsymbol{Z}_d = \boldsymbol{B}_d \boldsymbol{U}_{ds}$ (for $d = 1, 2$), and the inverse of the random effects covariance matrix $\boldsymbol{G}$ in (4) becomes a block - diagonal matrix

$$\boldsymbol{G}^{-1} = \begin{pmatrix} \frac{1}{\tau_2^2}\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{q_1} & & \\ & \frac{1}{\tau_1^2}\boldsymbol{I}_{q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1 & \\ & & \frac{1}{\tau_2^2}\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1 - q_1} + \frac{1}{\tau_1^2}\boldsymbol{I}_{c_2 - q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1 \end{pmatrix},$$

where $\tilde{\boldsymbol{\Sigma}}_d$ is the sub-matrix of $\boldsymbol{\Sigma}_d$ with the non-zero eigenvalues, $\tau_1^2 = \frac{\phi}{\lambda_1}$ and $\tau_2^2 = \frac{\phi}{\lambda_2}$. As can be observed, under this new configuration, the smoothing parameterλ_d is given by the ratio of the variance components, i.e., $\lambda_d = \frac{\phi}{\tau_d^2}$, $(d = 1, 2)$. Note the relationship between each block of $\boldsymbol{G}^{-1}$ and each block of the random model matrix $\boldsymbol{Z}$ defined in (5). Each variance component τ_d (as well as $\tilde{\boldsymbol{\Sigma}}_d$) *appears* in $\boldsymbol{G}^{-1}$ whenever the $\boldsymbol{Z}_d$ matrix is in the corresponding block of $\boldsymbol{Z}$. This correspondence might be useful to better understand how $\boldsymbol{G}^{-1}$ is constructed in the 3D case, which will be presented in Section 4 (or, by extension, in the d-dimensional case). In relation to this correspondence, the block-structure of $\boldsymbol{Z}$ leads also to a very interesting decomposition of the penalized part of the bidimensional surface f in (1) in three different terms: (a) a term associated with $\boldsymbol{Z_2}\Box\boldsymbol{X_1}$ that contains the smooth *main* effect of x_2 and (q_1-1) varying coefficient terms (Hastie and Tibshirani 1993) with x_2 varying smoothly with x_1 $\left(f_2\left(x_2\right) + \sum_{j=1}^{q_1-1} x_1^j h_{j2}\left(x_2\right)\right)$, (b) a term associated with $\boldsymbol{X_2}\Box\boldsymbol{Z_1}$ with the smooth *main* effect of x_1 and $(q_2 - 1)$ varying coefficient terms with x_2 varying smoothly with x_1 $\left(f_1\left(x_1\right) + \sum_{j=1}^{q_2-1} x_2^j h_{j1}\left(x_1\right)\right)$; and, (c) a *pure smooth* interaction term associated with $\boldsymbol{Z_2}\Box\boldsymbol{Z_1}$ $\left(f_{1|2}\left(x_1, x_2\right)\right)$.

As for the estimation of any GLMM, estimation of model (4) involves two interrelated stages: (a) fixed and random effects coefficients estimation ($\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$); and (b) variance components estimation (τ_1^2, τ_2^2, and, possibly, ϕ). In our context, and for fixed values of the variance components, estimation of the model's fixed and random effects presents no problem. These can be obtained using *Penalized Quasi-likelihood* (PQL) methods (Stiratelli et al. 1984, Schall 1991, Breslow and Clayton 1993). PQL is a very simple method for estimation of GLMMs, and can be easily implemented by iterative fitting a *working* linear mixed model to a *working* dependent variable $\boldsymbol{z}$, on the basis of a Fisher scoring algorithm which involves a weight matrix $\boldsymbol{W}$ that is updated at each iteration (we describe this point in detail in Appendix A). However, estimation of τ_1^2, τ_2^2, and ϕ can not be accommodated using standard procedures for variance components estimation in mixed models (or, more precisely, standard mixed modelling software), since the covariance matrix of the random effects $\boldsymbol{G}$ (see (8) in Section 3) has a non-standard form, with a block involving both the variance components τ_1^2 and τ_2^2. In the following section we present a computational efficient algorithm for estimating variance components. Following Harville (1977) and Schall (1991), we derived closed-form expression for the estimates of the variance components which in turn avoids the need of using numerical optimization methods and thus rendering very good computing times.

3 Variance components estimation: the m-schall algorithm

In this section we present the main result of this paper. Since, on the basis of PQL, estimation of model (4) is implemented by repeated estimation of a *working* linear mixed model (see Appendix A), we focus here on the estimation of the variance components in each of these iterations. Accordingly, and by a slight abuse of terminology, we will refer to the derived expressions for the variance components as ML or REML estimates, although, strictly speaking, it only applies for normally distributed responses with identity link function.

For the sake of illustration, in this part we restrict our attention to the estimation of the variance components based on REML. However, ML estimates can be also easily obtained following the same reasoning that will be used for REML. The corresponding closed-form expressions for ML estimates of the variance components has been added in Appendix C. We have called the proposed algorithm *m-schall* (from multidimensional Schall), as it can be seen as a generalization of the algorithm by Schall (1991) - for variance components estimation - to deal with non-standard structures of the covariance matrix of the random effects.

For ease of readability, we shall use the following notation to denote operations on diagonal matrices: let $\boldsymbol{A}$ and $\boldsymbol{M}$ be diagonal matrices, $\vec{\boldsymbol{A}}$ denotes the vector containing the diagonal elements of $\boldsymbol{A}$, $\boldsymbol{A}^2$ denotes the diagonal matrix, whose diagonal is formed by the element-wise square of $\vec{\boldsymbol{A}}$, $1/\boldsymbol{A}$ denotes the diagonal matrix formed by the element-wise inverses of $\vec{\boldsymbol{A}}$, and $\boldsymbol{A}/\boldsymbol{M}$ denotes the diagonal matrix formed by the element-wise quotient of $\vec{\boldsymbol{A}}$ and $\vec{\boldsymbol{M}}$

Theorem. *In each iteration of the Fisher-Scoring algorithm, REML estimates of the variance components τ_d ($d = 1, 2$) and, when unknown, ϕ are given by*

$$\hat{\tau}_d^2 = \frac{\hat{\boldsymbol{\alpha}}^t \boldsymbol{\Lambda}_d \hat{\boldsymbol{\alpha}}}{ed_d}, \tag{6}$$

$$\hat{\phi} = \frac{\left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}\right)^t \widetilde{\boldsymbol{W}} \left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}\right)}{n - \sum_{d=1}^2 ed_d - rank(\boldsymbol{X})},$$

with

$$ed_d = trace\left(\boldsymbol{Z}^t \boldsymbol{P} \boldsymbol{Z} \boldsymbol{G} \frac{\boldsymbol{\Lambda}_d}{\tau_d^2} \boldsymbol{G}\right),$$

where $\boldsymbol{P} = \boldsymbol{V}^{-1} - \boldsymbol{V^{-1}X}\left(\boldsymbol{X^tV^{-1}X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V^{-1}}$ *with* $\boldsymbol{V} = \boldsymbol{W}^{-1} + \boldsymbol{ZGZ}^t$, $\widetilde{\boldsymbol{W}} = \phi\boldsymbol{W}$, *and*

$$\boldsymbol{\Lambda}_2 = \begin{pmatrix} \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{q_1} & & \\ & \boldsymbol{0}_{q_2(c_1-q_1)} & \\ & & \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1} \end{pmatrix},$$

$$\boldsymbol{\Lambda}_1 = \begin{pmatrix} \boldsymbol{0}_{q_1(c_2-q_2)} & & \\ & \boldsymbol{I}_{q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1 & \\ & & \boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1 \end{pmatrix},$$

where $\boldsymbol{0}_q$ *is a square matrix of zeroes of order equal to* q*.*

Proof. Ignoring the dependence of $\boldsymbol{W}$ on τ_d $(d = 1, 2)$, the approximate restricted log-likelihood of the *working* linear mixed model is given by (Breslow and Clayton 1993)

$$l^* = -\frac{1}{2}\log|\boldsymbol{V}| - \frac{1}{2}\log|\boldsymbol{X^tV^{-1}X}| - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^t\boldsymbol{V}^{-1}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}}).$$

The REML estimates of the variance components are then obtained in the usual manner by maximizing this quantity. Taking derivatives with respect to the variance components τ_d^2 $(d = 1, 2)$, we obtain (see Appendix B for details)

$$\frac{\partial l^*}{\partial \tau_d^2} = -\frac{1}{2} trace\left(\boldsymbol{Z}^t\boldsymbol{PZ}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\right) + \frac{1}{2}\hat{\boldsymbol{\alpha}}^t\boldsymbol{G}^{-1}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\boldsymbol{G}^{-1}\hat{\boldsymbol{\alpha}}. \tag{7}$$

Now, we need to calculate $\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}$. Given that $\boldsymbol{G}^{-1}$ is a diagonal matrix, it follows that $\boldsymbol{G}$ is easily obtained

$$\boldsymbol{G} = \begin{pmatrix} \tau_2^2/\boldsymbol{d}_2 & & \\ & \tau_1^2/\boldsymbol{d}_1 & \\ & & 1/(\boldsymbol{d}_2^*/\tau_2^2 + \boldsymbol{d}_1^*/\tau_1^2) \end{pmatrix}. \tag{8}$$

where $\boldsymbol{d}_1 = \boldsymbol{I}_{q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1$, $\boldsymbol{d}_2 = \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{q_1}$, $\boldsymbol{d}_1^* = \boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1$, and $\boldsymbol{d}_2^* = \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1}$. Thus, $\frac{\partial \boldsymbol{G}}{\partial \tau_2^2}$, can be expressed as

$$\begin{aligned}\frac{\partial \boldsymbol{G}}{\partial \tau_2^2} &= \frac{1}{\tau_2^4}\begin{pmatrix} \frac{\boldsymbol{d}_2}{\left(\frac{\boldsymbol{d}_2}{\tau_2^2}\right)^2} & & \\ & \mathbf{0}_{q_2(c_1-q_1)} & \\ & & \frac{\boldsymbol{d}_2^*}{\left(\frac{\boldsymbol{d}_2^*}{\tau_2^2}+\frac{\boldsymbol{d}_1^*}{\tau_1^2}\right)^2} \end{pmatrix}\\ &= \frac{1}{\tau_2^4}\boldsymbol{G}\begin{pmatrix} \boldsymbol{d}_2 & & \\ & \mathbf{0}_{q_2(c_1-q_1)} & \\ & & \boldsymbol{d}_2^* \end{pmatrix}\boldsymbol{G}\\ &= \frac{1}{\tau_2^4}\boldsymbol{G}\boldsymbol{\Lambda}_2\boldsymbol{G}, \end{aligned} \tag{9}$$

with $\boldsymbol{\Lambda}_2 = \mathrm{diag}(\vec{\boldsymbol{d}}_2, \vec{\mathbf{0}}_{q_2(c_1-q_1)}, \vec{\boldsymbol{d}}_2^*)$, where $\vec{\mathbf{0}}_{q_2(c_1-q_1)}$ is a vector of zeroes of length $q_2(c_1 - q_1)$. Similarly, we obtain the expression for $\frac{\partial \boldsymbol{G}}{\partial \tau_1^2}$ as

$$\frac{\partial \boldsymbol{G}}{\partial \tau_1^2} = \frac{1}{\tau_1^4}\boldsymbol{G}\boldsymbol{\Lambda}_1\boldsymbol{G}. \tag{10}$$

with $\boldsymbol{\Lambda}_1 = \mathrm{diag}(\vec{\mathbf{0}}_{q_1(c_2-q_2)}, \vec{\boldsymbol{d}}_1, \vec{\boldsymbol{d}}_1^*)$. By pluggin expression (9) or (10) in (7) we obtain that the first-order partial derivatives of the approximate restricted log-likelihood become

$$2\frac{\partial l^*}{\partial \tau_d^2} = -\frac{1}{\tau_d^2}trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}\right) + \frac{1}{\tau_d^4}\hat{\boldsymbol{\alpha}}^t\boldsymbol{\Lambda}_d\hat{\boldsymbol{\alpha}}. \tag{11}$$

Then, REML estimates of the variance components τ_d $(d = 1, 2)$ are found by equating expression (11) to zero, which gives

$$\hat{\tau}_d^2 = \frac{\hat{\boldsymbol{\alpha}}^t\boldsymbol{\Lambda}_d\hat{\boldsymbol{\alpha}}}{trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}\right)}.$$

Before proceeding with the estimation of ϕ - if unknown - it is important to observe that the sum of the quantities involved in the denominators of the variance components estimates corresponds to the effective dimension of the penalized part (or random part) of the fitted model

$$
\begin{aligned}
trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_1}{\tau_1^2}\boldsymbol{G}\right) + trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_2}{\tau_2^2}\boldsymbol{G}\right) &= trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\right)\\
&= trace\left(\boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^t\boldsymbol{P}\right)\\
&= trace\left(\boldsymbol{H}_{Random}\right)
\end{aligned}
$$

where $\boldsymbol{H}_{Random}$ denotes the hat matrix (Hastie and Tibshirani 1990) of the random part (see (16)).

