

MOX-Report No. 85/2023

**Computationally efficient techniques for Spatial Regression with
Differential Regularization**

Arnone, E.; De Falco, C.; Formaggia, L.; Meretti, G.; Sangalli, L.M.

MOX, Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

mox-dmat@polimi.it

<https://mox.polimi.it>

1 **COMPUTATIONALLY EFFICIENT TECHNIQUES FOR SPATIAL**
2 **REGRESSION WITH DIFFERENTIAL REGULARIZATION**

3 ELEONORA ARNONE*, CARLO DE FALCO†, LUCA FORMAGGIA†, GIORGIO
4 MERETTI†, AND LAURA M. SANGALLI†

5 **Abstract.** We investigate some computational aspects of an innovative class of PDE-regularized
6 statistical models: Spatial Regression with Partial Differential Equation regularization (SR-PDE).
7 These physics-informed regression methods can account for the physics of the underlying phenomena
8 and handle data observed over spatial domains with nontrivial shapes, such as domains with con-
9 cavities and holes or curved domains. The computational bottleneck in SR-PDE estimation is the
10 solution of a computationally demanding linear system involving a low-rank but dense block. We
11 address this aspect by innovatively using Sherman–Morrison–Woodbury identity. We also investigate
12 the efficient selection of the smoothing parameter in SR-PDE estimates. Specifically, we propose ad
13 hoc optimization methods to perform Generalized Cross-Validation, coupling suitable reformulation
14 of key matrices, e.g., those based on Sherman–Morrison–Woodbury formula, with stochastic trace
15 estimation, to approximate the equivalent degrees of freedom of the problem. These solutions permit
16 high computational efficiency also in the context of massive data.

17 **Key words.** Smoothing, semiparametric regression with roughness penalties, functional data
18 analysis, spatial data analysis, Sherman–Morrison–Woodbury identity, stochastic trace estimation

19 **AMS subject classifications.** 62G05, 62G08, 65D10

20 **1. Introduction.** This paper deals with some computational aspects of a novel
21 class of statistical models: Spatial Regression with Partial Differential Equation regu-
22 larization (SR-PDE) [see, e.g., the review in [Sangalli, 2021](#)]. These models can handle
23 spatial and functional data with possibly complicated shapes, observed over multi-
24 dimensional domains. SR-PDE constitutes a new addition to an extremely versatile
25 category of semiparametric and nonparametric methods, extensively used in appli-
26 cations, and based in turn on smoothers such as univariate and multivariate splines,
27 thin-plate splines and spherical splines [see, e.g., the textbooks by [Wahba, 1990](#),
28 [Green and Silverman, 1994](#), [Ruppert et al., 2003](#), [Wood, 2017](#), [Wang, 2019](#), and refer-
29 ences therein], and more recently on soap film smoothing [[Wood et al., 2008](#)] and on
30 bivariate-splines over triangulations [[Lai and Schumaker, 2007](#), [Baramidze et al., 2006](#),
31 [Lai and Wang, 2013](#), [Wang et al., 2020](#)]. The essential form of the estimation problem
32 considered by these methods consists in the minimization of a regularized least-square
33 functional, where the regularization involves suitable (partial) differential operators.
34 In particular, the regularizing term in SR-PDE involves a partial differential equation
35 that encodes the available problem-specific information about the phenomenon under
36 study. Such PDE is defined over the spatial domain over which the data are observed,
37 which may display a non-trivial geometry and non-Euclidean features, such as con-
38 cavities and holes, or a folded nature. Figure 1, for instance, illustrates the modeling
39 of a neuroimaging signal observed over the cerebral cortex. Here the cortex, repre-
40 sented by a two-dimensional Riemannian manifold and suitably approximated by a
41 triangular mesh, constitutes the domain over which the data, i.e., the neuroimaging
42 signal, are observed. The method is designed to provide estimates in the context of
43 massive datasets over domains approximated by meshes having thousands of nodes
44 (as for instance the mesh representing the cortical surface of the brain, involving

*Dipartimento di Management, Univesità degli Studi di Torino, Italy

†MOX – Dipartimento di Matematica, Politecnico di Milano, Italy

Corresponding author: Laura M. Sangalli, laura.sangalli@polimi.it

45 approximately 32000 nodes). This makes computational tractability a crucial issue.
 46 To tackle this problem, we here propose various solutions that drastically reduce the
 47 computational cost of SR-PDE estimation. These computationally efficient solutions
 48 are implemented in the package `fdaPDE` [Arnone et al., 2022], a R/C++ library released
 49 through The Comprehensive R Archive Network R Core Team [2021], and available
 50 at <http://CRAN.R-project.org/package=fdaPDE>.

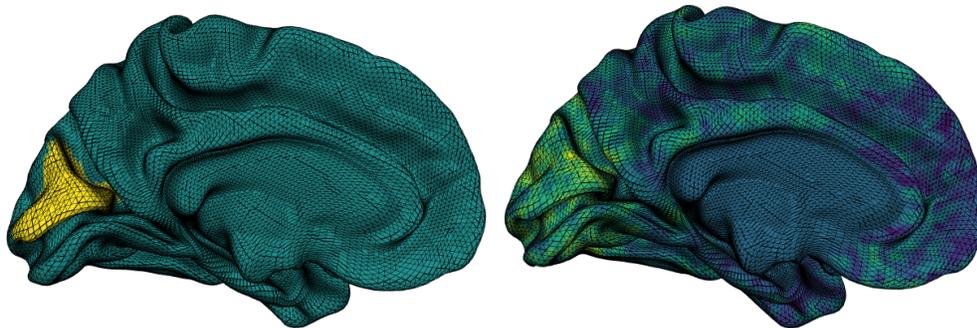


Fig. 1: The figure on the left represents the cortical surface of the brain’s left hemisphere, represented by a triangular mesh with approximately 32000 nodes; the area highlighted in yellow is the cuneus. The figure on the right represent a functional connectivity map, with respect to the cuneus, for an healthy subject; this map, extracted from a functional magnetic resonance imaging scan, indicates the regions of the cortex more highly connected with the cuneus. SR-PDE permits to analyse this signal, appropriately accounting for the non-trivial geometry of the cortical surface.

51 We first focus on efficient solutions for SR-PDE estimation problems. This prob-
 52 lem is discretized by means of finite elements over triangular meshes that approximate
 53 the spatial domain of interest. After discretization, the estimation problem reduces to
 54 the solution of a linear system, which may involve a low-rank but dense block, when
 55 covariates are included in the model. In particular, numerical simulations show that
 56 the cost of this operation is approximately linear in the number of mesh nodes when
 57 a purely nonparametric model is considered (i.e., in the absence of covariates), but is
 58 super-linear when a semiparametric model is used to account for covariate information.
 59 Standard techniques, such as those usually employed in classical semiparametric con-
 60 texts, based for instance on smoothing splines and thin-plate splines [see, e.g., Wahba,
 61 1990, Hastie and Tibshirani, 1990, Green and Silverman, 1993, Wood, 2017], exploit
 62 the band-limited representations of the key matrices involved in these splines repre-
 63 sentations. Unfortunately, the usage of finite elements in SR-PDE produces sparse
 64 but not intrinsically banded systems. In general, the sparsity degree of the system
 65 matrix and the pattern of its non-zero entries depends on the geometry of the mesh
 66 and on node ordering. For this reason, here we derive ad hoc efficient solutions to
 67 address semiparametric SR-PDE problems. These are based on numerical linear alge-
 68 bra methods, such as appropriate reformulations of the estimation problem based on
 69 Sherman–Morrison–Woodbury (SMW) formula [Sherman and Morrison, 1950, Wood-
 70 bury, 1950]. These solutions dramatically decrease the computational cost, enabling
 71 the use of SR-PDE with massive datasets and large meshes.

72 We hence focus on an efficient selection of the smoothing parameter that trades
 73 off data fidelity and regularity of SR-PDE estimates. An appropriate selection of this

74 smoothing parameter is indeed crucial to obtain meaningful estimates. The value
 75 of the smoothing parameter is here selected via minimization of Generalized Cross-
 76 Validation (*GCV*), a well-established performance criterion for automatic parameter
 77 tuning, first conceived by Craven and Wahba [1978/79] and Golub et al. [1979], in clas-
 78 sical smoothing settings. Unfortunately, the evaluation of *GCV* is computationally
 79 demanding. Indeed, it requires the computation of the trace of the so-called smooth-
 80 ing matrix, whose expression in turns involves the inversion of a large and partly
 81 dense matrix. The trace estimation problem has been investigated in the classical
 82 context of smoothing spline regression [see, e.g., Bates and Wahba, 1983, Hutchinson
 83 and de Hoog, 1985, 1986/87, Utreras, 1981], but always taking advantage of band-
 84 limited representations of spline matrices, which are instead unavailable for SR-PDE.
 85 Inspired by the work of Hutchinson [1989], we here propose to estimate the trace of
 86 the smoothing matrix via Monte-Carlo approximation. In particular, we managed
 87 to combine the Hutchinson estimator with the SMW reformulation of the estimation
 88 problem, thus drastically reducing the time required for the calculation of *GCV*. The
 89 resulting algorithm is then incorporated in a Newton-type optimization based on finite
 90 differences. This automatizes the selection of the smoothing parameter and efficiently
 91 locates the optimal one.

92 The present paper is structured as follows. Section 2 provides a self-contained
 93 description of the fundamental SR-PDE estimation problem. For simplicity of expo-
 94 sition, we focus here on the most basic formulation of SR-PDE; we briefly discuss the
 95 numerical discretization of the estimation problem and the resulting linear system
 96 whose solution produces the SR-PDE estimator. Section 3 studies numerical linear
 97 algebra solutions, based on SMW matrix identity, to speed up the resolution of the
 98 system presented in Section 2. The differences in execution times are highlighted
 99 in Section 4, where we compare the proposed approach based on the identity of the
 100 SMW matrix with the standard solution to the estimation problem based on sparse
 101 LU decomposition, as well as to solution based on iterative methods. The following
 102 sections focus on *GCV* computation. Section 5 introduces the concept of equivalent
 103 degrees of freedom (*edf*) for *GCV* evaluation and proposes an innovative SMW-based
 104 stochastic estimator to speed up their computation. Section 6 reports a simulation
 105 study that shows the performance of the method proposed in Section 5. In Section 7
 106 we apply SR-PDE to the study of neuronal connectivity on the cerebral cortex, show-
 107 ing the high level of complexity that the methodology is able to consider, thanks to
 108 the computationally efficient strategies investigated in this work. Section 8 briefly
 109 outlines some more complex SR-PDE estimation problems; such modeling extensions,
 110 for instance, to space-time data, are already implemented in the `fdapDE` library, and
 111 exploit the efficient computational techniques here described, suitably adapted to
 112 these more general model settings.

