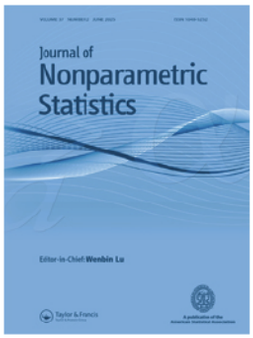

Journal of Nonparametric Statistics

ISSN: 1048-5252 (Print) 1029-0311 (Online) Journal homepage: www.tandfonline.com/journals/gnst20

TSSS: a novel triangulated spherical spline smoothing for surface-based data

Zhiling Gu, Shan Yu, Guannan Wang, Ming-Jun Lai & Lily Wang

To cite this article: Zhiling Gu, Shan Yu, Guannan Wang, Ming-Jun Lai & Lily Wang (07 Jan 2025): TSSS: a novel triangulated spherical spline smoothing for surface-based data, Journal of Nonparametric Statistics, DOI: 10.1080/10485252.2025.2449886

To link to this article: https://doi.org/10.1080/10485252.2025.2449886

JOURNAL OF NONPARAMETRIC STATISTICS
https://doi.org/10.1080/10485252.2025.2449886

TSSS: a novel triangulated spherical spline smoothing for surface-based data

Zhiling Gu[a,b], Shan Yu[c], Guannan Wang[d], Ming-Jun Lai [e] and Lily Wang [f]

[a]Department of Biostatistics, Yale University, New Haven, CT, USA; [b]Department of Statistics, Iowa State University, Ames, IA, USA; [c]Department of Statistics, University of Virginia, Charlottesville, VA, USA; [d]Department of Mathematics, William & Mary, Williamsburg, VA, USA; [e]Department of Mathematics, The University of Georgia, Athens, GA, USA; [f]Department of Statistics, George Mason University, Fairfax, VA, USA

ABSTRACT

Surface-based data are prevalent across diverse practical applications in various fields. This paper introduces a novel nonparametric method to discover the underlying signals from data distributed on complex surface-based domains. The proposed approach involves a penalised spline estimator defined on a triangulation of surface patches, enabling effective signal extraction and recovery. The proposed method offers superior handling of 'leakage' or 'boundary effects' over complex domains, enhanced computational efficiency, and capabilities for analyzing sparse and irregularly distributed data on complex objects. We provide rigorous theoretical guarantees, including convergence rates and asymptotic normality of the estimators. We demonstrate that the convergence rates are optimal within the framework of nonparametric estimation. A bootstrap method is introduced to quantify the uncertainty in the proposed estimators and to provide pointwise confidence intervals. The advantages of the proposed method are demonstrated through simulations and data applications on cortical surface neuroimaging data and oceanic near-surface atmospheric data.

ARTICLE HISTORY
Received 7 May 2024
Accepted 29 December 2024

KEYWORDS
Complex domain; nonparametric smoothing; sphere-like surface; spherical splines; triangulation

AMS SUBJECT CLASSIFICATIONS
62G05; 62G08; 62G20

1. Introduction

Surface-based data are widely observed in various fields, and extracting useful information from data distributed on surfaces is of great significance. For example, in planetary science, scientists are interested in the movement of tectonic plates (Chang et al. 2000; Marzio et al. 2019) observed on the surfaces of celestial bodies. In cosmology, completing the missing data and correcting the noisy observations for cosmic microwave background radiation are problems of interest (Abrial et al. 2008). In meteorology, hourly surface-based data are the most used and requested type of climatology data, and efforts have been made to integrate data from different stations in various repositories (Smith et al. 2011). Additional examples include the recovery of high-resolution time series of aerosol optical

CONTACT Lily Wang lwang41@gmu.edu Department of Statistics, George Mason University, Fairfax, VA 22030, USA

Supplemental data for this article can be accessed online at http://dx.doi.org/10.1080/10485252.2025.2449886.