Finally, an estimate of ϕ is obtained, as before, by taking derivatives of the approximate restricted log-likelihood with respect to ϕ

$$
\frac{\partial l^*}{\partial \phi} = -\frac{1}{2} trace\left(\boldsymbol{P}\frac{\partial \boldsymbol{V}}{\partial \phi}\right) + \frac{1}{2}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^t\boldsymbol{V}^{-1}\frac{\partial \boldsymbol{V}}{\partial \phi}\boldsymbol{V}^{-1}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}}).
$$

First, by Equation (5.2) in Harville (1977), we have that $\boldsymbol{V}^{-1}(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}) = \boldsymbol{W}(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{Z}\hat{\boldsymbol{\alpha}})$. Moreover, given that $\boldsymbol{V}$ depends on ϕ through $\boldsymbol{W}^{-1}$ which can be rewritten as $\boldsymbol{W} = \frac{1}{\phi}\widetilde{\boldsymbol{W}}$, with $\widetilde{\boldsymbol{W}}$ being a diagonal matrix with elements $\widetilde{w}_{ii} = \left\{g'\left(\mu_i\right)^2\nu\left(\mu_i\right)\right\}^{-1}$, and ignoring again the dependence of $\widetilde{\boldsymbol{W}}$ on ϕ, it then follows that

$$
2\frac{\partial l^*}{\partial \phi} = -\frac{1}{\phi} trace\left(\boldsymbol{P}\boldsymbol{W}^{-1}\right) + \frac{1}{\phi^2}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}})^t\widetilde{\boldsymbol{W}}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}).
$$

By equating the above expression to zero, we obtain

$$
\hat{\phi} = \frac{(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}})^t\widetilde{\boldsymbol{W}}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}})}{trace\left(\boldsymbol{P}\boldsymbol{W}^{-1}\right)},
$$

where (see Equation 5.3 in Harville 1977 and expressions (15), (16), and (17))

$$
\begin{aligned}
trace\left(\boldsymbol{P}\boldsymbol{W}^{-1}\right) &= trace\left(\boldsymbol{W}^{-1}\boldsymbol{P}\right)\\
&= trace\left(\boldsymbol{I}_n - [\boldsymbol{X}|\boldsymbol{Z}\boldsymbol{G}]\boldsymbol{C}^{-1}\begin{bmatrix}\boldsymbol{X}^t\boldsymbol{W}\\ \boldsymbol{Z}^t\boldsymbol{W}\end{bmatrix}\right)\\
&= trace\left(\boldsymbol{I}_n - [\boldsymbol{X}|\boldsymbol{Z}\boldsymbol{G}]\begin{bmatrix}\left(\boldsymbol{X}\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}\boldsymbol{V}^{-1}\\ \boldsymbol{Z}^t\boldsymbol{P}\end{bmatrix}\right)\\
&= n - trace\left(\boldsymbol{X}\left(\boldsymbol{X}\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}\boldsymbol{V}^{-1}\right) - trace\left(\boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^t\boldsymbol{P}\right)\\
&= n - rank\left(\boldsymbol{X}\right) - \sum_{d=1}^{2}\mathrm{ed}_d.
\end{aligned}
$$

Note that $\boldsymbol{H} = [\boldsymbol{X}|\boldsymbol{Z}\boldsymbol{G}]\boldsymbol{C}^{-1}[\boldsymbol{X}|\boldsymbol{Z}]^t\boldsymbol{W}$ corresponds with the hat matrix of the fitted model, whose trace, as shown, can be decomposed as the sum of the traces of the hat matrices of the unpenalized (or fixed) part and the penalized (or random) part. □

As shown in the proof of the theorem, $\mathrm{ed}_1 + \mathrm{ed}_2$ corresponds to the effective dimension of the penalized part of the fitted model. This effective dimension (plus the dimension of the unpenalized part), can be interpreted, as usual, as a measure of the smoothness of the fitted interaction surface. It would be nevertheless interesting to elucidate the interpretation of ed_d in this context. One could be tempted to interpret these quantities as a measure of the smoothness in the corresponding covariate (as e.g in the additive case or, in the multidimensional setting, in the P-spline ANOVA proposed by Lee et al. (2013)). However, a detailed evaluation on how these values are computed brings a completely different, and maybe surprising, result. The *computation* of the trace of $\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}$, given that $\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}$ is a diagonal matrix (since $\boldsymbol{\Lambda}_d$ and $\boldsymbol{G}$ are), can be obtained as

$$
trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}\right) = \sum_{j=1}^{(c_1-q_1)(c_2-q_2)}\gamma_j\varphi_j^d, \qquad (12)
$$

where γ_j is the jth element of the diagonal of $\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}$ and φ_j^d is the jth element of the diagonal of $\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}$. Given that the trace of $\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}$ corresponds to the trace of the hat matrix of the penalized part (see proof of the theorem), expression (12) can be interpreted as a decomposition of the *effective dimension* of the fitted model into components related to each covariate x_d according to the values of φ_j^d. Taking a look at the $\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}$ matrix, we have

$$\frac{\boldsymbol{\Lambda}_2}{\tau_2^2}\boldsymbol{G} = \begin{pmatrix} \boldsymbol{I}_{q_1(c_2-q_2)} & & \\ & \boldsymbol{0}_{q_2(c_1-q_1)} & \\ & & \frac{\frac{1}{\tau_2^2}\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1}}{\frac{1}{\tau_2^2}\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1} + \frac{1}{\tau_1^2}\boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1} \end{pmatrix},$$

and,

$$\frac{\boldsymbol{\Lambda}_1}{\tau_1^2}\boldsymbol{G} = \begin{pmatrix} \boldsymbol{0}_{q_1(c_2-q_2)} & & \\ & \boldsymbol{I}_{q_2(c_1-q_1)} & \\ & & \frac{\frac{1}{\tau_1^2}\boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1}{\frac{1}{\tau_2^2}\boldsymbol{\Sigma}_2 \otimes \boldsymbol{I}_{c_1-q_1} + \frac{1}{\tau_1^2}\boldsymbol{I}_{c_2-q_2} \otimes \boldsymbol{\Sigma}_1} \end{pmatrix}.$$

As a result, the first $q_1(c_2 - q_2)$ elements of the diagonal of $\boldsymbol{Z}^t\boldsymbol{PZG}$ are allocated to covariate x_2, the following $q_2(c_1 - q_1)$ to x_1, and the last $(c_1 - q_1)(c_2 - q_2)$ elements are allocated among x_1 and x_2 according to weights φ_j^d that are inversily proportional to the variance component associated with the corresponding covariate. However, an alternative interpretation can be provided by expressing these weights as

$$\frac{\tau_1^2\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1}}{\tau_1^2\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1} + \tau_2^2\boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1} \quad \text{and} \quad \frac{\tau_2^2\boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1}{\tau_1^2\tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_1-q_1} + \tau_2^2\boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_1}.$$

It follows that the last $(c_1-q_1)(c_2-q_2)$ elements of the diagonal of $\boldsymbol{Z}^t\boldsymbol{PZG}$ are allocated to x_1 according to weights that are *directly* proportional to the variance component associated with x_2 (and the same holds for x_2).

Correspondingly, and taking in mind the three-term decomposition of the bidimensional surface f in (1) explained in Section 2.1, each ed_d can be obtained as the sum of two components, that could be interpreted as follows: one that gathers the amount of smoothing along x_d (a sort of *within* smoothness), and the other one that gathers how much the smooth effect of x_d varies along the other covariate (*between* smoothness).

3.1 Estimation algorithm

In this section we summarize the algorithm for the estimation of model (4):

Initialize. Set initial values for model's fixed and random effects and variance components. For instance, $\hat{\beta}_k^{(0)} = \hat{\alpha}_l^{(0)} = 0$ $(1 \leq k \leq q_1q_2,\ 1 \leq l \leq (c_1 - q_1)(c_2 - q_2))$ and $\hat{\tau}_1^{2(0)}$

$= \hat{\tau}_2^{2(0)}$. In those situations where ϕ is unknown, establish an initial value for this parameter, e.g. $\hat{\phi}^{(0)} = 1$. Set $k = 0$

Step 1. Given the initial *estimates* of model's fixed and random effects, construct the *working* response variable $\boldsymbol{z}$ and the matrix of weights $\boldsymbol{W}$ as follows

$$z_i = g(\mu_i^{(k)}) + (y_i - \mu_i^{(k)})g'(\mu_i^{(k)}),$$

$$w_{ii} = \left\{\hat{\phi}^{(k)} g'(\mu_i^{(k)})^2 \nu(\mu_i^{(k)})\right\}^{-1},$$

with $\boldsymbol{\mu}^{(k)} = g^{-1}\left(\boldsymbol{X}\hat{\boldsymbol{\beta}}^{(k)} + \boldsymbol{Z}\hat{\boldsymbol{\alpha}}^{(k)}\right)$.

Step 1.1. Given the initial *estimates* of variance components, estimate $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$ by solving the linear system (17). Let $\hat{\boldsymbol{\alpha}}$ and $\hat{\boldsymbol{\beta}}$ be these estimates.

Step 1.2. Estimate the variance components as

$$\hat{\tau}_d^2 = \frac{\hat{\boldsymbol{\alpha}}^t \boldsymbol{\Lambda}_d \hat{\boldsymbol{\alpha}}}{ed_d^{(k)}},$$

and, when necessary,

$$\hat{\phi} = \frac{\left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}\right)^t \widetilde{\boldsymbol{W}} \left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}\right)}{n - \sum_{d=1}^{2} \mathrm{ed}_d^{(k)} - p},$$

with

$$ed_d^{(k)} = trace\left(\boldsymbol{Z}^t \boldsymbol{P}^{(k)} \boldsymbol{Z} \boldsymbol{G}^{(k)} \frac{\boldsymbol{\Lambda}_d}{\hat{\tau}_d^{2(k)}} \boldsymbol{G}^{(\mathbf{0})}\right),$$

where $\boldsymbol{P}^{(k)}$ and $\boldsymbol{G}^{(k)}$ denote the corresponding $\boldsymbol{P}$ and $\boldsymbol{G}$ matrices obtained on the basis of the *initial* estimates.