113 **2. Background.** Let Ω be a two-dimensional domain, and, for simplicity of
 114 exposition, assume $\Omega \subset \mathbb{R}^2$, with boundary $\partial\Omega \in \mathcal{C}^2$. We will later comment on the
 115 case where Ω is a two-dimensional manifold. Let $\{\mathbf{p}_i = (x_i, y_i)\}_{i=1}^n \in \Omega$ be a finite
 116 set of known locations. At each point \mathbf{p}_i , a noisy evaluation $z_i \in \mathbb{R}$ of a variable
 117 of interest is available. Moreover, a q -dimensional vector of deterministic covariates
 118 $\mathbf{w}_i = (w_{i1}, \dots, w_{iq})^\top$ is also observed. We assume that the data generation process
 119 satisfies a semiparametric model with additive error:

$$120 \quad z_i = \mathbf{w}_i^\top \boldsymbol{\beta} + f(\mathbf{p}_i) + \varepsilon_i, \quad i = 1, \dots, n,$$

121 where $\boldsymbol{\beta} \in \mathbb{R}^q$ is a vector of regression coefficients, $f : \Omega \rightarrow \mathbb{R}$ is a twice-differentiable
 122 deterministic field, and $\{\varepsilon_i\}_{i=1}^n$ are independent random errors, also called residuals,
 123 with zero mean and constant variance σ^2 . Both $\boldsymbol{\beta}$ and f are unknown, and their
 124 estimation is ill-posed without further assumptions. The SR-PDE approach proposes
 125 to estimate the couple $(f, \boldsymbol{\beta})$ by minimizing the following penalized sum-of-square-
 126 error functional, trading off data-fidelity and model-fidelity:

$$127 \quad (2.1) \quad J_\lambda(\boldsymbol{\beta}, f) = \sum_{i=1}^n (z_i - \mathbf{w}_i^\top \boldsymbol{\beta} - f(\mathbf{p}_i))^2 + \lambda \int_{\Omega} (\mathcal{L}f - u)^2,$$

128 where $\lambda > 0$ is a tuning parameter. The regularizing term involves a PDE, $\mathcal{L}f = u$,
 129 that encodes the available problem-specific information. In particular, \mathcal{L} is a second-
 130 order, linear differential operator with known, smooth, bounded coefficients, possibly
 131 spatially varying in Ω , of the form:

$$132 \quad \mathcal{L}f = -\operatorname{div}(K\nabla f) + \mathbf{b} \cdot \nabla f + cf.$$

133 $K \in \mathbb{R}^{2 \times 2}$ is a bounded symmetric and positive definite diffusion tensor, $\mathbf{b} \in \mathbb{R}^2$ is a
 134 bounded transport vector and $c \geq 0$ a reaction factor. The PDE parameters K and \mathbf{b}
 135 are convenient tools to model anisotropic effects. Moreover, they can vary over Ω , thus
 136 modeling nonstationarity. Further flexibility is enabled by the forcing term $u \in L^2(\Omega)$.
 137 The problem-specific information may also concern the conditions that f satisfies at
 138 the boundary of the domain: $\mathcal{B}_c f = \gamma$ on $\partial\Omega$. \mathcal{B}_c indicates the linear operator that
 139 implements Dirichlet, Neumann or Robin boundary conditions, or a combination of
 140 the three [see, for a complete treatment of boundary conditions, [Azzimonti et al.,](#)
 141 [2014](#)]. The higher the parameter λ , the stronger the PDE regularization. Conversely,
 142 if λ is chosen small, the solution is more adapted to the data. When no knowledge on
 143 the phenomenon under investigation is available, isotropic smoothing can be obtained
 144 by setting $\mathcal{L} = \Delta$ (the Laplacian operator) and $u = 0$.

Let $H^2(\Omega)$ denote the Sobolev space of twice differentiable functions with two
 distributional derivatives in $L^2(\Omega)$. We estimate $(\boldsymbol{\beta}, f)$ minimizing [Equation \(2.1\)](#) in
 $\mathbb{R}^q \times V_\gamma(\Omega)$, where $V_\gamma(\Omega)$ is a suitable subset of $H^2(\Omega)$, made of functions compliant
 with the boundary conditions:

$$V_\gamma(\Omega) = \{f \in H^2(\Omega) : \mathcal{B}_c f = \gamma\}.$$

145 The estimation problem is formalized as follows.

146 **PROBLEM 2.1.** Find $(\hat{f}, \hat{\boldsymbol{\beta}}) \in V_\gamma(\Omega) \times \mathbb{R}^q$ such that

$$147 \quad (\hat{f}, \hat{\boldsymbol{\beta}}) = \underset{(f, \boldsymbol{\beta}) \in V_\gamma(\Omega) \times \mathbb{R}^q}{\operatorname{arg\,min}} J_\lambda(\boldsymbol{\beta}, f).$$

148 We denote by $\mathbf{z} = (z_1, \dots, z_n)^\top$ the vector of observations at the locations and
 149 by $\mathbf{1}_n$ the vector of \mathbb{R}^n composed of ones. Then, we define the design matrix $W =$
 150 $[w_{ij}] \in \mathbb{R}^{n \times q}$, whose i -th row is \mathbf{w}_i^\top . We assume $q < n$ and that W is full-rank. We
 151 also assume that $\mathbf{1}_n \notin \operatorname{Range}(W)$. This condition is standard in the semiparametric
 152 regression framework since the constant term, i.e., the intercept of the regression
 153 model, is already included in the nonparametric term f . Let H be the projection
 154 matrix onto the image of W , $\operatorname{Im}(W)$, and Q the projector onto $\operatorname{Im}(W)^\perp$, i.e.,

$$155 \quad (2.2) \quad H = W(W^\top W)^{-1}W^\top, \quad Q = I_n - H,$$

156 where I_n is the $n \times n$ identity matrix. Note that $W^\top W$ is invertible thanks to the
 157 fact that $q < n$ and W is full-rank.

158 The minimization problem is well posed under mild regularity conditions on the
 159 differential operator \mathcal{L} and the boundary conditions [see, e.g., [Azzimonti et al., 2014](#)],
 160 and \hat{f} satisfies the following fourth-order variational problem:

$$161 \quad (2.3) \quad \mathbf{v}_n^\top Q \hat{\mathbf{f}}_n + \lambda \int_{\Omega} (\mathcal{L}v)(\mathcal{L}\hat{f}) = \mathbf{v}_n^\top Q \mathbf{z} + \int_{\Omega} u(\mathcal{L}v), \quad \forall v \in V(\Omega),$$

162 where $\hat{\mathbf{f}}_n = (\hat{f}(\mathbf{p}_1), \dots, \hat{f}(\mathbf{p}_n))^\top$ and $\mathbf{v}_n = (v(\mathbf{p}_1), \dots, v(\mathbf{p}_n))^\top$ are the vectors ob-
 163 tained evaluating \hat{f} and v at the n data locations.

164 Here, we have introduced SR-PDE, assuming Ω is a two-dimensional planar do-
 165 main. However, [Lila et al. \[2016\]](#), [Ettinger et al. \[2016\]](#), [Wilhelm and Sangalli \[2016\]](#)
 166 extended the methodology to the case where Ω is a two-dimensional Riemannian man-
 167 ifold embedded in a 3D space, such as in the case of the neuroimaging data in Figure
 168 1, where Ω is the cortical surface. In this case, the estimation functional to be mini-
 169 mized is similar to the one in [Equation \(2.1\)](#), with the regularizing term replaced by
 170 $\lambda \int_{\Omega} (\Delta_{\Omega} f(\mathbf{p}))^2$, where Δ_{Ω} is the Laplace-Beltrami operator associated with Ω [see,
 171 e.g., [Sario et al., 1977](#), Chapter 2]. The Laplace-Beltrami operator is the most natural
 172 generalization of the concept of Laplacian for fields defined over surfaces embedded
 173 in a 3D space. Its involvement in the regularizing term is meant to penalize the local
 174 curvature of f , in a way that complies with the curved nature of the domain and is
 175 independent from the specific coordinate system used to describe it. The discretiza-
 176 tion of the estimation problem is analogous to the case of the planar domain [see, e.g.,
 177 [Lila et al., 2016](#)]. For this reason, in the following, we will, for simplicity, continue
 178 the exposition assuming Ω is a planar domain.

179 **2.1. Discretization of the estimation problem.** To approximate the solu-
 180 tion, we resort to numerical discretization. To this end, we characterize [Equation \(2.3\)](#)
 181 using lower-order expressions. For clarity of exposition, we present the discretization
 182 for homogeneous Neumann boundary conditions, that is $\mathcal{B}_c f = \frac{\partial f}{\partial n} = 0$ on $\partial\Omega$. Nev-
 183 ertheless, we point out that `fdapde` library also implements homogeneous Dirichlet,
 184 nonhomogeneous Dirichlet and mixed conditions, as detailed in [Azzimonti et al. \[2014\]](#).
 185 First, we decouple (2.3) as an equivalent second-order variational system [see, e.g.,
 186 [Azzimonti et al., 2014](#)]. Let a be the following bilinear form associated with operator
 187 \mathcal{L} : $a(f, v) = \int_{\Omega} [K \nabla f \cdot \nabla v + (\mathbf{b} \cdot \nabla f)v + cfv]$. The mixed weak formulation of (2.3)
 188 becomes: let $V = [V(\Omega) \cap C^0(\overline{\Omega})] \times H^1(\Omega)$ and find $(\hat{f}, \hat{g}) \in V$ such that

$$189 \quad \begin{cases} \mathbf{v}_n^\top Q \hat{\mathbf{f}}_n + \lambda a(\hat{f}, v) = \mathbf{v}_n^\top Q \mathbf{z} & \forall v \in V, \\ - \int_{\Omega} (\hat{g}w) + a(\hat{f}, w) = \int_{\Omega} (uw) & \forall w \in V. \end{cases}$$

191 Let \mathcal{T} be a triangulation of Ω and let $\Omega_{\mathcal{T}}$ be the union of the triangles in \mathcal{T} .
 192 We denote by \mathbb{P}_r the space of polynomials of maximal order $r \in \mathbb{N}_0$ with $r \geq 1$,
 193 and we consider the finite element space of globally continuous, piecewise polynomial
 194 functions:

$$195 \quad V_{\mathcal{T}}^r(\Omega) = \{ \mathbf{v}_h \in C^0(\overline{\Omega_{\mathcal{T}}}) : \mathbf{v}_h|_T \in \mathbb{P}_r, \forall T \in \mathcal{T} \}.$$

196 Let $\boldsymbol{\psi} = (\psi_1, \dots, \psi_N)^\top$ be the set of Lagrangian basis functions associated with
 197 the nodes $\{\xi_1, \dots, \xi_N\}$ of $\Omega_{\mathcal{T}}$. Clearly, $V_{\mathcal{T}}^r(\Omega) = \text{span}\{\boldsymbol{\psi}\}$, therefore, any $v_{\mathcal{T}} \in V_{\mathcal{T}}^r(\Omega)$
 198 can be expressed as $v_{\mathcal{T}}(\mathbf{p}) = \boldsymbol{\psi}(\mathbf{p})^\top \mathbf{v}$, where $\mathbf{v} = (v_1, \dots, v_N)^\top \in \mathbb{R}^N$ is a vector that

199 collects the projections of $v_{\mathcal{T}}$ onto the basis. In particular, due to the Lagrangian
200 property, we have $v_i = v_{\mathcal{T}}(\boldsymbol{\xi}_i)$.

201 Let $\Psi = [\Psi_{ij}] = [\psi_j(\mathbf{p}_i)]$ be the matrix $n \times N$ whose entry ij -th is the evaluation
202 of the j -th basis function at the i -th spatial location. Moreover, let us consider the
203 $N \times N$ matrices

$$204 \quad R_0 = \int_{\Omega_{\mathcal{T}}} \boldsymbol{\psi} \boldsymbol{\psi}^{\top} \quad \text{and} \quad R_1 = \int_{\Omega_{\mathcal{T}}} \left(\nabla \boldsymbol{\psi}^{\top} K \boldsymbol{\psi} + \nabla \boldsymbol{\psi}^{\top} \mathbf{b} \boldsymbol{\psi}^{\top} + c \boldsymbol{\psi} \boldsymbol{\psi}^{\top} \right),$$

205 and the vector $\mathbf{u} = \int_{\Omega_{\mathcal{T}}} u \boldsymbol{\psi} \in \mathbb{R}^N$. The regularity conditions on the operator \mathcal{L}
206 ensure that R_1 is semi-positive with $\ker(R_1) \subseteq \text{span}(\mathbf{1}_N)$. These conditions are for
207 example satisfied when \mathcal{L} is the Laplacian or the Laplace-Beltrami operator and the
208 boundary conditions are homogeneous Neumann [see, e.g., [Azzimonti et al., 2014](#), for
209 the details].