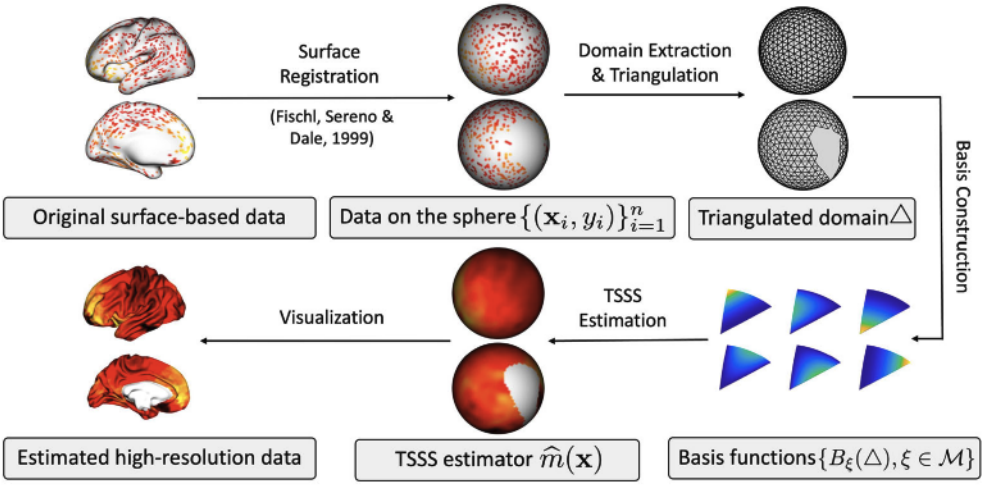

Figure 1. Workflow of the TSSS estimation method for brain surface data.

depth (Zhang, Zhou, Zhao, et al. 2022) and total electron content (Sun et al. 2022) from surface-based measurements.

Furthermore, in biomedical research, efforts have been made to explore the functional properties of proteins using molecular surface data (Kinoshita and Nakamura 2003). Another active area of investigation is cortical and cortical surface functional magnetic resonance imaging (fMRI) in neuroimaging. This technique offers the advantage of recovering anatomical structures with higher precision compared to volumetric fMRI methods (Brodoehl et al. 2020; Lila and Aston 2020; Mejia et al. 2020; Cole et al. 2021; Zhang, Wu, et al. 2023). To analyze data distributed on general surfaces, it is common to employ mappings that project the data onto a unit sphere (Fischl et al. 1999; Gu et al. 2004), as depicted in Figure 1.

Spherical harmonics (SH) (Seeley 1966; Abrial et al. 2008) are widely used to model and denoise spherical data, interpolating functions on the whole sphere. Thin plate splines on the sphere (TPSOS) (Wahba 1981) were later proposed as a special case of SH while accounting for the smoothness of functions. However, SH and TPSOS may lack the flexibility to capture complex signals. SH relies on a fixed set of basis functions that span the entire sphere, requiring many functions for highly localised or rapidly changing signals, which increases computational complexity. Similarly, TPSOS, while incorporating smoothness, faces difficulties with complex functions due to its reliance on polynomial functions of trigonometric terms, limiting its flexibility.

Another popular method is the kernel-based approach (Cao et al. 2013); however, it becomes computationally intensive for large datasets primarily due to the computation and storage requirements of kernel matrices and the need to solve optimisation problems involving these matrices. The size of the kernel matrix is proportional to the number of data points, which makes the computation and storage requirements overgrow with the size of the dataset. The limitations of existing methods motivate the need for alternative methods that can offer enhanced flexibility in capturing the complexity of the underlying signal for spherical data.

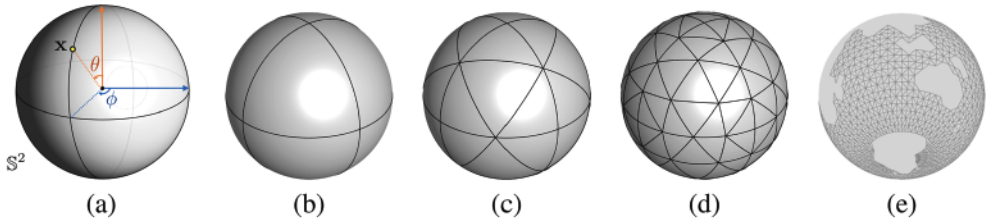

Figure 2. Illustrations of (a) colatitude θ and longitude ϕ, (b)–(d) spherical triangulations with varying levels of refinements, and (e) triangulation of the ocean surface. The triangulation refinement is achieved by connecting the centre of each edge with the spherical edge with the shortest geodesic length.

In addition, data are commonly collected on complex patches/regions of surfaces with irregular boundaries or holes, such as the ocean domain shown in Figure 2(e). Conventional tools mentioned above for analyzing spherical data suffer from the problem of 'leakage' across irregular-shaped domains due to inappropriate linking of parts of the domain (Ramsay 2002; Wood et al. 2008). Motivated by recent works, such as Lai and Wang (2013) and Wang, Wang, et al. (2020), Bernstein–Bézier polynomials defined on triangles are useful tools over planar domains with irregular shapes. Given the similarity and common properties between classical polynomial splines over planar triangulation and the spherical setting, spherical Bernstein-Bézier splines (Alfeld et al. 1996a, 1996b) could be well suited for spherical data interpolation and approximation problems for complex sphere-like surfaces. Their efficacy in geopotential reconstruction has been demonstrated in Baramidze and Lai (2004), Baramidze et al. (2006), Lai et al. (2008), and Baramidze and Lai (2011). In this paper, we introduce a novel application of spherical splines specifically tailored to analyze data distributed on complex surface-based domains, such as surface patches with complex boundaries. This triangulated spherical spline smoothing (TSSS) approach enables the extraction of underlying signals from intricate data structures on surface patches in the presence of noise, providing a robust and computationally efficient method for accurately estimating and modelling complex surface patterns.