Step 1.3. Repeat Step 1.1 and Step 1.2 with $\hat{\tau}_1^{2(k)}$, $\hat{\tau}_2^{2(k)}$, and, if updated, $\hat{\phi}^{(k)}$ being replaced by $\hat{\tau}_1^2$, $\hat{\tau}_2^2$, and $\hat{\phi}$ respectively, until the convergence criterion

$$\frac{|\hat{\phi} - \hat{\phi}^{(k)}| + \sum_{d=1}^{2} |\hat{\tau}_d^2 - \hat{\tau}_d^{2(k)}|}{3} \leq \varsigma,$$

where ς is a small threshold (the tolerance for the convergence criterion), e.g, 1×10^{-6}.

Step 3. Repeat Step 1. with the model's fixed and random effects and variance components being replaced by those obtained in the last iteration of Steps 1.1 - Step 1.3, until the convergence criterion

$$\frac{\|\boldsymbol{\eta}^{(k+1)} - \boldsymbol{\eta}^{(k)}\|^2}{\|\boldsymbol{\eta}^{(k+1)}\|^2} \leq \upsilon,$$

where υ is a small threshold.

3.2 Computational aspects

We present here some computational aspects that can be applied for the fast implementation of the estimation algorithm presented in Section 3.1. Specifically, we focus on the computation of variance components (Step 1.2). Nevertheless, it should be also noted that, when the data is in an array structure, the generalized linear array model (GLAM) by Currie et al (2006) can be used for the construction of the model matrices involved in the linear system (17), thus improving the speed of the estimation algorithm.

The estimation of the variance components by using the expression given in (6) requires the computation of the trace of $\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}$, which involves the computation and manipulation of several $n \times n$ matrices. As pointed out before, this computation can be relaxed by taking into account that both, $\boldsymbol{G}$ and $\boldsymbol{\Lambda}_d$ are diagonal matrices, and, therefore, $\boldsymbol{G}\boldsymbol{\Lambda}_d\boldsymbol{G}$ is also a diagonal matrix. Then

- $\boldsymbol{G}\boldsymbol{\Lambda}_d\boldsymbol{G} = \mathrm{diag}(\vec{G} * \vec{\boldsymbol{\Lambda}}_d * \vec{G})$, with $*$ denoting the element-wise vector product.
- Computation of the former trace only requires the computation of the diagonal of $\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}$.

Moreover, by expression (5.3) in Harville (1977) we have

$$\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z} = \left[\boldsymbol{0}_{q_1q_2}|\boldsymbol{I}_{(c_1-q_1)(c_2-q_2)}\right]\boldsymbol{C}^{-1}\left[\boldsymbol{X}|\boldsymbol{Z}\right]^t\boldsymbol{W}\boldsymbol{Z},$$

with $\boldsymbol{C}^{-1}$ being the inverse of matrix $\boldsymbol{C}$ defined in (17). Correspondingly, the computation of its diagonal can be carried out by the column-wise addition of

$$\left(\left[\boldsymbol{0}_{q_1q_2}|\boldsymbol{I}_{(c_1-q_1)(c_2-q_2)}\right]\boldsymbol{C}^{-1}\right)^t \odot \begin{bmatrix}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{Z}\\ \boldsymbol{Z}^t\boldsymbol{W}\boldsymbol{Z},\end{bmatrix} \tag{13}$$

where $\odot$ denotes the Hadamard or element-wise matrix product. For ease of notation, let's denote $\boldsymbol{\zeta}^t$ this diagonal vector, and $\boldsymbol{\xi}^{dt} = \vec{G} * \vec{\boldsymbol{\Lambda}}_d * \vec{G}$. Then, it follows that:

$$trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}\right) = \frac{1}{\tau_d^2}\sum_{j=1}^{(c_1-q_1)(c_2-q_2)} \zeta_j\xi_j^d.$$

Note that no new matrices have to be computed to evaluate expression (13), since all of them have been already computed for the estimation of $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}$.

4 Some extensions

In this section we present some extensions of the m-schall algorithm presented in Section 3. As will be observed, the key point when it came to extending the m-schall algorithm is to determine the variance-covariance matrix $\boldsymbol{G}$ of the random effects as well as its derivatives with respect to the variance components. Specifically, the only requirement will be to specify the form of the matrix $\boldsymbol{\Lambda}$ involved in the expression of the estimate of each variance component (see (6)). This feature makes, for instance, straightforward the extension of the m-schall algorithm to deal with the ANOVA-type decomposition presented in Lee and Durbán (2011). We therefore focus here on presenting more complex extensions. We first present the generalization of the m-schall algorimth to the three dimensional case, and then we show how the algorithm can be also incorporated into the estimation of a GAMM (Lin and Zhang 1999) with sets of i.i.d Gaussian random effects.

4.1 Extension to the three dimensional case

Consider a three-dimensional generalized regression problem

$$g\left(E[y_i|\boldsymbol{x}_i]\right) = g\left(\mu_i\right) = \eta_i = f\left(x_{i1}, x_{i2}, x_{i3}\right),$$

where f is a smooth and unknown function. As for the bidimensional case, we model function f by tensor product of spline basis functions and we assume an anisotropic penalization

$$\breve{\boldsymbol{P}} = \lambda_1\breve{\boldsymbol{P}}_1 \otimes \boldsymbol{I}_{c_2} \otimes \boldsymbol{I}_{c_3} + \lambda_2\boldsymbol{I}_{c_1} \otimes \breve{\boldsymbol{P}}_2 \otimes \boldsymbol{I}_{c_3} + \lambda_3\boldsymbol{I}_{c_1} \otimes \boldsymbol{I}_{c_2} \otimes \breve{\boldsymbol{P}}_3.$$

Following the same procedure as in Section 2.1 for the bidimensional case (see Lee 2010, Lee and Durbán 2011for further details), we obtain the mixed model model matrices

$$\begin{aligned}\boldsymbol{X} =&[\boldsymbol{X}_1\Box\boldsymbol{X}_2\Box\boldsymbol{X}_3]\\ \boldsymbol{Z} =&[\boldsymbol{Z}_1\Box\boldsymbol{X}_2\Box\boldsymbol{X}_3|\boldsymbol{X}_1\Box\boldsymbol{Z}_2\Box\boldsymbol{X}_3|\boldsymbol{X}_1\Box\boldsymbol{X}_2\Box\boldsymbol{Z}_3|\boldsymbol{Z}_1\Box\boldsymbol{Z}_2\Box\boldsymbol{X}_3|\boldsymbol{Z}_1\Box\boldsymbol{X}_2\Box\boldsymbol{Z}_3|\\ &\boldsymbol{X}_1\Box\boldsymbol{Z}_2\Box\boldsymbol{Z}_3|\boldsymbol{Z}_1\Box\boldsymbol{Z}_2\Box\boldsymbol{Z}_3],\end{aligned}$$

and the inverse of the random effects covariance matrix $\boldsymbol{G}$

$$\boldsymbol{G}^{-1} = \begin{pmatrix} \frac{\boldsymbol{d}_{1u}}{\tau_1^2} &&&&&& \\ & \frac{\boldsymbol{d}_{2u}}{\tau_2^2} &&&&& \\ && \frac{\boldsymbol{d}_{3u}}{\tau_3^2} &&&& \\ &&& \frac{\boldsymbol{d}_{11}}{\tau_1^2}+\frac{\boldsymbol{d}_{21}}{\tau_2^2} &&& \\ &&&& \frac{\boldsymbol{d}_{12}}{\tau_1^2}+\frac{\boldsymbol{d}_{31}}{\tau_3^2} && \\ &&&&& \frac{\boldsymbol{d}_{22}}{\tau_2^2}+\frac{\boldsymbol{d}_{32}}{\tau_3^2} & \\ &&&&&& \frac{\boldsymbol{d}_{1t}}{\tau_1^2}+\frac{\boldsymbol{d}_{2t}}{\tau_2^2}+\frac{\boldsymbol{d}_{3t}}{\tau_3^2} \end{pmatrix},$$

where $\boldsymbol{d}_{1u} = \tilde{\boldsymbol{\Sigma}}_1 \otimes \boldsymbol{I}_{q_2} \otimes \boldsymbol{I}_{q_3}$, $\boldsymbol{d}_{2u} = \boldsymbol{I}_{q_1} \otimes \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{q_3}$, $\boldsymbol{d}_{3u} = \boldsymbol{I}_{q_1} \otimes \boldsymbol{I}_{q_2} \otimes \tilde{\boldsymbol{\Sigma}}_3$, $\boldsymbol{d}_{11} = \tilde{\boldsymbol{\Sigma}}_1 \otimes \boldsymbol{I}_{c_2-q_2} \otimes \boldsymbol{I}_{q_3}$, $\boldsymbol{d}_{12} = \tilde{\boldsymbol{\Sigma}}_1 \otimes \boldsymbol{I}_{q_2} \otimes \boldsymbol{I}_{c_3-q_3}$, $\boldsymbol{d}_{21} = \boldsymbol{I}_{c_1-q_1} \otimes \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{q_3}$, $\boldsymbol{d}_{22} = \boldsymbol{I}_{q_1} \otimes \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_3-q_3}$, $\boldsymbol{d}_{31} = \boldsymbol{I}_{c_1-q_1} \otimes \boldsymbol{I}_{q_2} \otimes \tilde{\boldsymbol{\Sigma}}_3$, $\boldsymbol{d}_{32} = \boldsymbol{I}_{q_1} \otimes \boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_3$, $\boldsymbol{d}_{1t} = \tilde{\boldsymbol{\Sigma}}_1 \otimes \boldsymbol{I}_{c_2-q_2} \otimes \boldsymbol{I}_{c_3-q_3}$, $\boldsymbol{d}_{2t} = \boldsymbol{I}_{c_1-q_1} \otimes \tilde{\boldsymbol{\Sigma}}_2 \otimes \boldsymbol{I}_{c_3-q_3}$, $\boldsymbol{d}_{3t} = \boldsymbol{I}_{c_1-q_1} \otimes \boldsymbol{I}_{c_2-q_2} \otimes \tilde{\boldsymbol{\Sigma}}_3$. As shown in Section 3, the covariance matrix $\boldsymbol{G}$ and its derivatives with respect to the variance components τ_d^2 $(d = 1, 2, 3)$ can be easily obtained

$$\frac{\partial \boldsymbol{G}}{\partial \tau_d^2} = \frac{1}{\tau_d^2}\boldsymbol{G}\boldsymbol{\Lambda}_d\boldsymbol{G},$$

with

$$\begin{aligned}\boldsymbol{\Lambda}_1 &= \operatorname{diag}(\vec{\boldsymbol{d}}_{1u}, \vec{\boldsymbol{0}}_{q_1q_3(c_2-q_2)}, \vec{\boldsymbol{0}}_{q_1q_2(c_3-q_3)}, \vec{\boldsymbol{d}}_{11}, \vec{\boldsymbol{d}}_{12}, \vec{\boldsymbol{0}}_{q_1(c_2-q_2)(c_3-q_3)}, \vec{\boldsymbol{d}}_{1t}),\\ \boldsymbol{\Lambda}_2 &= \operatorname{diag}(\vec{\boldsymbol{0}}_{q_2q_3(c_1-q_1)}, \vec{\boldsymbol{d}}_{2u}, \vec{\boldsymbol{0}}_{q_1q_2(c_3-q_3)}, \vec{\boldsymbol{d}}_{21}, \vec{\boldsymbol{0}}_{q_2(c_1-q_1)(c_3-q_3)}, \vec{\boldsymbol{d}}_{22}, \vec{\boldsymbol{d}}_{2t}),\\ \boldsymbol{\Lambda}_3 &= \operatorname{diag}(\vec{\boldsymbol{0}}_{q_2q_3(c_1-q_1)}, \vec{\boldsymbol{0}}_{q_1q_2(c_2-q_2)}, \vec{\boldsymbol{d}}_{3u}, \vec{\boldsymbol{0}}_{q_3(c_1-q_1)(c_2-q_2)}, \vec{\boldsymbol{d}}_{31}, \vec{\boldsymbol{d}}_{32}, \vec{\boldsymbol{d}}_{3t}).\end{aligned}$$

Finally, the estimates of the variance components are obtained according to expression (6).