210 There exists a unique pair of estimators $(\hat{\boldsymbol{\beta}}, \hat{f}_{\mathcal{T}}) \in \mathbb{R}^q \times V_{\mathcal{T}}^r(\Omega)$ that solves the
211 discretized counterpart of the estimation problem [see, e.g., [Azzimonti et al., 2014](#),
212 [Sangalli, 2021](#)]. Furthermore,

$$213 \quad (2.4) \quad \hat{\boldsymbol{\beta}} = (W^{\top} W)^{-1} W^{\top} (\mathbf{z} - \hat{\mathbf{f}}_n),$$

214 where $\hat{\mathbf{f}}_n = \Psi \hat{\mathbf{f}}$, $\hat{f}_{\mathcal{T}} = \hat{\mathbf{f}}^{\top} \boldsymbol{\psi}$, and $\hat{\mathbf{f}}$ is obtained by solving the regularized saddle-point
215 problem.

$$216 \quad (2.5) \quad M_S \begin{bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{g}} \end{bmatrix} = \mathbf{b}_S,$$

217 where

$$218 \quad (2.6) \quad M_S = \begin{bmatrix} -\Psi^{\top} Q \Psi & \lambda R_1^{\top} \\ \lambda R_1 & \lambda R_0 \end{bmatrix} \quad \text{and} \quad \mathbf{b}_S = \begin{bmatrix} -\Psi^{\top} Q \mathbf{z} \\ \lambda \mathbf{u} \end{bmatrix}.$$

219 In [Equation \(2.5\)](#), the penalization coefficient $\lambda > 0$ is taken as given.

220 **THEOREM 2.2.** *The matrix M_S is non-singular.*

221 *Proof.* We show that under the stated conditions $\ker(M_S) = \{\mathbf{0}\}$. Set $S =$
222 $\Psi^{\top} Q \Psi$. Note that S is semi-positive definite. Let $\mathbf{v} = [\mathbf{v}_1, \mathbf{v}_2] \in \mathbb{R}^{2N}$ and

$$223 \quad M_S \mathbf{v} = \mathbf{0} \Rightarrow \begin{cases} -S \mathbf{v}_1 + \lambda R_1^{\top} \mathbf{v}_2 = \mathbf{0}_N \\ \lambda R_1 \mathbf{v}_1 + \lambda R_0 \mathbf{v}_2 = \mathbf{0}_N \end{cases}$$

224 where $\mathbf{0}_N$ is the vector of \mathbb{R}^N composed of zeroes. We multiply the first equation by
225 $-\mathbf{v}_1^{\top}$, the second by \mathbf{v}_2^{\top} and sum member by member to obtain

$$226 \quad \mathbf{v}_1^{\top} S \mathbf{v}_1 + \lambda \mathbf{v}_2^{\top} R_0 \mathbf{v}_2 = 0.$$

227 Since R_0 is symmetric positive definite and S is semi-positive definite, we have that
228 $\mathbf{v}_2 = \mathbf{0}_N$. Consequently, we also have

$$229 \quad (2.7) \quad \mathbf{v}_1^{\top} S \mathbf{v}_1 + \lambda \mathbf{v}_1^{\top} R_1 \mathbf{v}_1 = 0.$$

230 If R_1 is positive definite, we immediately obtain that $\mathbf{v}_1 = \mathbf{0}_N$, which concludes the
231 proof. If instead the kernel of R_1 contains the constant vectors, we need to show that
232 $\mathbf{1}_N \notin \ker(S)$. We first note that, by the partition of unity property of finite element
233 shape functions, $\Psi \mathbf{1}_N = \mathbf{1}_n$, so the requirement is equivalent to $\mathbf{1}_n \notin \ker(Q)$.

234 However, $\mathbf{1}_n \in \ker(Q)$ implies, by the definition of Q , that $W(W^{\top} W)^{-1} W^{\top} \mathbf{1}_n = \mathbf{1}_n$,
235 in contradiction with the given hypothesis $\mathbf{1}_n \notin \text{Range } W$. This concludes the proof. \square

236 We will see in [Section 5](#) how an optimal value may be obtained by using GCV. To
 237 this end, it is useful to introduce the following factorization of M_S , which will be used
 238 for the GCV calculation. Since R_0 is non-singular and $\lambda > 0$, M_S can be factorized
 239 as:

$$240 \quad (2.8) \quad M_S = \begin{bmatrix} I_N & R_1^\top R_0^{-1} \\ O_N & I_N \end{bmatrix} \begin{bmatrix} -T & O_N \\ \lambda R_1 & \lambda R_0 \end{bmatrix},$$

241 with O_N the $N \times N$ matrix with all elements equal to zero, and

$$242 \quad (2.9) \quad T = \Psi^\top Q \Psi + \lambda R_1^\top R_0^{-1} R_1.$$

243 T is clearly non-singular under the same hypothesis of [Theorem 2.2](#), since it is a
 244 Schur complement. Therefore, $\hat{\mathbf{f}}$ is the solution of $T\hat{\mathbf{f}} = \Psi^\top Q\mathbf{z} + \lambda R_1^\top R_0^{-1}\mathbf{u}$, and then

$$245 \quad (2.10) \quad \hat{\mathbf{f}}_n = \Psi \hat{\mathbf{f}} = S\mathbf{z} + \lambda \Psi T^{-1} R_1^\top R_0^{-1} \mathbf{u},$$

246 where

$$247 \quad (2.11) \quad S = \Psi T^{-1} \Psi^\top Q.$$

248 Thanks to [Equation \(2.4\)](#), [Equation \(2.10\)](#) and [Equation \(2.11\)](#), we can find the fitted
 249 values $\hat{\mathbf{z}} = W\hat{\beta} + \hat{\mathbf{f}}_n$ as

$$250 \quad \hat{\mathbf{z}} = S_Q \mathbf{z} + \mathbf{r},$$

251 where $\mathbf{r} = \lambda Q \Psi T^{-1} R_1^\top R_0^{-1} \mathbf{u}$ and

$$252 \quad (2.12) \quad S_Q = H + QS$$

253 is a symmetric and positive definite matrix that we shall name *smoothing matrix*, in
 254 analogy to more classical linear semiparametric regression models [see, e.g., [Eubank,](#)
 255 [1999](#), [Green and Silverman, 1993](#)].

256 **3. System solving.** To compute the solution of the problem we solve directly
 257 [Equation \(2.5\)](#), without resorting to [Equation \(2.9\)](#). This allows us to obtain the
 258 misfit of the PDE $\hat{\mathbf{g}}$, together with $\hat{\mathbf{f}}$. This section investigates how to deal with
 259 [Equation \(2.5\)](#) efficiently. We start by investigating the structure of the system matrix
 260 M_S . We recall that M_S is a $2N \times 2N$ matrix, where N is the number of mesh nodes. In
 261 real applications, N is often chosen large to improve the quality of the final estimate.
 262 Furthermore, each block of M_S shows a different degree of sparsity. The north-west
 263 block is the most critical. If we consider a model with covariates, Q is not sparse, and
 264 consequently, the north-west block of M_S becomes dense. Instead, if we study a purely
 265 nonparametric model without covariates, $Q = I_n$, and the north-west block is sparse.
 266 The remaining blocks of the system matrix are always sparse, with a mesh-dependent
 267 sparsity pattern. Thus, in the absence of covariates, the whole matrix M_S is sparse,
 268 and [Equation \(2.5\)](#) can be solved by resorting, for instance, to a sparse LU solver like
 269 the one in the SuiteSparse library [suite](#)¹. The computational cost of this operation
 270 depends on the fill-in. However, the simulations reported in this paper show that
 271 it can easily be less than quadratic in N . The semiparametric case is more critical.
 272 Sparse solvers prove ineffective since M_S has a dense block. Similarly, dense solvers
 273 cannot exploit the partial sparsity of M_S and display at least $\mathcal{O}(N^3)$ complexity.

¹<https://people.engr.tamu.edu/davis/suitesparse.html>

274 We here propose a technique based on SMW decomposition to reduce the com-
 275 puting times when covariates are present. This choice proves valuable to distribute
 276 the computational cost of a single inefficient inversion (that of M_S) on a cascade of
 277 sparse factorizations. In analogous settings, SMW approach was proposed by [Eubank](#)
 278 [et al. \[2004\]](#) to address smoothing spline estimation in varying-coefficient models and
 279 by [Lai and Vemuri \[1997\]](#) for PDE-penalized surface smoothing.

280 In SR-PDE, we start by exploiting the projection property $H+Q = I_n$ to factorize
 281 the system matrix additively

$$282 \quad M_S = \begin{bmatrix} -\Psi^\top \Psi & \lambda R_1^\top \\ \lambda R_1 & \lambda R_0 \end{bmatrix} + \begin{bmatrix} \Psi^\top H \Psi & O_N \\ O_N & O_N \end{bmatrix},$$

283 where O_N is the $N \times N$ zero matrix. We define the $2N \times 2N$ matrices

$$284 \quad (3.1) \quad A = \begin{bmatrix} -\Psi^\top \Psi & \lambda R_1^\top \\ \lambda R_1 & \lambda R_0 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} \Psi^\top H \Psi & O_N \\ O_N & O_N \end{bmatrix}.$$

285 The matrix A is sparse. Indeed, it coincides with the system matrix of a nonparametric
 286 SR-PDE problem, with the same data, but without covariates. In turn, B has just
 287 one dense block, the north-west block, which has a rank of at most $\min\{N, n, q\}$. It
 288 should now be noticed that in practical applications q is typically in the order of units
 289 or at most tens, so that $q \ll \min\{N, n\}$ and $\text{rank}(B) = q$. We can hence exploit the
 290 small rank of the matrix B to propose an efficient solver for the system. We propose
 291 an efficient decomposition of M_S that takes advantage of the SMW matrix identity
 292 [see [Woodbury, 1950](#), [Sherman and Morrison, 1950](#)]. Assume that M_S and A are
 293 invertible. If we have some matrices $U \in \mathbb{R}^{2N \times q}$, $C \in \mathbb{R}^{q \times q}$ and $V \in \mathbb{R}^{q \times 2N}$, with C
 294 invertible, such that $M_S = A + UCV$, then, according to the SMW identity, we have
 295 the following:

$$296 \quad (3.2) \quad M_S^{-1} = (A + UCV)^{-1} = A^{-1} - A^{-1}U(C^{-1} + VA^{-1}U)^{-1}VA^{-1}.$$

297 To exploit the SMW formula, we are left to express B as a suitable UCV product.
 298 From [Equation \(2.2\)](#), we can write

$$299 \quad \Psi^\top H \Psi = \underbrace{\Psi^\top W}_{\tilde{U}} \underbrace{(W^\top W)^{-1}}_{\tilde{C}} \underbrace{W^\top \Psi}_{\tilde{V}}.$$

300 Setting $\tilde{U} = \Psi^\top W \in \mathbb{R}^{N \times q}$, $\tilde{C} = (W^\top W)^{-1} \in \mathbb{R}^{q \times q}$ and $\tilde{V} = W^\top \Psi \in \mathbb{R}^{q \times N}$, we
 301 derive $B = UCV$ with the following definition:

$$302 \quad (3.3) \quad U = \begin{bmatrix} \tilde{U} \\ O_N \end{bmatrix} \in \mathbb{R}^{2N \times q}, \quad C = \tilde{C} \in \mathbb{R}^{q \times q}, \quad V = [\tilde{V} \quad O_N] \in \mathbb{R}^{q \times 2N}.$$

303 The following algorithm synthesizes how to make efficient use of the SMW de-
 304 composition to solve [Equation \(2.5\)](#).

Algorithm 3.1 System solution via SMW decomposition**Require:** $\Psi, W, R_1, R_0, \mathbf{b}_S, \lambda$

- 1: Build A from R_0, R_1, Ψ and λ , as in Equation (3.1);
- 2: SparseLU factorize A and store its factorization;
- 3: **if** $\exists W$ **then**
- 4: Compute and store U, V as in Equation (3.3);
- 5: Solve $\mathbf{A}\mathbf{y} = \mathbf{b}_S$;
- 6: Solve $\mathbf{A}\mathbf{Y} = \mathbf{U}$;
- 7: Compute $\mathbf{G} = \mathbf{W}^\top \mathbf{W} + \mathbf{V}\mathbf{Y}$;
- 8: Factorize \mathbf{G} and store its factorization;
- 9: Solve $\mathbf{G}\boldsymbol{\theta} = \mathbf{V}\mathbf{y}$;
- 10: Solve $\mathbf{A}\mathbf{v} = \mathbf{U}\boldsymbol{\theta}$;
- 11: $\mathbf{x} = \mathbf{y} - \mathbf{v}$.
- 12: **else**
- 13: Solve $\mathbf{A}\mathbf{x} = \mathbf{b}_S$.
- 14: **end if**

Ensure: $\mathbf{x} = \begin{bmatrix} \hat{\mathbf{f}} \\ \hat{\mathbf{g}} \end{bmatrix}$ such that $M_S \mathbf{x} = \mathbf{b}_S$

305 In synthesis, due to the SMW formula applied to Equation (2.5), we can replace
 306 the inefficient inversion of the partially dense matrix M_S with the cheaper inversions
 307 of the sparse matrix A (for which we can use, e.g., a sparse LU solver) and of the
 308 dense, yet very small, $G \in \mathbb{R}^{q \times q}$. Simulations show that this leads to great advantages
 309 in computing times.