Importantly, we investigate the statistical properties of the proposed TSSS method. Specifically, we establish the convergence rate of the TSSS estimator, which is governed by the fineness of the triangle mesh, the degree and smoothness of the spherical spline, the penalty parameter, and the smoothness of the mean function. We further provide the conditions to achieve the optimal nonparametric convergence rate (Stone 1982). We also derive the asymptotic normality for the TSSS estimator. However, due to the complexity of the spline basis functions, obtaining the exact form of the standard error can be challenging. Therefore, we propose a wild-bootstrap-based method to estimate the standard error.

In addition to theoretical contributions, the proposed TSSS method overcomes the limitations of existing methods and offers several advantages. Firstly, TSSS effectively addresses the problem of 'leakage' across complex domains by utilising information from neighbouring triangles to accurately denoise or deblur data while preserving geometric features and spatial structures. This makes it ideal for analyzing data on complex patches or regions of surfaces with irregular boundaries or holes. Secondly, TSSS employs compactly supported basis functions and a sparse or roughness penalty, similar to P-splines (Eilers and

Marx 2021) and bivariate penalised splines on triangulation (BST; Lai and Wang 2013; Yu et al. 2020; Wang et al. 2023). This enables the model to capture intricate patterns in the data without becoming overly complex, mitigate boundary effects, and facilitate robust smoothing of unevenly distributed data, such as sparse and irregular data on surface-based complex domains. In addition, TSSS offers computational efficiency as a global estimation method with an explicit model expression. Its efficiency stems from the need to solve only a single linear system, which reduces the computational complexity and makes it suitable for handling large datasets.

The remainder of this paper is organised as follows. Section 2 introduces the triangulation of a sphere and the spline space defined on the triangulation. In Section 3, we present the asymptotic properties of the proposed TSSS estimator, including convergence rates and asymptotic normality. In addition, we introduce a bootstrap method to effectively quantify the uncertainty associated with the TSSS estimators and to provide pointwise confidence intervals (PCIs). Section 4 outlines the implementation details of the TSSS method, including the selection of triangulation, spline basis, and penalty parameters. In Section 5, we present simulation studies to evaluate the finite sample performance of the TSSS estimator for functions observed on the grid of the whole sphere and functions observed on a complex spherical domain, and the uncertainty quantification of the TSSS estimators. In Section 6, TSSS is applied to cortical surface fMRI data and near-surface ocean-atmospheric data. Section 7 summarises the main contributions of this paper and concludes with some remarks. Technical details are provided in the Supplementary Materials.

2. Penalised spline estimators on triangulated spheres

Consider a set of observations $\{(\theta_i, \phi_i, Y_i)\}_{i=1}^n$, where $Y_i \in \mathbb{R}$ represents the response variable, $\theta_i \in [0, \pi]$ and $\phi_i \in [0, 2\pi)$ denote the corresponding colatitude and longitude as illustrated in Figure 2(a). It is worth pointing out that a sphere $\mathbb{S}^2$ is a 2-dimensional (2D) object, which, for convenience of analysis, is embedded in a 3-dimensional (3D) Euclidean space $\mathbb{R}^3$. Consequently, it is equivalent to considering any point on $\mathbb{S}^2$ in the following form $\mathbf{X}_i = (X_{i1}, X_{i2}, X_{i3})^\top \in \mathbb{R}^3$, where $\|\mathbf{X}_i\|_2^2 = X_{i1}^2 + X_{i2}^2 + X_{i3}^2 = 1$. Suppose $\{(\mathbf{X}_i, Y_i), \mathbf{X}_i \in \Omega\}_{i=1}^n$ is an i.i.d. sample of size n observed on a domain Ω. Here, Ω can be the entire unit spherical domain $\mathbb{S}^2$ or a specific spherical patch or region within $\mathbb{S}^2$. We assume the sample is drawn from the distribution of $(\mathbf{X}, Y)$, $\mathbf{X} \in \Omega$, through

$$Y_i = m(\mathbf{X}_i) + \sigma(\mathbf{X}_i)\epsilon_i, \quad i = 1, 2, \ldots, n, \tag{1}$$

where $m(\cdot)$ and $\sigma(\cdot)$ are the conditional mean and standard deviation functions, ϵ_i's are i.i.d random errors with mean $\mathrm{E}(\epsilon_i) = 0$, and variance $\mathrm{Var}(\epsilon_i) = 1$. In addition, we assume that ϵ_i and $\mathbf{X}_i$ are independent for all i. The problem of interest is to estimate the unknown mean function $m(\mathbf{x})$ given observations $\{(\mathbf{X}_i, Y_i), \mathbf{X}_i \in \Omega\}_{i=1}^n$.