4.2 Extension to Generalized Additive Mixed Models

Consider the generalized additive mixed model

$$g\left(E[y_i|\mathbf{x}_i, \boldsymbol{u}]\right) = g\left(\mu_i\right) = \eta_i = f_{(1,2)}\left(x_{i1}, x_{i2}\right) + \sum_{p=3}^{P} f_p\left(x_{ip}\right) + \boldsymbol{U}_{i1}^t \boldsymbol{u}_1 + \ldots + \boldsymbol{U}_{ic}^t \boldsymbol{u}_c, \quad (14)$$

where $f_{(1,2)}$ and f_p $(p = 3, \ldots, P)$ are smooth functions, $\boldsymbol{u}_j$ are $k_j \times 1$ vectors of random effects, such that $\boldsymbol{u} = \left(\boldsymbol{u}_1^t, \ldots, \boldsymbol{u}_c^t\right)^t \sim N\left(0, \boldsymbol{\Omega}\right)$, where $\boldsymbol{\Omega} = \boldsymbol{diag}\left(\sigma_1^2 \mathbf{1}_{k_1}, \ldots, \sigma_c^2 \mathbf{1}_{k_c}\right)$, and $\boldsymbol{U}_{ij}$ are known vectors of *covariates* associated with the random effects.

To estimate model (14), each f_p $(p = 3, \ldots, P)$ is approximated by a low-rank spline basis (with penalty matrix $\lambda_p \breve{\boldsymbol{P}}_p$), and, $f_{1|2}$, as shown in Section 2, by the tensor product of two univariate spline basis and anisotropic penalty. Moreover, we also adopt here the equivalence between (14) and a GLMM. On the basis of the SVD of the penalty matrices $\breve{\boldsymbol{P}}_j$ $(j = 1, \ldots, P)$, we obtain the mixed model model matrices

$$\begin{aligned} \boldsymbol{X} =& [\boldsymbol{X}_2 \Box \boldsymbol{X}_1 | \tilde{\boldsymbol{X}}_3 | \ldots | \tilde{\boldsymbol{X}}_P] \\ \boldsymbol{Z} =& [\boldsymbol{Z}_2 \Box \boldsymbol{X}_1 | \boldsymbol{X}_2 \Box \boldsymbol{Z}_1 | \boldsymbol{Z}_2 \Box \boldsymbol{Z}_1 | \, \boldsymbol{Z}_3 | \ldots | \boldsymbol{Z}_P | \boldsymbol{U}_1 | \ldots | \boldsymbol{U}_c], \end{aligned}$$

with $\boldsymbol{X}_l$ and $\boldsymbol{Z}_l$ $(l = 1, \ldots, P)$ as defined in Section 2.1 and $\tilde{\boldsymbol{X}}_p = \left[\boldsymbol{x}_p | \ldots | \boldsymbol{x}_p^{(q_p - 1)}\right]$, where the vector of ones has been removed from $\boldsymbol{X}_p$ to ensure identifiability $(p = 3, \ldots, P)$. Finally, $\boldsymbol{U}_j$ $(j = 1, \ldots, c)$ are the random effect matrices associated with the *proper* random effects $\boldsymbol{u}_j$. It is straightforward to show that the covariance matrix $\boldsymbol{G}$ of the random effects $\left(\boldsymbol{\alpha}^t, \boldsymbol{u}^t\right)^t$ becomes

$$\boldsymbol{G} = \begin{pmatrix} \widetilde{\boldsymbol{G}} & & & & \\ & \frac{\tau_3^2}{\boldsymbol{\Sigma}_3} & & & \\ & & \ddots & & \\ & & & \frac{\tau_P^2}{\boldsymbol{\Sigma}_P} & \\ & & & & \boldsymbol{\Omega} \end{pmatrix},$$

with $\widetilde{\boldsymbol{G}}$ being defined in (8), and τ_p^2 being the variance components associated to the smooth function f_p $(p = 3, \ldots, P)$.

Closed-formed expression for the estimates of variance components τ_l $(l = 1, \ldots, P)$ and σ_j^2 $(j = 1, \ldots, c)$ based on REML/ML can be then obtained using the same procedure as presented in Section 3 and Appendix C. As pointed out before, we just need to calculate the Λ matrix involved in the derivative of $\boldsymbol{G}$ with respect to each variance component. In the case of τ_1^2 and τ_2^2, these matrices are equivalent to those defined in (9) or (10), but with a sub-matrix of zeroes corresponding to those blocks of matrix $\boldsymbol{G}$ where τ_p^2 and σ_j^2 appear. Moreover, for each τ_p^2, it is easy to show that

$$\boldsymbol{\Lambda}_p = diag(\vec{\mathbf{0}}_{(c_1 c_2 - q_1 q_2)}, \vec{\mathbf{0}}_{(c_3 - q_3)}, \ldots, \vec{\vec{\tilde{\boldsymbol{\Sigma}}}}_p, \ldots, \vec{\mathbf{0}}_{(c_P - q_P)}, \vec{\mathbf{0}}_K), \quad \text{with} \quad K = \sum_{j=1}^{c} k_j,$$

and, as far as the variance components σ_j^2 $(j = 1, \ldots, c)$ is concerned, we obtain

$$\boldsymbol{\Lambda}_j = diag(\vec{\mathbf{0}}_{(c_1 c_2 - q_1 q_2)}, \vec{\mathbf{0}}_{(c_3 - q_3)}, \ldots, \vec{\mathbf{0}}_{(c_P - q_P)}, \vec{\mathbf{0}}_{k_1}, \ldots, \mathbf{1}_{k_j}, \ldots, \vec{\mathbf{0}}_{k_c}).$$

5 Simulation Study

This section reports the results of a simulation study conducted to study the empirical performance of the estimation procedure described in Section 3 above. Specifically, the aims of this study were twofold: (a) to evaluate the practical behaviour on the basis of the Mean Square Error (MSE); and (b) to study the achievement in terms of the computing time.

For these purposes, we compared the m-schall algorithm with the method given in Wood (2011). In that paper, the author presents a fast and stable approach to the estimation of the smoothing parameters of a GAM based on ML or REML. That approach outperforms - in terms of MSE, convergence failures, and computational cost - previous approaches in this context (see Wood 2011for further details), and therefore it has been chosen as the benchmark method for our simulations. Moreover, the method is implemented in the `gam()` function of the `R`-package `mgcv` (version 1.7-22) (Wood 2006a). The `mgcv` package has become, in recent years, the reference `R`-package for the estimation of GAMs, due to its versatility, easy-to-use interface and good and stable performance. Note that the `R`-package `mgcv` also includes a funcion `bam()` specially designed to deal with very large datasets, which in turn can be much faster than `gam()`. We are aware that the evaluation of the proposed algorithm as far as the computing time is concerned would be more accurate and fair with respect the `bam()` function. However, preliminary simulation studies have revealed that, in some circumstances, this function presents severe problems of convergence, thus rendering computing times of about 30 minutes for small sample sizes. Moreover, for moderate sample sizes (as those used in this study) the computing time can be even larger than with the use of `gam()`. For all these reasons, in this simulation study we have restricted the comparisons of our approach to the `gam()` function.

5.1 Scenarios and Setup

In the first study, 200 values of covariates x_1 and x_2 were simulated independently from a uniform distribution on the interval $[0, 1]$, and the following scenario was considered:

$$\eta = f\left(x_1, x_2\right) = \cos\left(2\pi\sqrt{\left(x_1 - 0.5\right)^2 + \left(x_2 - 0.5\right)^2}\right).$$

Note that this scenario was also used in Lee et al. (2013). The response data y was then generated under two different distributions:

1. $y = \eta + \varepsilon$, where $\varepsilon \sim N\left(0, \sigma^2\right)$ with $\sigma \in \{0.1; 0.5; 1\}$.
2. $y \sim \mathit{Bernoulli}\left(p\right)$, with $p = \exp\left(\tilde{\eta}\right)/\exp\left(1 + \tilde{\eta}\right)$, where $\tilde{\eta} = \left(\eta + 0.2\right)/0.5$,

where the scaling factors that appear in the Bernoulli case were used to control the signal-to-noise ratio. For each marginal, 14-dimensional basis were chosen, and $R = 500$ replicates were performed.

On the basis of the previous scenario, we then evaluated the impact of increasing the sample size, and as a consecuence the basis dimension, on the computing time. Here, the simulations were done assuming a sample size of 1000, and only $\sigma = 0.5$ was considered for the Gaussian case. $R = 100$ replicates were perfomed, and 30-dimensional marginals were chosen.

Finally, we also undertook a small simuation study with three covariates. Five hundred values of covariates x_1, x_2, and x_3 were simulated independently from a uniform distribution on the interval $[0, 1]$, and the response was generated from (see also Wood 2006b)

$$\begin{aligned} y =& 1.5\exp\left(-\frac{\left(x_1 - 0.2\right)^2}{5} - \frac{\left(x_2 - 0.5\right)^2}{3} - \frac{\left(x_3 - 0.9\right)^2}{4}\right) \\ &+ 0.5\exp\left(-\frac{\left(x_1 - 0.3\right)^2}{4} - \frac{\left(x_2 - 0.7\right)^2}{2} - \frac{\left(x_3 - 0.4\right)^2}{6}\right) \\ &+ \exp\left(-\frac{\left(x_1 - 0.1\right)^2}{5} - \frac{\left(x_2 - 0.3\right)^2}{5} - \frac{\left(x_3 - 0.7\right)^2}{4}\right) + \varepsilon, \end{aligned}$$

where $\varepsilon \sim N\left(0, \sigma^2\right)$. As for the first study, different levels of noise were considered ($\sigma \in \{0.1; 0.5; 1\}$), $R = 500$ replicates were performed, but only 7-dimensional marginals were used, yieding a basis dimension of 343.

For both, the m-schall algorithm and the `gam()` function, cubic B-splines basis functions with second order difference penalty ($q = 2$) were chosen to obtain the marginal model matrices, and REML criterion was used for the estimation of the variance components. For the `gam()` function, the tensor product of marginal bases (function `te()`) was used and anisotropy was assumed. With respect to the numerical options for the fitting process, for the m-schall algorithm, the tolerance for the convergence criterion of the variance components and the Fisher's scoring algorithm was set to 1×10^{-6}, and the starting values of the variance components and the fixed and random effects were set to 1 and 0 respectively. As far as `gam()` function is concerned, the numerical options were those established by default. The evaluation of the practical performance of both approaches was judged on the basis of the MSE, computed at the observed covariate values. For Gaussian data, the true linear predictor was chosen as the target. However, in the case of binary data, the MSE was computed on the response scale (the probability). Finally, with regard to the evaluation of the computing time, for the m-schall algorithm the times reported include the computing time needed for (a) the construction of the matrices involved in the algorithm; and (b) the algorithm itself. All the computations were done in a 2.40GHz Intel Core i5 processor computer with 4GB of RAM.