310 SMW decomposition proves valuable also for storage reasons. Indeed, it allows
 311 us to completely avoid storing the partially dense $2N \times 2N$ matrix M_S or the dense
 312 $n \times n$ matrix Q . In turn, it requires only the sparse matrix Ψ and the full, but smaller,
 313 $n \times q$ matrix W (we recall that $q \ll n$). Furthermore, since some inversions are to be
 314 performed more than once, fdapde implementation automatically factorizes A and G
 315 and stores the factors in memory for reuse.

316 **4. Numerical experiment with the SMW decomposition.** We wish to
 317 compare the time required to solve Equation (2.5) with the SMW decomposition with
 318 respect to standard solvers to verify the computational advantage represented by the
 319 novel approach. In particular we compare the proposed approach with a standard
 320 sparse LU solver with sparsity preserving reordering [Davis, 2004, Amestoy et al.,
 321 2004, Eaton et al., 2022], and with two iterative solvers: the Biconjugate Gradient
 322 STABILized (BICGSTAB) method and the Generalized Minimal Residual (GMRES)
 323 method [van der Vorst, 1992, Saad and Schultz, 1986, Saad, 2003, Eaton et al., 2022].
 324 In order to improve the performances of the two iterative solvers, we employ as a
 325 preconditioner the matrix A defined in (3.1). Indeed, without a preconditioner, the
 326 two iterative solvers do not converge in a reasonable number of iterations.

327 We consider a square regular domain: $\Omega = [0, 1] \times [0, 1]$. We sample the n data
 328 points $\{z_i\}_{i=1}^n$ according to $z_i = \beta_1 w_{1i} + \beta_2 w_{2i} + f(\mathbf{p}_i) + \varepsilon_i$; see Section 2 for the
 329 notation. The spatial field f is chosen with sharp oscillations,

$$330 \quad f(x, y) = \sin\{2\pi[(0.5 \sin(5\pi y) \exp(-1) + 1)x \cos(1) + y \sin(1)]\} \\
 \cdot \cos\{2\pi[(0.5 \sin(5\pi y) \exp(-1) + 1)x \sin(1) - (0.5 \sin(5\pi x) \exp(-1) + 1)y]\}.$$

331 We consider $q = 2$ stochastic covariates: w_1 is a Gaussian random variable of null

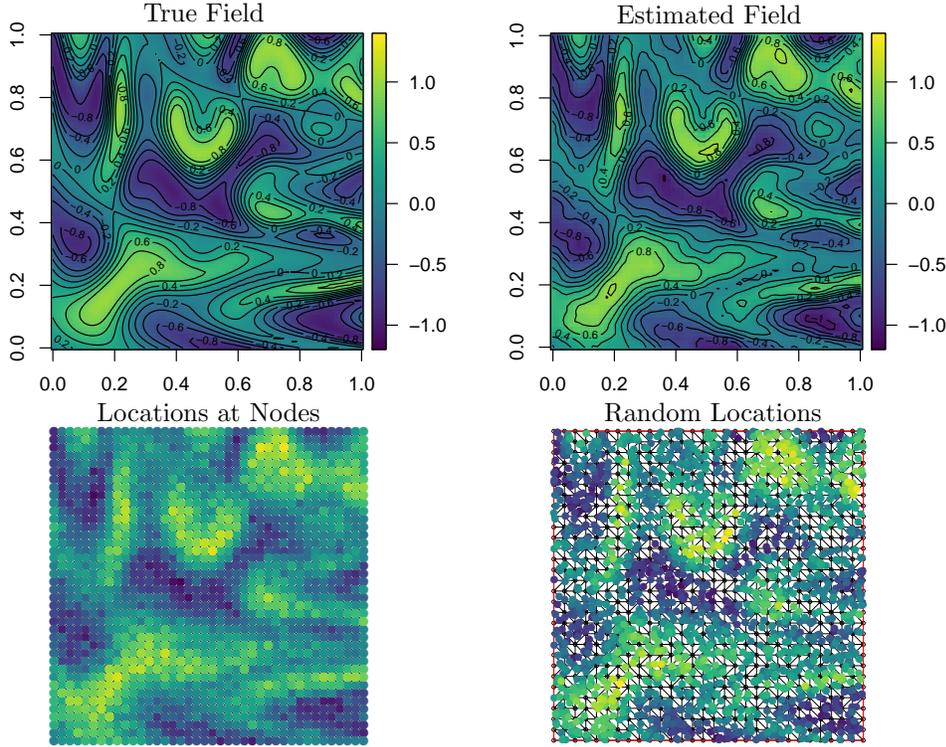


Fig. 2: Top-left: true field f . Top-right: field estimated from data sampled in the first repetition of Simulation 1, using SR-PDE on a regular mesh with 30^2 nodes. Bottom-left: data sampled at mesh nodes. Bottom-right: data sampled at locations randomly scattered over the domain.

332 mean, and standard deviation 0.05 while w_2 comes from an exponential distribution
 333 of mean 0.1. We set $\beta_1 = 2$, $\beta_2 = 0.5$. We sampled ε as the realization of a Gaussian
 334 random variable with zero mean and standard deviation of 5% of the data range, that
 335 is, equal to $0.05(r_2 - r_1)$, where (r_1, r_2) is the range of the total signal $\{\beta_1 w_1 + \beta_2 w_2 +$
 336 $f(\mathbf{p}_i)\}_{i=1}^n$.

337 To address the estimation problem, we consider SR-PDE with Laplacian regular-
 338 ization (i.e., $\mathcal{L} = \Delta$ and $u = 0$ in Equation (2.1)). The smoothing parameter λ is kept
 339 fixed at a value selected by the minimization of GCV, as described in Section 5.

340 **4.1. Simulation 1: increasing number of observations n , increasing**
 341 **number of mesh nodes N .** We consider meshes with increasing refinement, with
 342 N nodes on regular square lattices, and N takes values $30^2, 40^2, \dots, 90^2$. For each
 343 value of N , we sample $n = N$ observations. We examine two scenarios: sampling an
 344 observation at each mesh node or sampling the observations at locations randomly
 345 scattered over the domain. In both cases, we measure the average time employed
 346 by `fdapDE` library to solve the problem over 30 replicas of each experimental setting,
 347 with four different solution methods:

- 348 1. LU: standard sparse LU solver;
- 349 2. BICGSTAB: the preconditioned BICGSTAB solver;

- 350 3. GMRES: the preconditioned GMRES solver;
 351 4. SMW: sparse LU solver, with SMW decomposition.

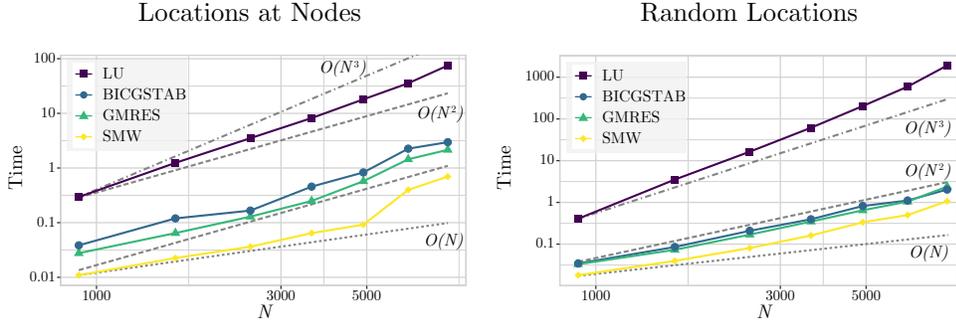


Fig. 3: Simulation 1: increasing number of observations n , increasing number of mesh nodes N ; average computing time, over 30 simulation replicates, with the four considered solvers. Left: data locations at mesh nodes, as in the bottom-left panel of Figure 2. Right: data locations randomly scattered over the domain, as in the bottom-right panel of Figure 2.

352 We start with the first setting: locations coincident with the mesh nodes. Figure 3,
 353 left, shows that the comparison of the computing times for the four methods, strongly
 354 favors the proposed SMW approach. Indeed, SMW shows a significantly lower CPU
 355 time, for all the considered dimensions. Moreover, LU times display a rate of growth,
 356 with respect to N , of order between 2 and 3, while the order for iterative approaches
 357 and SMW is between 1 and 2. The reason behind the inefficiency of the standard LU
 358 sparse solver is that it applies the sparse solver to M_S , which has a $N \times N$ full north-
 359 west block. Table 1 reports the mean CPU times over the 30 repetitions (and the
 360 standard deviations of CPU times in brackets) for all considered solvers and meshes
 361 dimensions, on a Intel Core i7-4510U, 2.6 GHz, 8 GB RAM machine.

362 Table 2 and the right panel of Figure 3 report the CPU times for the second
 363 setting, in which the number of locations n is still equal to the number of nodes N ,
 364 but the coordinates of the locations do not coincide with the coordinates of the mesh
 365 nodes. We observe that, generally, the methods employ a longer time to solve the
 366 system. However, iterative approaches and SMW maintain a computing time of an
 367 order less than quadratic, while LU is largely affected by the lower sparsity of Ψ , and
 368 displays an order that is at least cubic.

369 **4.2. Simulation 2: increasing number of observations n , fixed number**
 370 **of mesh nodes N .** We fix the number of nodes $N = 8100$ and we progressively
 371 increase the amount of observations n , by uniformly sampling on the square domain.
 372 We perform 30 repetitions of each experimental setting.

373 We compare the average computing times with the four methods considered in
 374 the previous simulation. Table 3 reports the CPU times for all the methods. Figure 4,
 375 left, shows the relationship between times and observations. SMW approach is always
 376 faster than the other methods; however, the computational cost increases with n . This
 377 is to be expected since the number of data influences the sparsity Ψ and hence the
 378 overall sparsity of the north-west block of M_S . This fact lowers the efficiency gained by
 379 the SMW method, but the decomposition still proves more effective than the standard

Mesh nodes	LU	BICGSTAB	GMRES	SMW
900	0.29 (0.063)	0.04 (0.022)	0.03 (0.009)	0.01 (0.002)
1600	1.24 (0.157)	0.12 (0.036)	0.06 (0.010)	0.02 (0.003)
2500	3.52 (0.139)	0.17 (0.033)	0.13 (0.021)	0.04 (0.005)
3600	8.20 (0.299)	0.46 (0.085)	0.25 (0.048)	0.06 (0.012)
4900	18.05 (0.582)	0.83 (0.171)	0.57 (0.253)	0.09 (0.023)
6400	35.57 (1.278)	2.26 (0.642)	1.45 (0.431)	0.40 (0.265)
8100	75.06 (5.402)	2.97 (0.263)	2.15 (0.462)	0.70 (0.462)

Table 1: Simulation 1: increasing number of observations n , increasing number of mesh nodes N , locations at mesh nodes. Mean time in seconds taken to solve the system (2.5) over the 30 simulation replicates. In brackets, the standard deviation of the employed time. The errors associated with the computed solutions is of order 10^{-4} for BICGSTAB and 10^{-9} for the other methods.