2.1. Triangulation and spherical Bernstein–Bézier polynomials

Triangulation has been widely used for irregular 3D, spherical and planar domains due to its ability to effectively approximate complex geometries and simplify computational tasks (Lai and Schumaker 2007; Mark et al. 2008; De Loera et al. 2010). For domain $\Omega \subseteq \mathbb{S}^2$, a

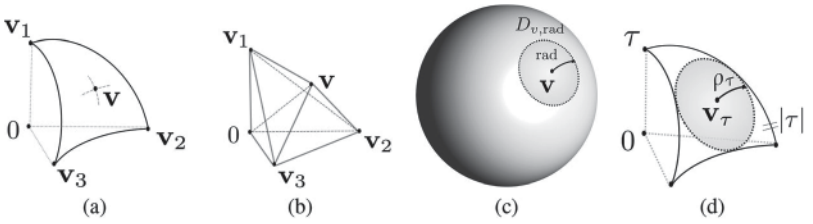

Figure 3. Illustrations of (a) a spherical triangle τ, (b) tetrahedrons related to $\mathbf{v}$, (c) a spherical cap $D_{v,\mathrm{rad}}$, and (d) an inscribed cap of spherical triangle τ.

triangulation $\triangle$ is defined as a collection of N spherical triangles $\triangle = \{\tau_1, \ldots, \tau_N\}$, such that $\mathbf{\Omega} = \cup_{i=1}^{N} \tau_i$ (Lai and Schumaker 2007). Each spherical triangle $\tau := \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$ comprises the set of points in $\mathbf{\Omega}$ that lie within the region bounded by three circular arcs $\langle \mathbf{v}_i, \mathbf{v}_{i+1} \rangle, i = 1, 2, 3$. Here, $\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3$ are the three vertices of τ, and $\mathbf{v}_4$ is identified as $\mathbf{v}_1$. These spherical triangles either share an edge (circular arc), share a vertex, or do not intersect each other. Figure 2 provides an illustration of spherical triangulations.

Further, we define the spherical barycentric coordinates $b_1(\mathbf{v}), b_2(\mathbf{v}), b_3(\mathbf{v})$ for any point $\mathbf{v} \in \mathbb{S}^2$ relative to a nondegenerate spherical triangle $\tau = \langle \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$, such that $\mathbf{v} = b_1(\mathbf{v})\mathbf{v}_1 + b_2(\mathbf{v})\mathbf{v}_2 + b_3(\mathbf{v})\mathbf{v}_3$. Note that $b_\kappa(\mathbf{v}) = \mathrm{vol}(\tau_\kappa)/\mathrm{vol}(\tau)$ is unique, where τ_κ is tetrahedron $\langle \mathbf{0}, \mathbf{v}, \mathbf{v}_{\kappa+1}, \mathbf{v}_{\kappa+2} \rangle$, and τ is tetrahedron $\langle \mathbf{0}, \mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3 \rangle$. Here, $\mathbf{0}$ represents the centre of a sphere; see Figure 3. This holds for $\kappa = 1, 2, 3$, where $\mathbf{v}_2 = \mathbf{v}_5$.