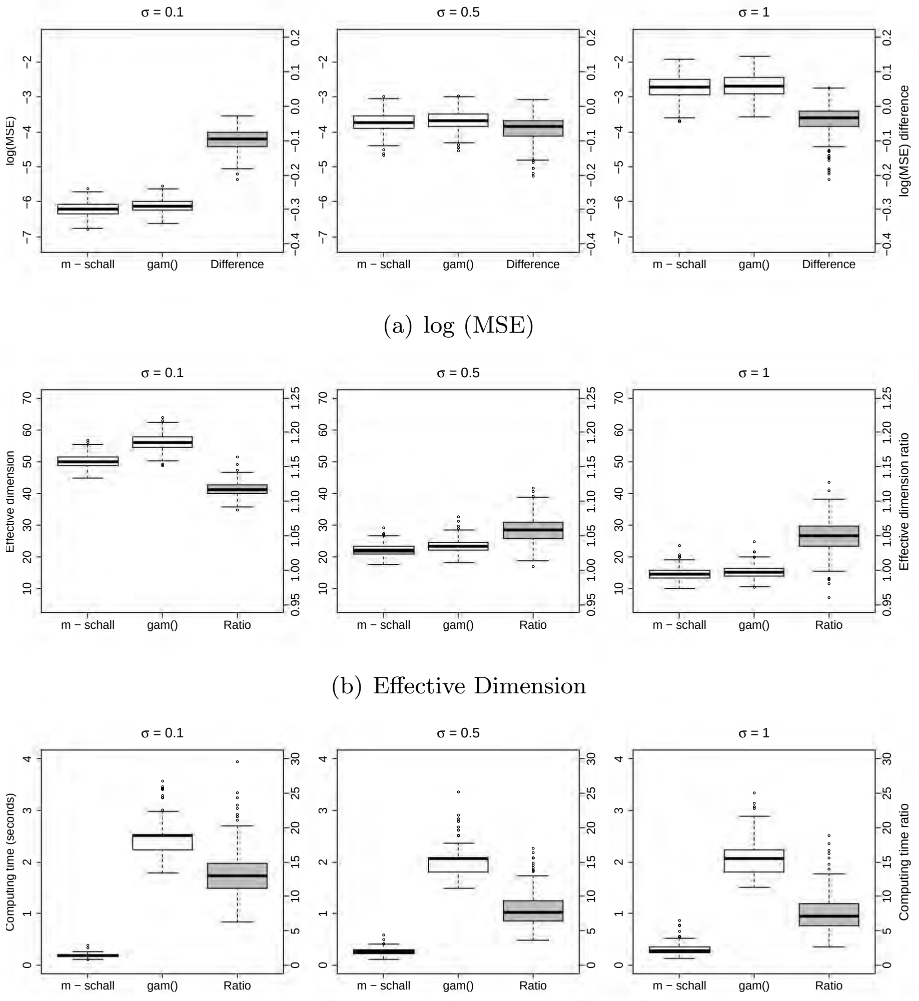
5.2 Results

Figure 1(a) shows the results in terms of the MSE for the two dimensional case, the Gaussian distribution and a sample size of $n = 200$. The figure shows the log(MSE) of both approaches (left y-axis) as well as the difference between the log(MSE) of the m-schall algorithm and the method by Wood (2011) (right y-axis). Thus, in this latter case, values lower than zero indicate a better behaviour of the new proposal. As can be observed, the m-schall algorithm gave better performance in all cases. However, the differences between both approaches diminish as the signal-to-noise ratio increases. It should be noted that this behaviour has been also observed in Wood et al. (2013), when comparing the proposal presented in that paper and the method given in Wood (2006b). Figure 1(b) depicts the behaviour of both approaches, as far as the effective dimension is concerned. For ease of interpretation, we have also incorporated into this figure the ratio of the effective dimension of the Wood (2011)'s method to the m-schall algorithm (right y-axis). As can be observed, the m-schall algorithm provides, in general, a lower effective dimension than the method given in Wood (2011), although again, these differences diminish as the signal-to-noise ratio increases. On the basis of both MSE and effective dimension results, we hypothesize that for small signal-to-noise ratio, the method given in Wood (2011) tends to undersmooth when compared to our approach. In Figure 1(c), the results with respect to the computing time of both approaches are presented, as well as their ratio. Again, for the m-schall algorithm the computing times are influenced by the signal-to-noise ratio. The larger the signal-to-noise ratio, the slower the convergence of the algorithm. For instance, the median (range) of the number of iterarions was 12 (7, 23), 17 (8, 41), and 19 (9, 61), for $\sigma = 0.1$, $\sigma = 0.5$, and for $\sigma = 1$ respectively. Despite this fact, our proposal outperfoms Wood (2011)'s method, requiring, in median, between 13.0 (for $\sigma = 0.1$) and 7.12 (for $\sigma = 1$) times less of the computing time.

The results for the two dimensional case, the Bernoulli distribution and $n = 200$ are shown in Figure 2. Again, our method outperforms Wood (2011)'s method in terms of both, the MSE and the computing time. However, the differences in this case are not as marked as for the Gaussian case. For instance, the required computing times of our algorithm was, in median, 3.19 times less than with Wood (2011)'s method. Once more, the effective dimension of the m-schall algorithm was lower than the effective dimension obtained with the `gam()` function. Finally, as regards the number of PQL iterations needed to reach convergence, the median (range) was 5 (4, 7).

Figure 3 depicts the results for the two dimensional case and $n = 1000$, for both the MSE and the computing time respectively. As pointed out before, for the Gaussian case only $\sigma = 0.5$ was considered. As far as the computing time is concerned, as can be observed when comparing these results with those presented in Figures 1(c) and 2(c) for $n = 200$, the behaviour of m-schall algorithm with respect to the Wood (2011)'s method improves as the sample size, and therefore the basis dimension, increases. In this case, our method needed 10.55 and 4.00 times less than the method by Wood (2011), for the Gaussian and Bernoulli distribution respectively.

Finally, the results for the three dimensional case are depicted in Figure 4. The same pattern as in the previous studies is displayed (results for the effective dimension not shown). As regards the computing time, m-schall algorithm required, in median, 10, 4.66 and 3.65 (for $\sigma = 0.1$, $\sigma = 0.5$ and $\sigma = 1$ respectively) times less than the method given in Wood (2011).

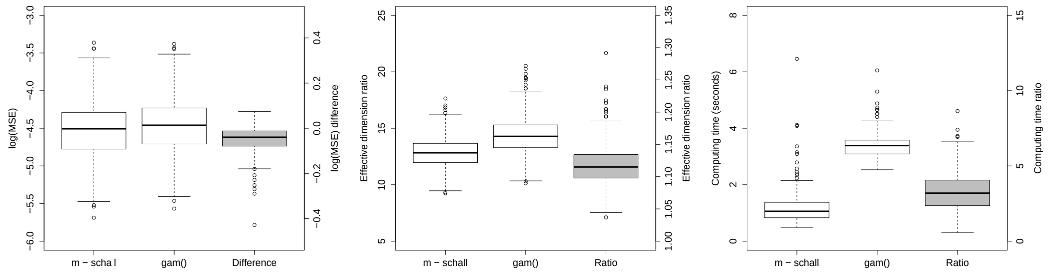


(a) log (MSE)

(b) Effective Dimension

(c) Computing Time (seconds)

Figure 1: Comparisons of the performance of the m-schall algorithm and the Wood (2011)'s method for the two dimensional case. The boxplots show the results for the Gaussian distribution, different levels of noise $\sigma \in \{0.1; 0.5; 1\}$, a sample size of $n = 200$ and $R = 500$ replicates. From top to botton: (a) log(MSE), (b) Effective dimension, and (c) Computing time. In each case, the two left boxplots show the log(MSE), the effective dimension or the computing time achieved by each approach (left y-axis), while the one of the right (right y-axis) shows: (a) the m-schall log(MSE) minus the log(MSE) of the `gam()` function, (b) the ratio of the effective dimension of the Wood (2011)'s method to the m-schall algorithm; and (c) the ratio of the computing time of the `gam()` function to the m-schall algorithm.



(a) log(MSE) (b) Effective Dimension (c) Computing Time (seconds)

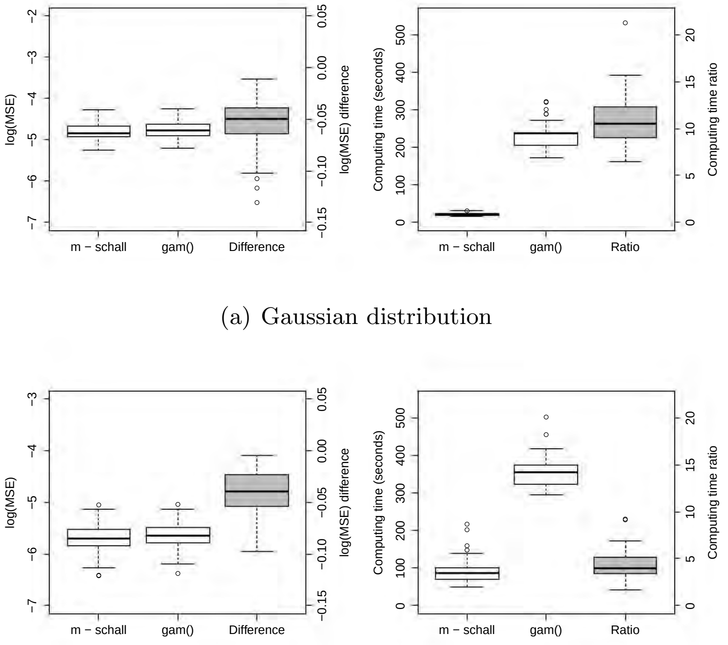
Figure 2: Comparisons of the performance of the m-schall algorithm and the Wood (2011)'s method for the two dimensional case. The boxplots show the results for the Bernoulli distribution, a sample size of $n = 200$ and $R = 500$ replicates. From left to right: (a) log(MSE), (b) Effective dimension, and (c) Computing time. In each case, the two left boxplots show the log(MSE), the effective dimension or the computing time achieved by each approach (left y-axis), while the one of the right (right y-axis) shows: (a) the m-schall log(MSE) minus the log(MSE) of the `gam()` function, (b) the ratio of the effective dimension of the Wood (2011)'s method to the m-schall algorithm; and (c) the ratio of the computing time of the `gam()` function to the m-schall algorithm.

6 Application to real data

In this section, we illustrate the utility of the computational algorithm presented in Section 3 using two real examples. The first example shows the performance of our approach in the simplest case, a 2D case, but with a rather large sample size that requires a relatively large number of inner knots. With the second example, we illustrate how the algorithm can also be used in a 3D case, with non gaussian response. Moreover, since the data in this case is in an array structure, we also take the advantage of the posibility of using GLAM in this context.

6.1 Precipitation Data

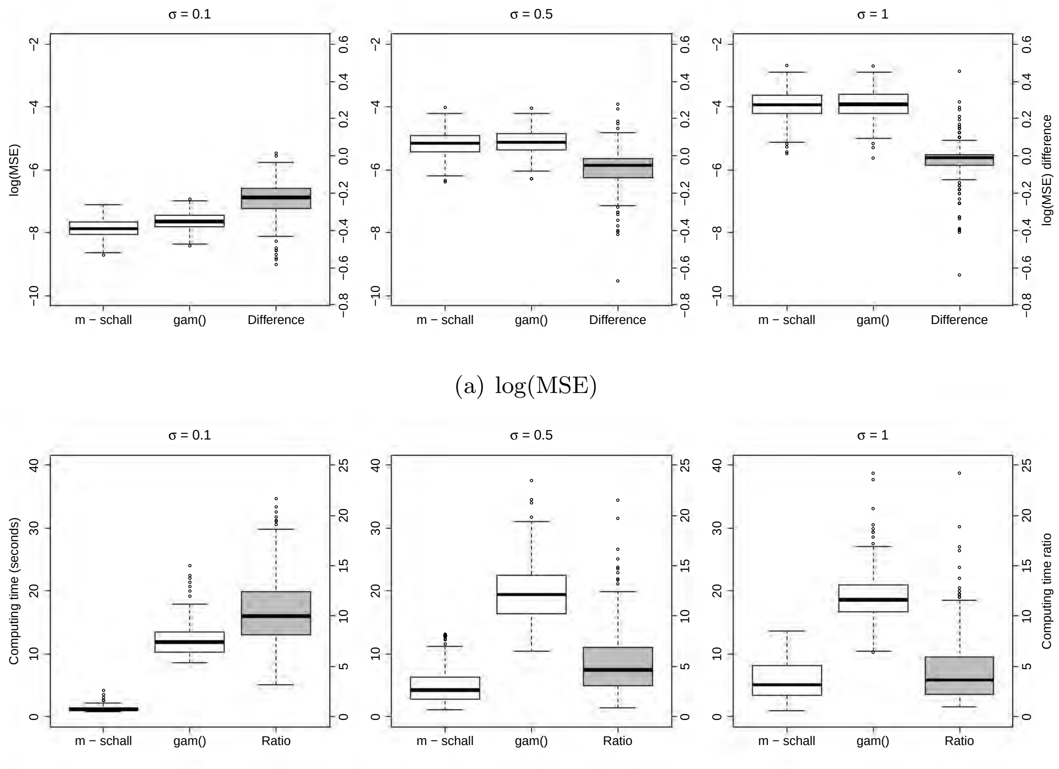
This dataset contains weather observation records compiled in the United States of America (USA). The data came from the National Climatic Data Center (NCDC) of the USA, and contain monthly total precipitation (in millimeters) from January 1895 to December 1997. For illustration purposes, we focus our analysis on estimating the spatial pattern of precipitation for April 1948 in the USA. This restricted dataset can be found in the