Mesh nodes	LU	BICGSTAB	GMRES	SMW
900	0.41 (0.092)	0.03 (0.008)	0.03 (0.012)	0.02 (0.004)
1600	3.52 (0.530)	0.09 (0.006)	0.07 (0.004)	0.04 (0.005)
2500	16.39 (1.526)	0.21 (0.046)	0.17 (0.028)	0.08 (0.014)
3600	61.40 (3.918)	0.39 (0.074)	0.35 (0.067)	0.16 (0.032)
4900	203.52 (10.477)	0.82 (0.152)	0.65 (0.132)	0.33 (0.104)
6400	593.44 (35.205)	1.11 (0.151)	1.04 (0.212)	0.50 (0.155)
8100	1902.60 (219.087)	2.03 (0.494)	2.39 (0.665)	1.07 (0.372)

Table 2: Simulation 1: increasing number of observations n , increasing number of mesh nodes N , locations randomly scattered. Mean time in seconds taken to solve the system (2.5) over the 30 simulation replicates. In brackets the standard deviation of the employed time. The errors associated with the computed solutions is of order 10^{-5} for BICGSTAB and 10^{-8} for the other methods.

380 methodology. Computing times grow less linearly with n in the SMW case.

381 **4.3. Simulation 3: fixed number of observations n , increasing number**
 382 **of mesh nodes N .** We sample $n = 1125$ data, whose locations are randomly selected
 383 in the unit square, and gradually increase the number of mesh nodes N .

384 The comparison again favors the SMW method, which, on average, outperforms
 385 the LU by an order of magnitude and shows the same order of BICGSTAB and
 386 GMRES but always with a lower CPU time.

387 From all the simulations, we can conclude that the the preconditioner considered
 388 for the iterative methods performs very well for the problem at hand, since BICGSTAB
 389 and GMRES exhibit the same order of magnitude of the SMW approach. However,
 390 we also have evidence that proposed SMW approach outperforms both the iterative
 391 methods, as well as the standard sparse LU solver.

392 **5. Estimation of λ by GCV .** An appropriate choice of the smoothing param-
 393 eter $\lambda > 0$ is crucial to appropriately balance the data-fidelity and model-fidelity
 394 terms in Equation (2.1). We may evaluate a candidate λ by means of Generalized
 395 Cross-Validation (GCV), a performance criterion originally conceived by Craven and
 396 Wahba [1978/79], Golub et al. [1979]. GCV provides computational advantages with

Observations	LU	BICGSTAB	GMRES	SMW
512	4.77 (1.010)	0.36 (0.093)	0.34 (0.076)	0.29 (0.069)
1024	29.47 (2.162)	0.56 (0.075)	0.51 (0.062)	0.37 (0.043)
2048	151.67 (8.982)	0.78 (0.143)	0.80 (0.246)	0.42 (0.090)
4096	386.77 (20.155)	1.37 (0.306)	1.63 (0.608)	0.54 (0.186)
8192	443.60 (100.019)	2.77 (0.384)	2.49 (0.608)	1.05 (0.329)

Table 3: Simulation 2: increasing number of observations n , fixed number of mesh nodes N . Mean time in seconds taken to solve the system (2.5) over the 30 simulation replicates. In brackets the standard deviation of the employed time. The errors associated with the computed solutions is of order 10^{-4} for BICGSTAB and 10^{-8} for the other methods.

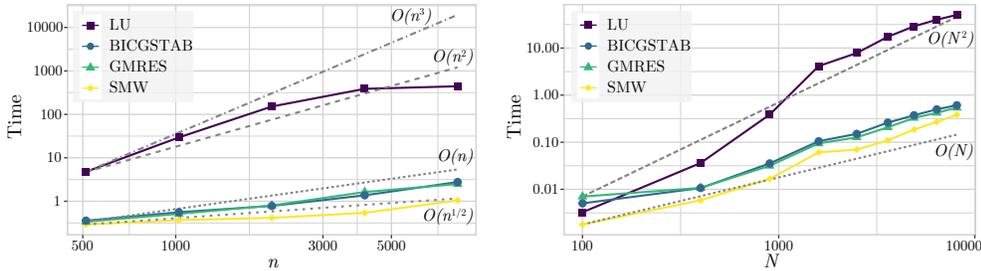


Fig. 4: Left: Simulation 2: increasing number of observations n , fixed number of nodes N (data locations randomly scattered over the domain); average computing time with the four considered solvers. Right: Simulation 3: fixed number of observations n , increasing number of mesh nodes N (data locations randomly scattered over the domain); average computing time with the four considered solvers.

397 respect to other popular statistical loss functions such as, e.g., the Akaike Informa-
 398 tion Criterion, Bayesian Information Criterion, or Mallows' C_p (see, e.g., Konishi and
 399 Kitagawa [2007]). Indeed, it does not require the knowledge of the residual variability
 400 σ^2 . In turn, GCV evaluation is based on the computation of the so-called equivalent
 401 degrees of freedom of the model, defined as

$$402 \quad (5.1) \quad edf = \text{tr}(S_Q) = q + \text{tr}(S)$$

403 where S and S_Q are given in Equation (2.11) and Equation (2.12) respectively.

404 The equivalent degrees of freedom are the sum of two contributions: the number
 405 of regressors $q \in \mathbb{N}$ in the parametric part of the model, and $\text{tr}(S) \in \mathbb{R}$, the degrees
 406 of freedom associated with the estimate of f , the non-parametric part of the model.
 407 The GCV function to be minimized is then derived as:

$$408 \quad GCV(\lambda) = n \sum_{i=1}^n \left(\frac{z_i - \hat{z}_i}{n - edf} \right)^2.$$

409 We consider different methods to efficiently estimate edf . In particular, we develop
 410 two alternative strategies: exact computation and stochastic approximation.

Mesh nodes	LU	BICGSTAB	GMRES	SMW
100	0.01 (0.004)	0.01 (0.005)	0.01 (0.012)	0.01 (0.001)
400	0.04 (0.002)	0.01 (0.001)	0.01 (0.001)	0.01 (0.001)
900	0.39 (0.046)	0.04 (0.006)	0.03 (0.004)	0.02 (0.002)
1600	4.14 (0.501)	0.11 (0.012)	0.09 (0.011)	0.06 (0.008)
2500	7.83 (0.386)	0.15 (0.025)	0.13 (0.017)	0.07 (0.008)
3600	17.34 (1.119)	0.26 (0.046)	0.21 (0.036)	0.11 (0.014)
4900	28.95 (1.358)	0.37 (0.050)	0.33 (0.062)	0.19 (0.028)
6400	39.95 (2.994)	0.50 (0.056)	0.42 (0.057)	0.27 (0.039)
8100	50.69 (3.793)	0.61 (0.123)	0.54 (0.060)	0.38 (0.035)

Table 4: Simulation 3: fixed number of observations n , increasing number of mesh nodes N . Mean time in seconds taken to solve the system (2.5) over the 30 simulation replicates. In brackets the standard deviation of the employed time. The errors associated with the computed solutions is of order 10^{-6} for BICGSTAB and 10^{-9} for the other methods.

411 **5.1. Exact computation of GCV.** We now study how to efficiently compute
412 S , in order to extract its trace, for the computation of the edf in Equation (5.1).
413 Note that S is not explicitly computed for the solution of the estimation problem.
414 Indeed, to solve the estimation problem, we consider the full system Equation (2.6)
415 (see Section 3). Here, instead, we directly resort to the definition of S and T in
416 Equation (2.9). As we can see from Equation (2.11), the most critical step in the def-
417 inition of S is the factorization and inversion of the $N \times N$ matrix T . Unfortunately,
418 since $T = T(\lambda)$, S has to be recomputed every time we investigate a different level
419 of smoothing. Moreover, T is dense, thus, it requires a computationally demanding
420 inversion for large N . The first inversion involved in the definition of T is the one
421 of R_0 . This operation is made less computationally demanding by resorting to mass
422 lumping. Indeed, R_0 is a finite element mass matrix and we can safely use its di-
423 agonal approximation. Moreover, $R = R_1^T R_0^{-1} R_1$ is independent from λ . Thus, its
424 computation has to be performed just once, also when assessing different smoothing
425 levels.

426 Since $S = \Psi T^{-1} \Psi^T Q$ and $\Psi^T Q$ is a $N \times n$ matrix, the computation of S is
427 relatively efficient whenever $n \ll N$, with a cost of approximately $\mathcal{O}(N^2 n)$ and $\gamma \in$
428 $[2, 3]$. We can simplify computations with some algebraic manipulations. For instance,
429 if the locations are a subset of the nodes, Ψ becomes a binary matrix with a single
430 one per row. Then, premultiplying $T^{-1} \Psi^T Q$ by Ψ reduces to the permutation of its
431 columns. Also, the left multiplication by Q can be made more efficient thanks to
Algorithm 5.1.

Algorithm 5.1 Left multiplication by Q

Require: W, \mathbf{x}

- 1: Compute, factorize and store $\Upsilon = W^T W$ for possible reuse;
- 2: Compute $\mathbf{v} = W^T \mathbf{x}$;
- 3: Solve $\Upsilon \mathbf{y} = \mathbf{v}$;
- 4: Output $\mathbf{x} - W \mathbf{y}$.

Ensure: $Q \mathbf{x}$

433 Note that $W \in \mathbb{R}^{n \times q}$ and $q \ll n$. Suppose $\mathbf{x} \in \mathbb{R}^n$, in the worst-case scenario,
 434 a simple multiplication by Q is $\mathcal{O}(n^2)$, while [Algorithm 5.1](#) is $\mathcal{O}(nq + q^3)$. Observe
 435 that [Algorithm 5.1](#) does not require the storage of Q . Moreover, thanks to the cyclic
 436 property of the trace operator, the same algorithm is used in the computation of
 437 $\text{tr}(S) = \text{tr}(\Psi T^{-1} \Psi^\top Q) = \text{tr}(Q \Psi T^{-1} \Psi^\top)$.

438 **5.2. Stochastic approximation of GCV.** An alternative to speed up the com-
 439 putation of GCV is to approximate *edf* by a stochastic technique. Recalling that *edf*
 440 is the trace of a matrix, we consider the stochastic trace estimation first proposed
 441 by [Girard \[1989\]](#) and later improved by [Hutchinson \[1989\]](#). In particular, Hutchin-
 442 son suggests approximating the trace of a symmetric matrix $S \in \mathbb{R}^{N \times N}$ exploiting
 443 the formula $\text{tr}(S) = \mathbb{E}[\mathbf{u}_S^\top S \mathbf{u}_S / (\mathbf{u}_S^\top \mathbf{u}_S)]$, where \mathbf{u}_S is a vector of N independent
 444 samples from a Rademacher distributed random variable. An unbiased estimator of
 445 *edf*, denoted by $\widehat{\text{edf}}$, is then proposed resorting to a Monte Carlo approximation.
 446 Hutchinson also proves that the choice of Rademacher distribution for \mathbf{u}_S makes $\widehat{\text{edf}}$
 447 satisfy the minimum variance criterion among the unbiased estimators of $\text{tr}(S)$, with
 448 $\text{Var}[\mathbf{u}_S^\top S \mathbf{u}_S] = 2 \sum_{i \neq j} S_{ij}^2$.

449 Applications of Hutchinson’s estimator to *edf* approximation have already been
 450 proposed, e.g., by [Golub and von Matt \[1997\]](#) in the context of classical Tikhonov
 451 regularization models. To optimize *edf* estimation in SR-PDE, we combine Hutchin-
 452 son’s approach with SMW decomposition, as seen in [Section 3](#). From [Equation \(2.11\)](#)
 453 we write $\text{tr}(S) = \mathbb{E}[\mathbf{u}_S^\top S \mathbf{u}_S] = \mathbb{E}[\mathbf{u}_S^\top \Psi T^{-1} \Psi^\top Q \mathbf{u}_S]$. In particular, considering [Equa-](#)
 454 [tion \(5.1\)](#), we have

$$455 \quad (5.2) \quad \text{edf} = q + \mathbb{E}[\mathbf{u}_S^\top \Psi T^{-1} \Psi^\top Q \mathbf{u}_S].$$

456 Passing to Monte Carlo estimators, we approximate *edf* by

$$457 \quad (5.3) \quad \text{edf} \approx q + \frac{1}{r} \sum_{i=1}^r \mathbf{u}_i^\top \Psi T^{-1} \Psi^\top Q \mathbf{u}_i,$$

458 where $\{\mathbf{u}_i = (u_{i[1]}, \dots, u_{i[n]})^\top\}_{i=1}^r$ are r i.i.d. samples of vector \mathbf{u}_S , and the Monte
 459 Carlo mean approximates the expected value in [Equation \(5.2\)](#); in particular, all the
 460 components of \mathbf{u}_i follow independent Rademacher distributions, to be simulated, e.g.,
 461 via Bernoulli samples.

462 In order to make the computation more efficient, we take advantage of the SMW
 463 system solution presented in [Section 3](#). In this case, different from what was done in
 464 the computation of the exact GCV, we do not compute T^{-1} explicitly; instead, we
 465 work directly with the whole system [Equation \(2.6\)](#). In particular, as summarized
 466 in [Algorithm 5.2](#), we exploit simultaneous calculations collecting all the $\{\mathbf{u}_i\}_{i=1}^r$ in
 467 a $n \times r$ matrix $U_S = [\mathbf{u}_1, \dots, \mathbf{u}_r]$, and we solve a linear system $M_S = B_S$ with the
 468 right-hand side $B_S \in \mathbb{R}^{2N \times r}$ defined in line 3 of [Algorithm 5.2](#).

469 The number of stochastic realizations r for the Monte Carlo mean in [Equa-](#)
 470 [tion \(5.3\)](#) trades off accuracy and computational complexity. We know that the higher
 471 r , the better the *edf* approximation. The library `fdapde` uses $r = 100$ as the default
 472 value for the number of realizations; this default value has been checked to provide
 473 good approximations in different experimental settings. The user may set different
 474 values, still getting strong computational savings, especially when dealing with mas-
 475 sive datasets and problems with large mesh sizes N . Indeed, the construction of the
 476 $2N \times r$ dense matrix U_S is less demanding than building T . Similarly, the construc-
 477 tion of B_S is favored by the sparsity of Ψ and the use of [Algorithm 5.1](#) for the left

Algorithm 5.2 Stochastic SMW edf computation

Require: Ψ, W, R_0, R_1

- 1: Obtain U_S using Bernoulli distributions;
- 2: Compute and store U, V as in Equation (3.3);
- 3: Build and store $B_S = \begin{bmatrix} \Psi^\top Q U_S \\ O_N \end{bmatrix}$ and $Y = \begin{bmatrix} U_S^\top \Psi \\ O_N \end{bmatrix}$ for possible reuse;
- 4: Solve $M_S X = B_S$ using SMW decomposition;
- 5: Compute $\widehat{edf}_i = q + Y^i \cdot X_i$, $i = 1, \dots, r$
where Y^i denotes the i -th row of Y and X_i the i -th column of X ;
- 6: Compute $\widehat{edf} = \frac{1}{r} \sum_{i=1}^r \widehat{edf}_i$.

Ensure: \widehat{edf}

478 multiplication by Q . Note that Lines 1-3 are in Algorithm 5.2 and do not depend on
 479 λ . Hence, if a user needs to compute edf for more than a single λ , B_S is available for
 480 reuse.

481 Line 4 of Algorithm 5.2 is the bottleneck of the algorithm, but the SMW decom-
 482 position makes it rather efficient. In fact, SMW decomposition takes advantage of the
 483 sparsity pattern of A to make the computing times approximately $\mathcal{O}(N)$. Moreover,
 484 still in Line 4, we solve the system applying SMW decomposition with a right-hand
 485 side B_S , of size $2N \times r$. In the worst case scenario, this operation costs as solving
 486 r times a system like (2.5), one for each column of B_S as the right-hand side. We
 487 compare this step with the bottleneck of the exact algorithm: the inversion of T .
 488 Equation (2.9) shows that T is dense and thus expensive to factorize in terms of
 489 computing time and memory. Moreover, the usage of T in Equation (2.11) costs as
 490 solving a dense system with a $N \times n$ right-hand side.

491 Now consider a fixed number of data points n . We want to evaluate the degrees
 492 of freedom for a vector of m with different values of the smoothing parameter. From
 493 tests performed with `fdapde` library, we observe that the computational cost of the
 494 stochastic strategy proves approximately $\mathcal{O}(Nmr)$ in the average case scenario, while
 495 the cost of the exact method is $\mathcal{O}(N^\gamma mn)$, with $\gamma = \gamma(n) \in [2, 3]$. Since generally
 496 $r \ll n$, the stochastic approach proves to be much more effective than the exact
 497 counterpart, especially in the context of large datasets.

498 **5.3. GCV optimization.** Exploiting the convexity of $GCV(\lambda)$, `fdapde` per-
 499 forms its minimization by a Newton method. Each iteration of Newton optimization
 500 takes advantage of the optimized techniques described in Subsection 5.1 and Subsec-
 501 tion 5.2, to evaluate $edf(\lambda)$. Unfortunately, each Newton step requires the first and
 502 second derivatives of $edf(\lambda)$. These terms are rather straightforward to compute using
 503 exact methods, but their stochastic estimates are too unreliable. For convenience in
 504 `fdapde` we have resorted to approximating the derivatives with second-order finite
 505 differences.

506 `fdapde` can hence rely on either an exact Newton or a stochastic three-points
 507 finite-differences Newton optimization. Simulation studies show that the results pro-
 508 duced by the stochastic method generally have a high degree of accuracy. Moreover,
 509 the number of iterations required by the two approaches is comparable and, in par-
 510 ticular, it is always in the order of units in non-pathological cases.

511 **6. Simulation 4: cost of edf computation.** We aim to compare the methods
 512 described in Section 5 in terms of accuracy and computing time. We consider the
 513 same experimental setting discussed in Subsection 4.3. For a fixed value of λ , we
 514 want to compute both the solution to the estimation problem and $GCV(\lambda)$. We fix
 515 $n = 1225$, and we progressively increase the mesh refinement from 900 to 4900 nodes.
 516 We consider 30 replicas of each experimental setting, with three different methods to
 517 compute edf :

- 518 1. *standard*: without SMW decomposition and with standard edf computation.
 519 This method explicitly computes the north-west block of M_S^{-1} , uses it to
 520 evaluate S and then extracts edf .
- 521 2. *exact*: SMW system solution and GCV calculated as in Subsection 5.1.
- 522 3. *stochastic*: SMW system solution and GCV computed as in Subsection 5.2,
 523 based on $r = 100$ realizations.

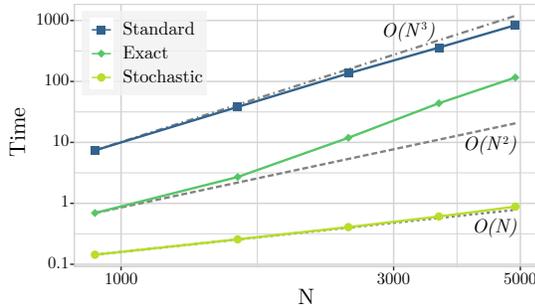


Fig. 5: Simulation 4: cost of edf computation, fixed number of observations n , increasing number of mesh nodes N ; computing time with the standard solution (Standard), with SMW system solution and GCV computed as in Subsection 5.1 (Exact) and with SMW system solution and GCV computed as in Subsection 5.2 based on $r = 100$ realizations (Stochastic).

524 We start from an analysis in terms of execution time. Figure 5 shows that the
 525 *exact* method is generally ten times faster than the *standard* one. Unfortunately,
 526 it still displays a super-linear (almost cubic) trend in the number of mesh nodes.
 527 Conversely, the *stochastic* approach exhibits a linear trend in N . This is because
 528 Algorithm 5.2 is conceived to bypass the slow inversion of T in the formula $S =$
 529 $\Psi T^{-1} \Psi^\top Q$ and, instead, it repeatedly exploits SMW-based system solutions with
 530 different right-hand-sides. Indeed, we recall that Algorithm 5.2 does not only apply
 531 SMW decomposition for system solution but also for edf estimation. Moreover, the
 532 method avoids computing S explicitly, and instead only estimates its trace.

533 It is difficult to evaluate a priori the relative efficiency of the *exact* and the
 534 *stochastic* approaches since this usually depends on the number of mesh nodes. When
 535 N is very small, the *exact* technique might be more effective. Conversely, in a big data
 536 framework, the *stochastic* algorithm compares more favorably, with a discrepancy that
 537 increases as N increases. Accordingly, stochastic edf evaluation is chosen as `fdapDE`
 538 default option for λ selection.

539 We now want to inspect the accuracy of *stochastic edf* estimation. Let us consider
 540 the case with $N = 1600$. Despite the variability of the *stochastic* approach, Figure 6,

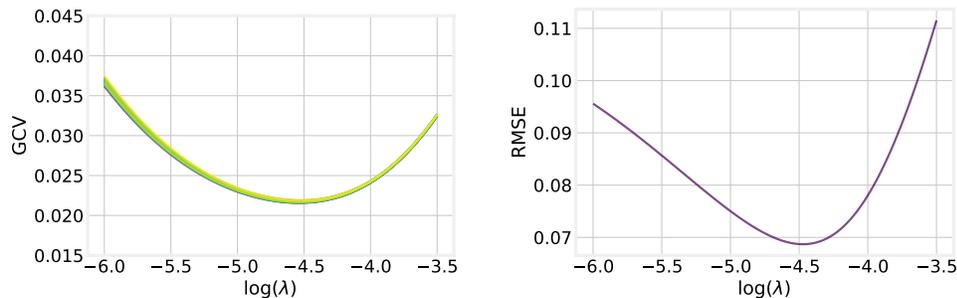


Fig. 6: Simulation 4: cost of edf computation. Left: 20 *stochastic GCV* replicas, $r = 100$. Right: *RMSE*.

541 left, shows that the edf estimation is able to reproduce the *GCV* curve with high
 542 accuracy. In particular, stochasticity is almost negligible in terms of the selection of
 543 the optimal λ when $r = 100$. Consider also the right panel of [Figure 6](#). This plot
 544 shows the value of the Root Mean Square Error (*RMSE*) between the predicted \hat{z}
 545 and the noise-filtered data, $z - \varepsilon$. This is the error we would ideally like to minimize.
 546 Comparing the two plots in the figure, we observe that the minimization of the *GCV*
 547 provides a good criterion for the automatic selection of the smoothing parameter, as
 548 it suggests a value close to the minimizer of *RMSE*. Moreover, the variability
 549 introduced by the stochastic approximation is irrelevant with respect to the error
 550 made by minimizing the *GCV* instead of the unknown *RMSE*.

551 **7. Case study: analysis of neural connectivity.** We here show a more complex
 552 example where we exploit the techniques illustrated in the preceding sections
 553 to analyze neuroimaging data. The recent development of noninvasive neuroimaging
 554 techniques represents a great challenge for the scientific community. Modern imag-
 555 ing techniques are now able to guarantee effective visualization of the human brain
 556 structure, function, and connectivity, with high resolution. Neuroscientists agree (see,
 557 e.g., [Glasser et al. \[2013\]](#) and its references) that a fair amount of the neural activity
 558 captured by brain scans is due to the cerebral cortex: a thin layer of gray matter
 559 with a highly folded geometry. As such, the cortex can be represented as a complex
 560 bidimensional Riemannian manifold, embedded in a 3D space. Still, nowadays, many
 561 neuroimaging studies are carried out neglecting this spatial structure, exploiting 3D
 562 methods that rely on the Euclidean distance. This choice is inappropriate since ar-
 563 eas of the cortex having different functionalities may be close in terms of Euclidean
 564 distance, due to the highly convoluted anatomy of the cortex. In turn, studies based
 565 on 2D geodesic distances along the surface are more likely to capture the intrinsic
 566 geometry of the cortex. This has encouraged the development of new statistical mod-
 567 els designed to fit complex spatial regression problems where data are located on
 568 convoluted domains or surfaces [see, e.g., [Chung et al., 2014](#), [Lila et al., 2016](#)]. As
 569 mentioned in [Section 2](#), SR-PDE naturally encompasses such a feature, being able to
 570 handle data observed over two-dimensional Riemannian manifolds.