(a) Gaussian distribution

(b) Bernoulli distribution

Figure 3: Comparisons of the performance of the m-schall algorithm and the Wood (2011)'s method for the two dimensional case. The boxplots show the results for a sample size of $n = 1000$ and $R = 100$ replicates. From top to botton: (a) Gaussian distribution and (b) Bernoulli distribution. In each case, the two left boxplots show the log(MSE) or the computing time achieved by each approach (left y-axis), while the one of the right (right y-axis) shows the m-schall log(MSE) minus the log(MSE) of the `gam()` function or the ratio of the computing time of the Wood (2011)'s method to the m-schall algorithm.



(a) log(MSE)

(b) Computing Time (seconds)

Figure 4: Comparisons of the log (MSE) and the computing time (in seconds) performance for the m-schall algorithm and the Wood (2011)'s method for the three dimensional case. The boxplots show the results for the Gaussian distribution, different levels of noise $\sigma \in \{0.1; 0.5; 1\}$, a sample size of $n = 500$ and $R = 500$ replicates. Top figure: log (MSE). Botton figure: Computing time (seconds). In each case, the two left boxplots show the log (MSE) or computing time for each approach (left y-axis), while the one of the right (right y-axis) shows: (top) the m-schall log(MSE) minus the log(MSE) of the `gam()` function; and (botton) the ratio of the computing time of the Wood (2011)'s method to the m-schall algorithm.

R-package spam, under the name USprecip, avaliable from cran.r-project.org (R Core Team 2013). Specifically, the dataset comprises a total of 11918 records. For each record, the longitude-latitude position of monitoring stations is provided, jointly with the monthly total precipitation in millimeters and a standardization of this raw observation, called *anomaly* (see Johns et al. 2003). From these 11918 records, only 5906 correspond to stations where monthly total precipitation values were observed, and the remainder 6012 correspond to missing station precipitation values, that have been filled in using spatial statistics (Johns et al. 2003). We therefore restricted our analysis to the 5906 true records.

Figure 5(a) shows the raw data of the monthly precipitation anomalies in USA for April 1948. Using our aproach, we fitted a 2D P-spline model with longitude and latitude as covariates, second order penalties and 40 inner knots for each marginal cubic B-spline basis. The model had therefore a basis dimension of 1936. The convergence tolerance of the variance components was set to 1×10^{-6}, and REML was used. The fitted surface is shown in Figure 5(b). The effective dimension for longitude and latitude was 302.656 and 408.757 respectively. As regards the computing time, the algorithm took 5.76 minutes.

For comparison purposes, we also analyzed this dataset using the gam() and bam() functions in R-package mgcv (version 1.7-22) (Wood 2006a). As pointed out before, the bam() function has been specially designed to deal with very large datasets. However, since a severe convergence problem was observed in the simulations when using this funcion, this dataset was therefore also analyzed using the gam() function. As before, tensor product smoothers, as well as second order penalties and 40 inner knots for each marginal cubic B-spline basis were used, and the REML criterion (method = "REML" and method = "fREML" for gam() and bam() respectively) was chosen for the automatic selection of the smoothing parameters (Wood 2011). In both cases, the numerical options for the fitting process were those established by default, and the fitting processes converged. Regarding the results using the bam() function, the fitted model had an effective dimension of 774.50, and the computing time achieved by this approach was 22.168 minutes, about 3.8 times more than with using our algorithm. As for the gam() function is concerned, the effective dimension was 796.1 and the computing time was increased until 48.217 minutes, 8.4 times more than using the proposed approach. All these numerical results are summarized in Table 1. It should be noted that the total effective dimension also incorporates the dimension of the unpenalized or parametric part.

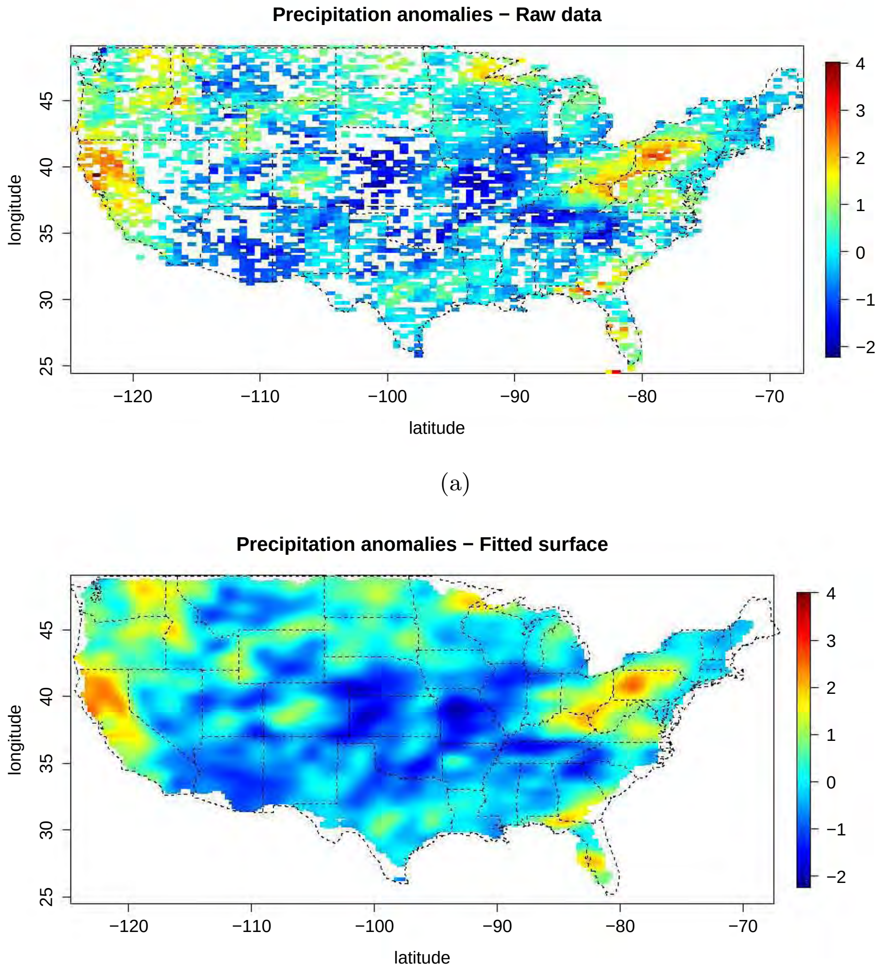
Table 1: Values of the effective dimension (ed) and the computing time (in minutes) for the analyses of the precipitation data

Approach	Longitude ed_1	Latitude ed_2	Total ed	Computing time (min)	Ratio
m-schall	302.66	408.76	715.42	5.761	-
te - `bam()`	-	-	775.50	22.168	3.85 : 1
te - `gam()`	-	-	797.10	48.217	8.37 : 1

6.2 Respiratory Data

This example uses American data on the number of deaths from respiratory disease (Currie et al 2006). The dataset contains the number of deaths according to the age at death (ranging from 1 to 105), the calendar year of death (from 1959 to 1998), and the month of death (ranged 1 to 12). The dataset also contains the number of days per month and year. Specifically, the dataset presents an array structure of dimension 105 × 40 × 12, yieding a total of 50400 observations. This feature offers us the opportunity of using, in combination with our approach, GLAM for the computation of the model matrices involved in (17).

Following the paper by Currie et al (2006) we modeled the number of deaths with a 3D P-spline model (with age, year and month as covariates) with Poisson error and log-link. The logarithm of the number of days in a month was used as an offset. For all the analyses, second order penalties jointly with 11, 6 and 3 inner knots for the marginal cubic B-spline basis of age, year and month respectively were used, yielding a basis dimension of 1050. Since the number of deaths in 1972 was an extreme oulier, we removed this year from the analyses by giving it a weight of zero (see Currie et al 2006). To speed up the computational time, an initial estimate of $\left(\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t\right)^t$ was obtained by assuming $log\{(\boldsymbol{y} + 0.5)/\boldsymbol{d}\}$ as an initial estimate of $\boldsymbol{X\beta} + \boldsymbol{Z\alpha}$, where $\boldsymbol{y}$ and $\boldsymbol{d}$ are the vectors containing the number of deaths and the number of days per month respectively. When fitting the model using the `bam()` and `gam()` functions, an initial estimate of $\boldsymbol{\mu}$ (argument `mustart`) was obtained on the basis of the initial estimate of $\left(\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t\right)$ previously explained. Regarding the proposed algorithm, the tolerance for convergence criterion of the variance components and the Fischer's scoring algorithm was set to 1×10^{-6}. As far as the analyses using the `R`-package `mgcv` is concerned, the numerical options for the fitting process were those established by default. To make the comparisons between our approach and those using the `R`-package



(a)

(b)

Figure 5: Monthly precipitation anomalies in USA for April 1948. (a) Raw data. (b) Estimated spatial pattern.

`mgcv` fair, we fitted the model using our algorithm with and without GLAM.

A detailed numerical result of the fitted models is shown in Table 2. With respect to the computing time, the comparisons of the different approaches was done with respect to our algorithm, but without the use of array methods. As can be observed, and as expected, the use of GLAM during the estimation process only has an impact on the computing time, being reduced in about 2.6 times. For the other approaches, the relative increase of

Table 2: Values of the effective dimension (ed) and the computing time (in minutes) for the analyses of the respiratory data

Approach	Age ed_1	Year ed_2	Month ed_3	Total ed	Computing time (min)	Ratio
m-schall	194.34	209.55	62.86	474.74	4.334	–
m-schall GLAM	194.35	209.55	62.86	474.74	1.696	0.39 : 1
te - `bam()`	-	-	-	639.50	27.496	6.34 : 1
te - `gam()`	-	-	-	638.60	105.388	24.32 : 1

the computing time was even more pronounced than for the precipitation data, about 6.4 times in the case of the `bam()` function and 24.3 for `gam()`. Despite the fact that the total effective dimension of the models fitted using the proposed algorithm and of those using the `R`-package `mgcv` differ by a large extent, the fitted values provided by both approaches were very similar. This can be observed in Figures 6 and 7. Figure 6 shows the histogram of the differences between the fitted values provided by the `gam()` function and those obtained with our algorithm. As can be observed, the majority of the differences lie between -5 and 5, a rather small range if we take into account that the observed number of deaths ranged between 0 and 1605. In Figure 7 the estimated log mortality against age, year, and month for different covariate values is shown. The black line corresponds to the proposed algorithm, and the red line to the `gam()` function. Again, it can be observed that both approaches have yielded similar results.