571 In this illustrative case study, we analyze a high-dimensional neuroimaging sig-
 572 nal on the brain cortex. Here we consider data collected by the Human Connectome
 573 Project [[Glasser et al., 2013](#)], obtained from functional Magnetic Resonance Imag-
 574 ing (fMRI), on a healthy subject in a resting state. The fMRI signal captures the

575 neural activity on the cerebral cortex, measuring the changes in the concentration of
 576 deoxy-hemoglobin in the blood. The preprocessing pipeline of the Human Connec-
 577 tome Project automatically performs an anatomic alignment of the signal to a freely
 578 available template of cerebral cortex anatomy, obtained by averaging the cortical sur-
 579 face of several healthy adult volunteers. In particular, we consider the left hemisphere
 580 of the template cortex, represented by a triangular mesh with about 32000 nodes; see
 581 [Figure 1](#). The preprocessed fMRI data set consists of a time series for each node of
 582 this triangular mesh.

583 We show an analysis of the Functional Connectivity (FC) obtained from the
 584 fMRI signal. The FC maps allow us to explore the degree of interconnection between
 585 different regions of the cortex. These maps are computed starting from the pairwise
 586 correlation between the signals referred at each node and the average time series on
 587 a selected Region Of Interest (ROI). Since the correlation is restricted to the interval
 588 $[-1, 1]$, the so-called Fisher’s r-to-z transformation [see, e.g., [Fisher, 1915](#)] is hence
 589 applied to obtain an unconstrained signal, leading to the FC map. This map highlights
 590 the areas of the cortex that are more closely related to ROI. In [Figure 1](#), we consider
 591 as ROI the cuneus, a small portion of the occipital lobe of the brain, which is involved
 592 in several basic visual processes. Resting-state cuneus activity allows neuroscientists
 593 to detect, for example, some forms of depression or severe gambling addiction. The
 594 right panel of [Figure 1](#) highlights the cuneus, whilst the left panel shows the FC with
 595 respect to the cuneus, for a healthy subject at resting state.

596 **7.1. SR-PDE analysis.** We consider here a nonparametric approach: $z_i =$
 597 $f(\mathbf{p}_i) + \varepsilon_i$ where z_i is the FC sampled at each node, f is the true FC and ε_i are
 598 random errors. We address the problem of identifying a proper smoothing level, using
 599 the stochastic *GCV* approach seen in [Algorithm 5.2](#). [Figure 7](#), left panel, reports ten
 600 stochastic *GCV* functions, obtained setting different random seeds, sampling each
 601 time $r = 100$ realizations. As in the simulation studies, we observe that the *GCV* is
 602 quite stable. Setting the tolerance to $1e-03$, the minimum is reached after only six
 603 iterations. In particular, the cost of a single stochastic Newton step is relatively small
 604 (approximately 19 seconds on a i7-6700HQ, 2.60 GHz, 8 GB RAM machine).

605 In contrast with the stochastic finite-differences Newton approach, exact *edf* eval-
 606 uation is not able to run on a 16 GB RAM machine, since the computational burden
 607 imposed by a 32k nodes mesh makes the process abort. As a consequence, also the
 608 *standard edf* computation, mentioned [Section 6](#) and used as a default strategy be-
 609 fore conceiving the *exact/stochastic* methods, is not viable. This is a crucial result
 610 because it highlights the two key contributions of the stochastic approach: it is able
 611 to save time at the price of negligible losses in terms of precision; moreover, there are
 612 situations where the stochastic *GCV* is the only computationally viable option.

613 [Figure 7](#), right panel, shows the estimate \hat{f} , obtained with the value of λ selected
 614 with *stochastic GCV*, minimized with Newton method. We observe that the method
 615 is able to choose an appropriate level of smoothing, producing a smooth estimate
 616 that captures the main features of the signal. The analysis could be also enriched
 617 by the inclusion of space-varying covariates in the model, such as for instance the
 618 cerebral cortex thickness. We point out that smoothing is a crucial step for subsequent
 619 analysis, and it enables the use of functional data analysis techniques for these complex
 620 data [see, e.g., [Ferraty and Vieu, 2006](#), [Ramsay and Silverman, 2008](#), [Kokoszka and](#)
 621 [Reimherr, 2017](#), for introductions to functional data analysis].

622 **8. Extension to more complex SR-PDE problems.** The previous sections,
 623 for simplicity of exposition, have focused on a basic formulation of SR-PDE. However,

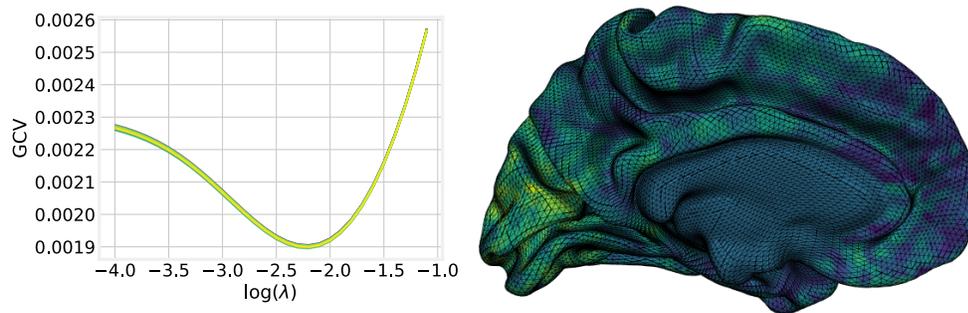


Fig. 7: Analysis of neuroimaging data in Figure 1. Left panel: ten stochastic GCV functions; right panel: an estimate of f .

624 SR-PDE is a rich class of techniques that includes various other more articulated mod-
 625 els, already implemented in the `fdapde` library [see, e.g., the review in Sangalli, 2021].
 626 The efficient techniques described in this work are appropriately adapted to deal with
 627 these more complex model settings. In some of these contexts, the strategies described
 628 here are indeed vital, since the estimation problem has to be solved recursively, as in
 629 the case of generalized linear SR-PDE, or for large spatio-temporal discretization, as
 630 for space-time SR-PDE. This section briefly outlines these two models extensions.

631 The generalized linear version of SR-PDE, developed in Wilhelm and Sangalli
 632 [2016], allows to consider response variables that have any distribution within the
 633 exponential family, thus significantly broadening the possible applications of these
 634 methods. Let Z_1, \dots, Z_n be independent responses coming from a distribution in the
 635 exponential family and assume that

$$636 \quad g(\mathbb{E}[Z_i]) = \theta_i(\boldsymbol{\beta}, f) = \mathbf{w}_i^\top \boldsymbol{\beta} + f(\mathbf{p}_i), \quad i = 1, \dots, n,$$

637 where $\mathbb{E}[Z_i]$ is the expected value of the response variable Z_i , conditionally on the co-
 638 variates, and g is a known link function, determined by the specific distribution of the
 639 response. The two unknowns, $\boldsymbol{\beta}$ and f , are estimated minimizing a functional likewise
 640 (2.1), but where the first term is replaced by $\sum_{i=1}^n l(z_i; \theta_i)$, with $l(\cdot; \theta_i)$ the negative
 641 log-likelihood of the response. This problem is computationally more demanding than
 642 minimization of Equation (2.1) since the new functional is no longer quadratic. The
 643 minimization is tackled with Functional Penalized Iterative Reweighted Least Squares
 644 (F-PIRLS); see Wilhelm and Sangalli [2016]. Each F-PIRLS step requires solving a
 645 weighted least-squares optimization problem of the type (2.1), and is solved using
 646 the SMW decomposition described in Section 3, suitably adapted to account for the
 647 presence of the diagonal weight matrix in the least square term. The selection of the
 648 smoothing parameter is currently performed by minimization of the GCV on a grid
 649 of candidate values, with the GCV evaluated post-convergence of F-PIRLS. Possi-
 650 ble future extensions shall involve the introduction of optimization methods, possibly
 651 performed at each F-PIRLS step, as done, for instance, by Gu [1992], Wood [2000,
 652 2004].

653 SR-PDE is also able to handle spatio-temporal problems. In this case, the data are
 654 sampled in a spatio-temporal domain $\Omega \times T$, where T is a finite time interval of interest.
 655 Bernardi et al. [2017] and Arnone et al. [2019] considers two different estimation
 656 functionals: the former involves two penalty terms, to regulate the smoothness of the

657 spatio-temporal field in space and in time, the latter entails a unique penalty that
 658 involves a parabolic PDE. The spatio-temporal problem is discretized using finite
 659 elements in space and either splines or finite differences in time; it is thus reduced
 660 to a system having the same structure as (2.5), but with larger blocks, that involve
 661 the bases in both space and in time [see, e.g. Sangalli, 2021, Arnone et al., 2021].
 662 Thanks to the similar structure of the problem system, it is possible also in this case
 663 to resort to SMW decomposition, as well as to stochastic GCV approximation. These
 664 strategies are indeed of crucial importance in this setting, as the dimension of the
 665 space-time estimation problem may be very large.

666 **9. Conclusions.** In this work, we investigated the computational tractability of
 667 SR-PDE. An appropriate use of SMW identity permits us to significantly decrease the
 668 time and memory consumption required to solve an SR-PDE problem. The simulation
 669 study of Section 4 displays a computational cost between $\mathcal{O}(N)$ and $\mathcal{O}(N^2)$, where N
 670 is the number of mesh nodes, with a gain of one magnitude order over the standard
 671 sparse LU solution of the estimation problem.

672 Particular attention has also been devoted to the automated selection of the
 673 smoothness parameter, via minimization of the *GCV* criterion. The selection of an
 674 appropriate value of the smoothing parameter is in fact a crucial aspect of the method-
 675 ology but is computationally highly demanding. The simulation study in Section 6
 676 shows a significant reduction in the computational cost of selecting the smoothing
 677 parameter when using the proposed SMW-based stochastic trace estimation tech-
 678 nique, to evaluate the *edf* needed for computation of the *GCV* index. Stochastic *edf*
 679 estimation is also shown to scale approximately linearly with N .

680 Finally, this work has focused on a basic formulation of SR-PDE estimation, but
 681 we briefly outlined in Section 7 the broad applicability of this class of models, which
 682 have proven to be highly valuable for dealing with a variety of complex estimation
 683 problems.

684 **Acknowledgments.** This work is based on preliminary explorations by Clelia
 685 Bambini and Luca Giussani. The authors also thank Michelle Carey for discussions
 686 on these topics. C.d.F. and L.F. have been partially funded by the Italian Research
 687 Center on High-Performance Computing, Big Data and Quantum Computing (ICSC),
 688 European Union - Next Generation EU. The present research is within the framework
 689 of MUR grant Dipartimento di Eccellenza 2023-2027, Dipartimento di Matematica,
 690 Politenico di Milano.

691 **References.**

- 692 P. R. Amestoy, T. A. Davis, and I. S. Duff. Algorithm 837: Amd, an approximate
 693 minimum degree ordering algorithm. *ACM Trans. Math. Softw.*, 30(3):381–388,
 694 sep 2004. ISSN 0098-3500. doi: 10.1145/1024074.1024081. URL [https://doi.org/
 695 10.1145/1024074.1024081](https://doi.org/10.1145/1024074.1024081).
- 696 E. Arnone, L. Azzimonti, F. Nobile, and L. M. Sangalli. Modeling spatially dependent
 697 functional data via regression with differential regularization. *J. Multivariate Anal.*,
 698 170:275–295, 2019.
- 699 E. Arnone, L. M. Sangalli, and A. Vicini. Smoothing spatio-temporal data with
 700 complex missing data patterns. *Statistical Modelling*, page 1471082X211057959,
 701 2021.
- 702 E. Arnone, A. Clemente, L. M. Sangalli, E. Lila, J. Ramsay, and L. Formaggia.
 703 *fdaPDE: Physics-Informed Spatial and Functional Data Analysis*, 2022. URL [https:
 704 //CRAN.R-project.org/package=fdaPDE](https://CRAN.R-project.org/package=fdaPDE). R package version 1.1-16.

- 705 L. Azzimonti, F. Nobile, L. M. Sangalli, and P. Secchi. Mixed Finite Elements for
 706 spatial regression with PDE penalization. *SIAM/ASA J. Uncertain. Quantif.*, 2(1):
 707 305–335, 2014.
- 708 V. Baramidze, M.-J. Lai, and C. Shum. Spherical splines for data interpolation and
 709 fitting. *SIAM Journal on Scientific Computing*, 28(1):241–259, 2006. ISSN 1064-
 710 8275. doi: 10.1137/040620722.
- 711 D. M. Bates and G. Wahba. A truncated singular value decomposition and other
 712 methods for generalized cross-validation. *Tech. Report 715, Department of Statis-*
 713 *tics, University of Wisconsin*, 1983.
- 714 M. S. Bernardi, L. M. Sangalli, G. Mazza, and J. O. Ramsay. A penalized regression
 715 model for spatial functional data with application to the analysis of the production
 716 of waste in Venice province. *Stochastic environmental research and risk assessment*,
 717 31(1):23–38, 2017.
- 718 M. Chung, J. Hanson, and S. Pollak. Statistical analysis on brain surfaces. *Technical*
 719 *report, University of WisconsinMadison*, 2014.
- 720 P. Craven and G. Wahba. Smoothing noisy data with spline functions: estimating the
 721 correct degree of smoothing by the method of generalized cross-validation. *Numer.*
 722 *Math.*, 31:377–403, 1978/79.
- 723 T. A. Davis. Algorithm 832: Umfpack v4.3—an unsymmetric-pattern multifrontal
 724 method. *ACM Trans. Math. Softw.*, 30(2):196–199, jun 2004. ISSN 0098-3500. doi:
 725 10.1145/992200.992206. URL <https://doi.org/10.1145/992200.992206>.
- 726 J. W. Eaton, D. Bateman, S. Hauberg, and R. Wehbring. *GNU Octave version 7.3.0*
 727 *manual: a high-level interactive language for numerical computations*, 2022. URL
 728 <https://www.gnu.org/software/octave/doc/v7.3.0/>.
- 729 B. Ettinger, S. Perotto, and L. M. Sangalli. Spatial regression models over two-
 730 dimensional manifolds. *Biometrika*, 103(1):71–88, 2016.
- 731 R. L. Eubank. *Nonparametric Regression and Spline Smoothing*. CRC Press, 2nd
 732 Edition, 1999.
- 733 R. L. Eubank, C. Huang, Y. M. Maldonado, N. Wang, S. Wang, and R. J. Buchanan.
 734 Smoothing spline estimation in varying-coefficient models. *J. R. Stat. Soc. Ser. B.*
 735 *Stat. Methodol.*, 66(3):653–667, 2004.
- 736 F. Ferraty and P. Vieu. *Nonparametric functional data analysis: theory and practice*,
 737 volume 76. Springer, 2006.
- 738 R. A. Fisher. Frequency distribution of the values of the correlation coefficient in
 739 samples from an indefinitely large population. *Biometrika*, 10(4):507–521, 1915.
- 740 D. A. Girard. A fast ‘Monte-Carlo cross-validation’ procedure for large least squares
 741 problems. *Numer. Math.*, 56(1):1–23, 1989.
- 742 M. F. Glasser, S. N. Sotiropoulos, J. A. Wilson, T. S. Coalson, B. Fischl, J. L. Ander-
 743 sson, J. Xu, S. Jbabdi, M. Webster, J. R. Polimeni, D. C. Van Essen, M. Jenkinson,
 744 and WU-Minn HCP Consortium. The minimal preprocessing pipelines for the hu-
 745 man connectome project. *NeuroImage*, 80:105–124, 2013.
- 746 G. Golub, M. Heath, and G. Wahba. Generalized cross-validation as a method for
 747 choosing a good ridge parameter. *Technometrics*, 21(2):215–223, 1979.
- 748 G. H. Golub and U. von Matt. Generalized cross-validation for large-scale problems.
 749 *J. Comput. Graph. Statist.*, 6(1):1–34, 1997.
- 750 P. J. Green and B. W. Silverman. *Nonparametric Regression and Generalized Linear*
 751 *Models*. Chapman and Hall/CRC, 1st Edition, 1993.
- 752 P. J. Green and B. W. Silverman. *Nonparametric regression and generalized linear*
 753 *models*, volume 58 of *Monographs on Statistics and Applied Probability*. Chapman
 754 & Hall, London, 1994. ISBN 0-412-30040-0. doi: 10.1007/978-1-4899-4473-3. URL

- 755 <http://dx.doi.org/10.1007/978-1-4899-4473-3>. A roughness penalty approach.
756 C. Gu. Cross-validating non-Gaussian data. *J. Comput. Graph. Statist*, 1(2):169–179,
757 1992.
- 758 T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman Hall, 1990.
- 759 M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for
760 Laplacian smoothing splines. *Commun. Stat. Simul. Comput.*, 18(3):1059–1076,
761 1989.
- 762 M. F. Hutchinson and F. R. de Hoog. Smoothing noisy data with spline functions.
763 *Numer. Math.*, 47(1):99–106, 1985.
- 764 M. F. Hutchinson and F. R. de Hoog. An efficient method for calculating smoothing
765 splines using orthogonal transformations. *Numer. Math.*, 50:311–319, 1986/87.
- 766 P. Kokoszka and M. Reimherr. *Introduction to functional data analysis*. Chapman
767 and Hall/CRC, 2017.
- 768 S. Konishi and G. Kitagawa. *Information Criteria and Statistical Modeling*. Springer
769 Publishing Company, 2007.
- 770 M.-J. Lai and L. L. Schumaker. *Spline functions on triangulations*, volume 110
771 of *Encyclopedia of Mathematics and its Applications*. Cambridge University
772 Press, Cambridge, 2007. ISBN 978-0-521-87592-9; 0-521-87592-7. doi: 10.1017/
773 CBO9780511721588. URL <http://dx.doi.org/10.1017/CBO9780511721588>.
- 774 M.-J. Lai and L. Wang. Bivariate penalized splines for regression. *Statistica Sinica*,
775 23(3):1399–1417, 2013. ISSN 1017-0405.
- 776 S.-H. Lai and B. C. Vemuri. Sherman-Morrison-Woodbury-formula-based algorithms
777 for the surface smoothing problem. *Linear Algebra Appl.*, 265(1-3):203–229, 1997.
- 778 E. Lila, J. A. D. Aston, and L. M. Sangalli. Smooth principal component analysis
779 over two-dimensional manifolds with an application to neuroimaging. *Ann. Appl.*
780 *Stat.*, 10(4):1854–1879, 2016.
- 781 R Core Team. *R: A Language and Environment for Statistical Computing*. R
782 Foundation for Statistical Computing, Vienna, Austria, 2021. URL [https://www.](https://www.R-project.org/)
783 [R-project.org/](https://www.R-project.org/).
- 784 J. O. Ramsay and B. W. Silverman. Functional data analysis. *Internet Adresi: http*,
785 2008.
- 786 D. Ruppert, M. P. Wand, and R. J. Carroll. *Semiparametric regression*. Number 12.
787 Cambridge university press, 2003.
- 788 Y. Saad. *Iterative Methods for Sparse Linear Systems*. Society for Industrial and
789 Applied Mathematics, second edition, 2003. doi: 10.1137/1.9780898718003. URL
790 <https://epubs.siam.org/doi/abs/10.1137/1.9780898718003>.
- 791 Y. Saad and M. H. Schultz. Gmres: A generalized minimal residual algorithm for
792 solving nonsymmetric linear systems. *SIAM Journal on Scientific and Statistical*
793 *Computing*, 7(3):856–869, 1986. doi: 10.1137/0907058.
- 794 L. M. Sangalli. Spatial regression with partial differential equation regularization.
795 *International Statistical Review*, 89(3):505–531, 2021. doi: 10.1111/insr.12444.
- 796 L. Sario, M. Nakai, C. Wang, and L. O. Chung. *Classification Theory of Riemannian*
797 *Manifolds*. Springer-Verlag Berlin Heidelberg, 1st ed., 1977.
- 798 J. Sherman and W. Morrison. Adjustment of an inverse matrix corresponding to a
799 change in one element of a given matrix. *Ann. Math. Stat.*, 21(1):124–127, 1950.
- 800 F. Utreras. Optimal smoothing of noisy data using spline functions. *SIAM J. Sci.*
801 *and Stat. Comput.*, 2(3):349–362, 1981.
- 802 H. A. van der Vorst. Bi-cgstab: A fast and smoothly converging variant of bi-cg
803 for the solution of nonsymmetric linear systems. *SIAM Journal on Scientific and*
804 *Statistical Computing*, 13(2):631–644, 1992. doi: 10.1137/0913035.

- 805 G. Wahba. *Spline models for observational data*, volume 59 of *CBMS-NSF Re-*
806 *gional Conference Series in Applied Mathematics*. Society for Industrial and Ap-
807 *plied Mathematics (SIAM)*, Philadelphia, PA, 1990. ISBN 0-89871-244-0. doi:
808 [10.1137/1.9781611970128](https://doi.org/10.1137/1.9781611970128). URL <http://dx.doi.org/10.1137/1.9781611970128>.
- 809 L. Wang, G. Wang, M.-J. Lai, and L. Gao. Efficient estimation of partially linear
810 models for spatial data over complex domains. *Statistica Sinica*, 30:347–369, 2020.
- 811 Y. Wang. *Smoothing splines: methods and applications*. Chapman and Hall/CRC,
812 2019.
- 813 M. Wilhelm and L. M. Sangalli. Generalized Spatial Regression with Differential
814 Regularization. *J. Stat. Comput. Simul.*, 86(13):2497–2518, 2016.
- 815 S. N. Wood. Modelling and smoothing parameter estimation with multiple quadratic
816 penalties. *J. R. Stat. Soc. Ser. B. Stat. Methodol.*, 62:413–428, 2000.
- 817 S. N. Wood. Stable and efficient multiple smoothing parameter estimation for gener-
818 alized additive models. *J. Am. Statist. Ass.*, 99:673–686, 2004.
- 819 S. N. Wood. *Generalized additive models*. Texts in Statistical Science Series. CRC
820 Press, Boca Raton, FL, 2017. ISBN 978-1-4987-2833-1. An introduction with R.
- 821 S. N. Wood, M. V. Bravington, and S. L. Hedley. Soap film smoothing. *Journal of the*
822 *Royal Statistical Society: Series B (Statistical Methodology)*, 70(5):931–955, 2008.
- 823 M. A. Woodbury. *Inverting modified matrices*. Memorandum Report 42, Statistical
824 Research Group, Princeton University, Princeton, NJ, 1950.

MOX Technical Reports, last issues

Dipartimento di Matematica
Politecnico di Milano, Via Bonardi 9 - 20133 Milano (Italy)

- 84/2023** Arnone, E.; De Falco, C.; Formaggia, L.; Meretti, G.; Sangalli, L.M.
Computationally efficient techniques for Spatial Regression with Differential Regularization
- 83/2023** Cavinato, L.; Massi, M.C.; Sollini, M.; Kirienko, M.; Ieva, F.
Dual adversarial deconfounding autoencoder for joint batch-effects removal from multi-center and multi-scanner radiomics data
- 81/2023** Buchwald, S.; Ciaramella, G.; Salomon, J.; Sugny, D.
A SPIRED code for the reconstruction of spin distribution
- 82/2023** Pozzi, G.; Ciarletta, P.
Geometric control by active mechanics of epithelial gap closure
- 79/2023** Agosti, A.; Bardin, R.; Ciarletta, P.; Grasselli, M.
A diffuse interface model of tumour evolution under a finite elastic confinement
- 78/2023** Antonietti, P.F.; Bonizzoni, F.; Corti, M.; Dall'Olio, A.
Discontinuous Galerkin for the heterodimer model of prion dynamics in Parkinson's disease
- 77/2023** Fumagalli, I.; Corti, M.; Parolini, N.; Antonietti, P. F.
Polytopal discontinuous Galerkin discretization of brain multiphysics flow dynamics
- 76/2023** Ieva, F.; Galliani, G.; Secchi, P.
The impact of public transport on the diffusion of COVID-19 pandemic in Lombardy during 2020
- 74/2023** Pidò, S.; Pinoli, P.; Crovari, P.; Ieva, F.; Garzotto, F.; Ceri, S.
Ask Your Data—Supporting Data Science Processes by Combining AutoML and Conversational Interfaces
- 75/2023** Archetti, A.; Ieva, F.; Matteucci, M.
Scaling survival analysis in healthcare with federated survival forests: A comparative study on heart failure and breast cancer genomics