7 Discussion

In this paper we considered the estimation of the smoothing parameters of a multidimensional tensor product generalized P-spline model with anisotropic penalty. On the basis of the mixed model representation of a P-spline and the use of PQL methods, closed-form expressions for the estimates of the variance components were obtained based on both approximate ML and REML. Besides the simple-achieved expressions of the estimates, which avoid the need of using numerical optimization methods, we also presented some computational aspects that can be used for the fast implementation of the proposed algorithm. For data arranged in multidimensional grids, GLAM methods (Currie et al 2006) can also be accommodated, improving even further the computational time. In addition, the

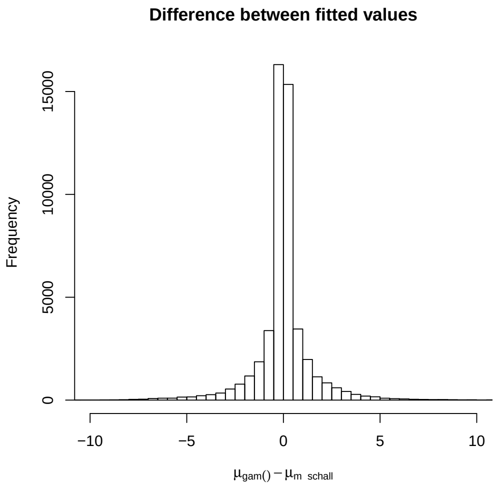
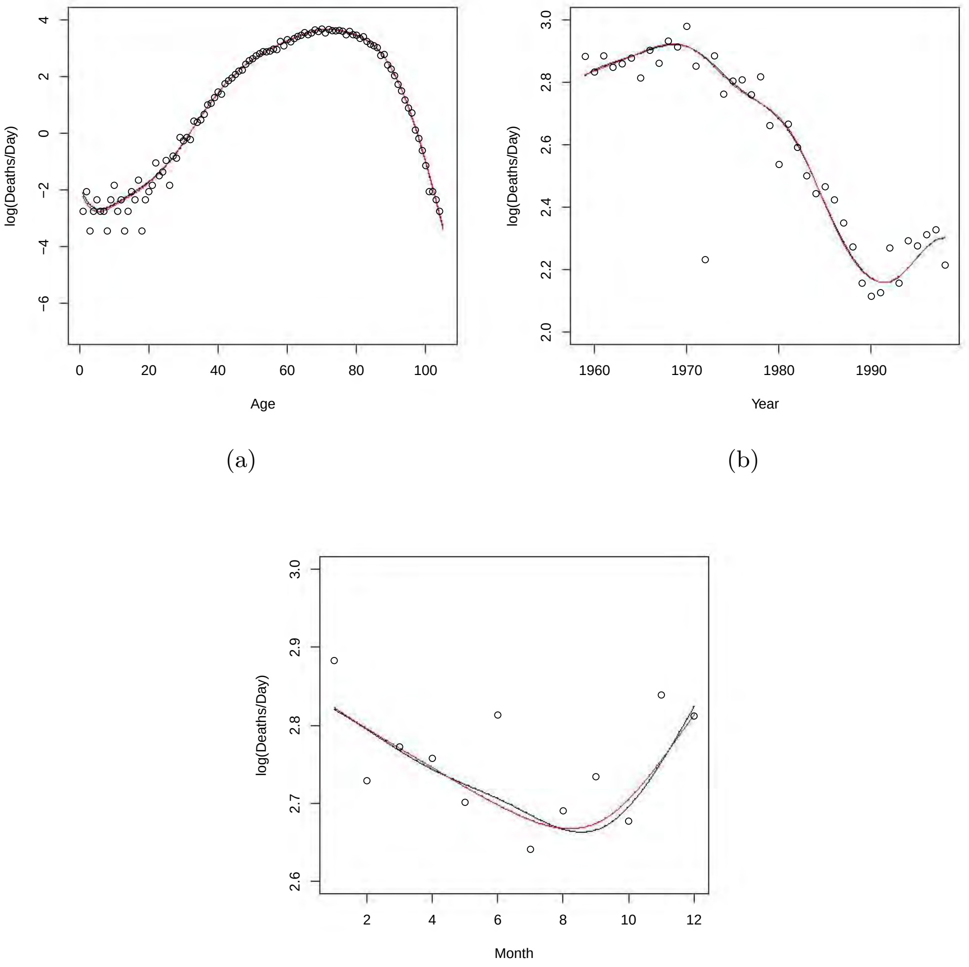


Figure 6: Distribution of the differences between the fitted values provided by the `gam()` function and those obtained with the m-schall algorithm for the respiratory data analysis.

proposed procedure can be easily integrated into the estimation of a GAMM with sets of independent random effects. For the sake of clarity, we focused here in a GAMM specified in terms of univariate effects jointly with a 2D interaction surface. Nevertheless, the m-schall algorithm can be also easily extended to deal with factor-by-surface interactions as well as with ANOVA-type interactions (Lee and Durbán 2011).

Results of the simulation study showed the good performance of the proposed method, in terms of both the MSE and the computing time, when compared with established approaches. It should be noted, however, that an undesirable property of our method is that it is affected by the signal-to-noise ratio. As the signal-to-noise ratio increases, differences between the new proposal and the method proposed by Wood (2011) become smaller. Although in the simulation study our method outperformed Wood (2011)'s method in all cases, this is an area that requires further investigation.

In both the simulation study and the precipitation data the initial estimates of the model's fixed and random effects were established to zero and the variance components to one. We are aware that more suitable initial estimates could even improve the behaviour of the proposed algoritm, yielding better computing times as well as avoiding convergence failures in the estimation procedure. As far as the fixed and random effects is concerned, our experience suggests that specifying an initial estimate of $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\alpha}$ on the basis of



(a)

(b)

(c)

Figure 7: Observed (∘) and smoothed (solid line) numbers of log(deaths/day) by age, year, and month. Black line: proposed algorithm. Red line: `gam()` function. (a) January 1959; (b) age 53 years, January; and (c) age 53 years, 1959.

the response vector $\boldsymbol{y}$ and the link function $g(\cdot)$, as done in the respiratory data example, usually provide good starting values for $\left(\boldsymbol{\beta}^t, \boldsymbol{\alpha}^t\right)^t$. In fact, we applied the m-schall algorithm to the two examples presented in the introduction of Wood (2008), by specifying as initial estimates those establish by default in the `gam()` function ($\boldsymbol{\eta} = g\left(\left(\boldsymbol{y} + 0.5\right)/2\right)$ for both the mackerel egg and the simulated data). In both cases, the m-schall algorithm converged and the obtained results were similar to those using Wood (2011)'s method.

It is well known that PQL methods can suffer from severe bias (Breslow and Clayton 1993, Lin and Breslow 1996), specially for clustered binary data when the cluster size is small. It can therefore be expected that the method proposed in this paper also inherits this behaviour. An extensive simulation study has been conducted (results not shown) to evaluate the practical performance of the m-schall algorithm in different scenarios, supplying, in general, good results. Nonetheless, the bias-corrected procedure proposed by Lin and Zhang (1999) for the GAMM framework can be easily accommodated into the m-schall algorithm. The study of computationally efficient ways for incorporating bias-corrected procedures in this setting remains an interesting area of research.

When it came to presenting the extension of the proposed procedure to the GAMM framework, sets of independent random effects were assumed. This random effect structure implies a diagonal variance-covariance matrix of the random effects, thus allowing the immediate incorporation of the m-schall algorithm into this context. Although this random effect structure might be sufficient in a wide area of real applications, as for instance in multilevel studies, a current line of research is focused on investigating the possibility of applying the m-schall algorithm in longitudinal studies with possibly correlated random intercepts and slopes.

A possible drawback of a tensor product P-spline model is that it assumes a smooth surface, i.e. a smooth transition of the effect across the whole surface. In some practical applications, however, more complex situations could arise, with effects that may not change in some regions of the surface, while changing rapidly in other regions. In these circumstances, the assumption of a single smoothing parameter for each covariate might be not sufficient to capture such local effect, and adaptive P-splines (Lang and Brezger 2004, Krivobokova et al. 2008) have been suggested. In adaptive P-splines the global smoothing parameters are replaced by locally adaptive smoothing parameters, thus allowing more flexibility. The extension of the m-shall algorithm to adaptive anisotropic P-splines is a current line of research.

Finally, software implementing the m-schall algorithm for the 2D and 3D cases can be obtained from the corresponding author. For the time being, the code consists of several easy-to-use functions, designed with our sights set on a future `R` package.

Acknowledgements

The authors would like to express their gratitude for the support received in the form of the Spanish Ministry of Economy and Competitiveness grants MTM2011-28285-C02-01 and MTM2011-28285-C02-02. Work of María Xosé Rodríguez - Álvarez was supported by grant CA09/0053 from the Instituto de Salud Carlos III. The research of Dae-Jin Lee was funded by an NIH grant for the Superfund Metal Mixtures, Biomarkers and Neurodevelopment project 1PA2ES016454-01A2.

Appendix A: Fixed and random effects coefficients estimation

For given values of the variance components τ_d $(d = 1, 2)$ and ϕ, estimation of the fixed and random effects coefficients of model (4), can be obtained by maximizing, with respect to $\boldsymbol{\beta}$ and $\boldsymbol{\alpha}$, the approximate penalized log-likelihood (see equation (6) in Breslow and Clayton 1993)

$$-\frac{1}{2\phi}\sum_{i=1}^{n} Dev_i\left(y_i, \mu_i\right) - \frac{1}{2}\boldsymbol{\alpha}^t\boldsymbol{G}^{-1}\boldsymbol{\alpha},$$

where Dev_i denotes the deviance. This maximization can be carried out on the basis of a Fisher-Scoring algorithm, involving a working dependent variable and a weight matrix, which should be updated at each iteration. Specifically, at $(k{+}1)$th Fisher-Scoring iteration, the working vector $\boldsymbol{z}$ is obtained as

$$z_i = g(\mu_i^{(k)}) + (y_i - \mu_i^{(k)})g'(\mu_i^{(k)}),$$

and the model's fixed and random effects are then estimated as

$$\begin{aligned}
\hat{\boldsymbol{\beta}}^{(k+1)} &= \left(\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V}^{-1}\boldsymbol{z}, && (15)\\
\hat{\boldsymbol{\alpha}}^{(k+1)} &= \boldsymbol{G}\boldsymbol{Z}^t\boldsymbol{V}^{-1}\left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\alpha}}^{(k+1)}\right)\\
&= \boldsymbol{G}\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{z}, && (16)
\end{aligned}$$

where

$$\begin{aligned}\boldsymbol{V} &= \boldsymbol{W}^{-1} + \boldsymbol{Z}\boldsymbol{G}\boldsymbol{Z}^t,\\ \boldsymbol{P} &= \boldsymbol{V}^{-1} - \boldsymbol{V^{-1}X}\left(\boldsymbol{X^tV^{-1}X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V^{-1}},\end{aligned}$$

and $\boldsymbol{W}$ is a diagonal matrix of weights with elements $w_{ii} = \left\{\phi g'(\mu_i^{(k)})^2\nu(\mu_i^{(k)})\right\}^{-1}$.

From a computational point of view, a more convenient method for jointly obtaining $\hat{\boldsymbol{\beta}}$ and $\hat{\boldsymbol{\alpha}}$ is by the solution of the linear system (see equation (9) in Breslow and Clayton 1993)

$$\underbrace{\begin{bmatrix}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{X} & \boldsymbol{X}^t\boldsymbol{W}\boldsymbol{Z}\boldsymbol{G}\\ \boldsymbol{Z}^t\boldsymbol{W}\boldsymbol{X} & \boldsymbol{I}+\boldsymbol{Z}^t\boldsymbol{W}\boldsymbol{Z}\boldsymbol{G}\end{bmatrix}}_{\boldsymbol{C}}\begin{bmatrix}\hat{\boldsymbol{\beta}}^{(k+1)}\\ \hat{\boldsymbol{b}}^{(k+1)}\end{bmatrix} = \begin{bmatrix}\boldsymbol{X}^t\boldsymbol{W}\boldsymbol{z}\\ \boldsymbol{Z}^t\boldsymbol{W}\boldsymbol{z}\end{bmatrix} \tag{17}$$

where $\hat{\boldsymbol{b}}^{(k+1)} = \boldsymbol{G}^{-1}\hat{\boldsymbol{\alpha}}^{(k+1)}$. Note that (17) corresponds to the normal equations of the best linear unbiased estimation (BLUE) of $\boldsymbol{\beta}$ and the best linear unbiased prediction (BLUP) of $\boldsymbol{\alpha}$ under the *working* linear mixed model

$$\boldsymbol{z} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{Z}\boldsymbol{\alpha} + \epsilon, \quad \text{with} \quad \boldsymbol{\alpha} \sim N(\boldsymbol{0}, \boldsymbol{G}) \quad \text{and} \quad \boldsymbol{\epsilon} \sim N(\boldsymbol{0}, \boldsymbol{W}^{-1}).$$

Appendix B: Derivatives of the approximate restricted maximum likelihood with respect to the variance components

Given the approximate restricted log-likelihood

$$l^* = \underbrace{-\frac{1}{2}\log|\boldsymbol{V}|}_{I} - \underbrace{\frac{1}{2}\log|\boldsymbol{X^tV^{-1}X}|}_{II} - \underbrace{\frac{1}{2}(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^t\boldsymbol{V}^{-1}(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}})}_{III}.$$

the corresponding derivatives with respect to the variance components τ_d^2 $(d = 1, 2)$ of each component are

Part I.

$$\frac{\partial \log|\boldsymbol{V}|}{\partial\tau_d^2} = trace\left(\boldsymbol{V}^{-1}\frac{\partial\boldsymbol{V}}{\partial\tau_d^2}\right).$$

PartII.

$$\begin{aligned}
\frac{\partial \log|\boldsymbol{X^t V^{-1} X}|}{\partial \tau_d^2} &= trace\left(\left(\boldsymbol{X^t V^{-1} X}\right)^{-1}\boldsymbol{X}^t \frac{\partial \boldsymbol{V^{-1}}}{\partial \tau_d^2}\boldsymbol{X}\right)\\
&= -trace\left(\left(\boldsymbol{X^t V^{-1} X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V^{-1}}\frac{\partial \boldsymbol{V}}{\partial \tau_d^2}\boldsymbol{V^{-1}X}\right)\\
&= -trace\left(\boldsymbol{V^{-1}X}\left(\boldsymbol{X^t V^{-1} X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V^{-1}}\frac{\partial \boldsymbol{V}}{\partial \tau_d^2}\right).
\end{aligned}$$

Part III.

$$\begin{aligned}
\frac{\partial(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^t\boldsymbol{V}^{-1}(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}})}{\partial \tau_d^2} &= (\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}})^t\frac{\partial \boldsymbol{V^{-1}}}{\partial \tau_d^2}\left(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\\
&= -\left(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^t\boldsymbol{V^{-1}}\frac{\partial \boldsymbol{V}}{\partial \tau_d^2}\boldsymbol{V^{-1}}\left(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\\
&= -\left(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)^t\boldsymbol{V}^{-1}\boldsymbol{Z}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\boldsymbol{Z}^t\boldsymbol{V}^{-1}\left(\boldsymbol{z}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\right)\\
&= -\hat{\boldsymbol{b}}^t\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\hat{\boldsymbol{b}}\\
&= -\hat{\boldsymbol{\alpha}}^t\boldsymbol{G}^{-1}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\boldsymbol{G}^{-1}\hat{\boldsymbol{\alpha}}.
\end{aligned}$$

Thus,

Part II + Part III

$$\begin{aligned}
-\frac{1}{2}(\log|\boldsymbol{V}|+\log|\boldsymbol{X^t V^{-1} X}|) &= -\frac{1}{2}trace\left(\left(\boldsymbol{V}^{-1}-\boldsymbol{V^{-1}X}\left(\boldsymbol{X^t V^{-1} X}\right)^{-1}\boldsymbol{X}^t\boldsymbol{V^{-1}}\right)\frac{\partial \boldsymbol{V}}{\partial \tau_d^2}\right)\\
&= -\frac{1}{2}trace\left(\boldsymbol{P}\frac{\partial \boldsymbol{V}}{\partial \tau_d^2}\right)\\
&= -\frac{1}{2}trace\left(\boldsymbol{P}\boldsymbol{Z}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\boldsymbol{Z}^t\right)\\
&= -\frac{1}{2}trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\right).
\end{aligned}$$

It then follows that

$$\frac{\partial l^*}{\partial \tau_d^2} = -\frac{1}{2}trace\left(\boldsymbol{Z}^t\boldsymbol{P}\boldsymbol{Z}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\right)+\frac{1}{2}\hat{\boldsymbol{\alpha}}^t\boldsymbol{G}^{-1}\frac{\partial \boldsymbol{G}}{\partial \tau_d^2}\boldsymbol{G}^{-1}\hat{\boldsymbol{\alpha}}.$$

Appendix C: Approximate maximum likelihood estimates of the variance components

The variance components estimates are obtained by maximizing the approximate log-likelihood

$$l = -\frac{1}{2}\log|\boldsymbol{V}| - \frac{1}{2}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}})^t\boldsymbol{V}^{-1}(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}}).$$

Taking derivatives with respect to the variance components τ_d^2 $(d = 1, 2)$, we obtain (see Section 3 and Appendix B for details)

$$2\frac{\partial l}{\partial \tau_d^2} = -\frac{1}{\tau_d^2} trace\left(\boldsymbol{Z}^t\boldsymbol{V}^{-1}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}\right) + \frac{1}{\tau_d^4}\hat{\boldsymbol{\alpha}}^t\boldsymbol{\Lambda}_d\hat{\boldsymbol{\alpha}}.$$

By equating the above expression to zero, the ML estimates of the variance components are obtained

$$\hat{\tau}_d^2 = \frac{\hat{\boldsymbol{\alpha}}^t\boldsymbol{\Lambda}_d\hat{\boldsymbol{\alpha}}}{\mathrm{ed}_d},$$

where

$$\mathrm{ed}_d = trace\left(\boldsymbol{Z}^t\boldsymbol{V}^{-1}\boldsymbol{Z}\boldsymbol{G}\frac{\boldsymbol{\Lambda}_d}{\tau_d^2}\boldsymbol{G}\right).$$

Finally, in those situations where ϕ is unknown, it is estimated as

$$\hat{\phi} = \frac{\left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}\right)^t\widetilde{\boldsymbol{W}}\left(\boldsymbol{z} - \boldsymbol{X}\hat{\boldsymbol{\beta}} - \boldsymbol{Z}\hat{\boldsymbol{\alpha}}\right)}{n - \sum_{d=1}^2 \mathrm{ed}_d}.$$

where $\widetilde{\boldsymbol{W}} = \phi\boldsymbol{W}$.

References

de Boor, C.A.: A practical guide to splines. Revised Edition. Springer-Verlag, New York (2001).

Breslow, N.E. and Clayton, D.G.: Aproximated inference in generalised linear mixed models. Journal of the American Statistical Association. **88**, 9-25 (1993).

Currie, I., Durban. M. and Eilers, P.H.C.: Generalized linear array models with applications to multidimensional smoothing. Journal of the Royal Statistical Society, Series B. **68**, 259 – 280 (2006).

Currie, I. and Durban. M.: Flexible smoothing with P-splines: a unified approach. Statistical Modelling. **4**, 333 – 349 (2002).

Eilers, P.H.C., Currie, I. and Durban. M.: Fast and compact smoothing on large multidimensional grids. Computational Statisticas & Data Analysis. 50, 61 - 76 (2006).

Eilers, P.H.C. and Marx, B.D.: Flexible smoothing with B-splines and penalties. Statistical Science. **11**, 89 – 121 (1996).

Eilers, P.H.C. and Marx, B.D.: Multivariate calibration with temperature interaction using two-dimensional penalized signal regression. Chemometrics and intelligent laboratory systems. **66**, 159 - 174 (2003).

Fahrmeir, L., Kneib, T. and Lang S.: Penalized structured additive regression for space-time data: a Bayesian perspective. Statistica Sinica. **14**, 715 – 745 (2004).

Hastie, T.J. and Tibshirani, R.J.: Generalized Additive Models. Chapman and Hall, London (1990).

Hastie, T.J. and Tibshirani, R.J.: Varying-coefficient models. Journal of the Royal Statistical Society. Series B. **55**, 757–796 (1993)

Harville, D.A.: Maximum Likelihood Approaches to Variance Component Estimation and to Related Problems. Journal of the American Statistical Association. **72**, 320 - 338 (1977).

Johns, C., Nychka, D., Kittel, T. and Daly, C: Infilling sparse records of spatial fields. Journal of the American Statistical Association. **98**, 796 – 806 (2003).

Krivobokova T., Crainiceanu, C.M. and Kauermann, G.: Fast Adaptive Penalized Splines. Journal of Computational and Graphical Statistics, **17**, 1–20 (2008).

Lang, S. and Brezger, A.: Bayesian P-splines. Journal of Computational and Graphical Statistics. **13**, 183-212 (2004).

Lee, D.-J.: Smothing mixed model for spatial and spatio-temporal data. PhD thesis, Department of Statistics, Universidad Carlos III de Madrid, Spain (2010).

Lee, D.-J. and Durbán, M.: P-spline ANOVA-type interaction models for spatio-temporal smoothing. Statistical Modelling. **11**, 49 - 69 (2011).

Lee, D.-J., Durbán, M. and Eilers, P.H.C.: Efficient two-dimensional smoothing with P-spline ANOVA mixed models and nested bases. Computational Statistics & Data Analysis, to appear (DOI: http://dx.doi.org/10.1016/j.csda.2012.11.013).

Lin, X. and Breslow, N.E.: Bias Correction in Generalized Linear Mixed Models With Multiple Components of Dispersion. Journal of the American Statistical Association. **91**, 1007 – 1016 (1996).

Lin, X. and Zhang, D.: Inference in generalized additive mixed models using smoothing splines. Journal of the Royal Statistical Society, Series B. **61**, 381 – 400 (1999).

Pawitan, Y.: In All Likelihood: Statistical Modelling and Inference Using Likelihood. Oxford University Press, USA (2001).

R Core Team. R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria. URL http://www.R-project.org/ (2013).

Ruppert, D., Wand, M.P. and Carroll, R.J.: Semiparametric Regression. Cambridge: Cambridge University Press (2003).

Schall, R.: Estimation in generalized linear models with random effects. Biometrika. **78**, 719 - 721 (1991).

Stiratelli, R., Laird, N.M., and Ware, J.H.: Random effects models with serial observations with binary responses. Biometrics. **40**, 719 -727 (1984).

Wand, M.P.: Smoothing and mixed models. Computational Statistics. **18**, 223 – 249 (2003).

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

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

Wood, S.N.: Generalized Additive Models. An introduction with R. Chapman & Hall/CRC (2006a).

Wood, S.N.: Low-rank scale-invariant tensor product smooths for generalized additive models. Journal of the Royal Statistical Society, Series B. **70**, 495 – 518 (2006b).

Wood, S.N.: Fast stable direct fitting and smoothness selection for generalized additive models. Biometrics. **62**, 1025 – 1036 (2008).

Wood, S.N.: Fast stable restricted maximum likelihood and marginal likelihood estimation of semiparametric generalized linear models. Journal of the Royal Statistical Society, Series B. **73**, 3 – 36 (2011).

Wood, S.N., Scheipl, F. and Faraway, J.J: Straightforward intermediate rank tensor product smoothing in mixed models. Statistics and Computing. **23**, 341 – 360 (2013).