For integers $d > r > 0$, let $C^r(\mathbf{\Omega})$ denote the space of rth continuously differentiable functions, and $\mathcal{H}_d$ represent the space of homogeneous trivariate polynomials of degree d given in Alfeld et al. (1996a). Following Baramidze et al. (2006) and Lai et al. (2008), we define *space of homogeneous spherical splines* as $\mathcal{S}_d^r(\triangle) = \{s \in C^r(\mathbf{\Omega}), s|_\tau \in \mathcal{H}_d, \tau \in \triangle\}$, where $s|_\tau$ refers to the polynomial piece of spline s restricted on triangle τ. Using spherical barycentric coordinates, we can construct a basis for $\mathcal{H}_d$, called *spherical Bernstein-Bézier (SBB) polynomials*: $B_{ijk}^\tau(\mathbf{x}) = \frac{d!}{i!j!k!} b_1(\mathbf{x})^i b_2(\mathbf{x})^j b_3(\mathbf{x})^k$, $i+j+k = d$, $\tau \in \triangle$. Consequently, any $s \in \mathcal{H}_d$ can be expressed as $s(\mathbf{x}) = \sum_{\tau \in \triangle} \sum_{i+j+k=d} \gamma_{ijk}^\tau B_{ijk}^\tau(\mathbf{x})$, where γ_{ijk}^τ's are referred to as *B-coefficients*. Since γ_{ijk}^τ's are linearly independent, the dimension of $\mathcal{H}_d(\tau)$ is $(d+2)(d+1)/2$, as seen in Definition 13.17 of Lai and Schumaker (2007). Moreover, a stable local basis $B_\xi, \xi \in \mathcal{M}$ can be constructed for $\mathcal{S}_d^r(\triangle)$, with $\mathcal{M}$ being the index set of the SBB bases. We refer to the Supplementary Material Section S2.3 for detailed illustrations.

To describe the local properties of a domain on the sphere, we refer to the concept of 'spherical cap' as introduced in Neamtu and Schumaker (2004). A spherical cap of radius rad associated with a point $\mathbf{v} \in \mathbb{S}^2$ is defined as the set of all points that have a geodesic distance to $\mathbf{v}$ at most rad. For a spherical triangle $\tau \in \triangle$, the *inscribed cap* the largest spherical cap contained within τ, with its centre $\mathbf{v}_\tau$ referred to as the *incenter* of τ, and its radius ρ_τ denoted as the *inradius* of τ. The longest spherical edge of τ is represented by $|\tau|$; see Figure 3. Denote by $|\triangle| = \max_{\tau \in \triangle} |\tau|$ the triangulation size corresponding to the longest spherical edge of all triangles in $\triangle$. Let $\rho_\triangle$ denote the smallest inradius in $\triangle$, i.e. and $\rho_\triangle = \min_{\tau \in \triangle} \rho_\tau$. In addition, we define A_τ as the area of the spherical triangle τ. To ensure that all triangles in $\triangle$ have comparable sizes, we assume $\triangle$ satisfies Assumption (C2) described in Section 3 below.

2.2. Penalised spline estimators

In this section, we present a penalised spline estimator of the regression function $m(\cdot)$ in the model (1). To construct the penalised spline estimator, we first quantify the roughness of $f : \Omega \to \mathbb{R}$. Let $D^{\alpha} f := D^{i}_{x_1} D^{j}_{x_2} D^{k}_{x_3} f$ be a partial derivative of degree $|\alpha| = i + j + k$ for f. The homogeneous extension of $f : \Omega \to \mathbb{R}$ of degree p, denoted as $f_p : \mathbb{R}^3 \to \mathbb{R}$, is defined as

$$f_p(\mathbf{x}) := \|\mathbf{x}\|^p f(\mathbf{x}/\|\mathbf{x}\|), \quad \mathbf{x} \in \mathbb{R}^3 \backslash \{\mathbf{0}\} \tag{2}$$

for any integer p, where $\|\mathbf{x}\|$ is the Euclidean norm of $\mathbf{x}$. We define energy functional of f as

$$\mathcal{E}(f) = \sum_{|\alpha|=2} \int_{\Omega} |D^{\alpha} f_p|^2 \mathrm{d}\mu, \tag{3}$$

where μ is Lebesgue measure on $\mathbb{S}^2$ and the extended function f_p is restricted to spherical domain Ω for integration. It is worth noting that any fixed value of p can be chosen as the degree of extension. In our implementation, following Lai et al. (2008), we set $p = 1$ for odd d and $p = 0$ for even d.

We propose the following *Triangulated Spherical Spline Smoothing (TSSS)* estimator for m, defined as the minimiser of the following objective function:

$$\widehat{m}_{\lambda} = \arg\min_{s \in \mathcal{S}^r_d(\triangle)} \sum_{i=1}^{n} \{Y_i - s(\mathbf{X}_i)\}^2 + \lambda \mathcal{E}(s), \tag{4}$$

where $\mathcal{S}^r_d(\triangle)$ is the space of homogeneous spherical splines defined in Section 2.1. The parameter λ is a nonnegative tuning parameter that controls the trade-off between fit and smoothness. Penalising the energy functional with $\lambda \mathcal{E}(s)$ for some $\lambda > 0$ reduces the roughness of the estimator, since higher values of $\mathcal{E}(s)$ indicate less smoothness.

Note that every function $s \in \mathcal{S}^r_d(\triangle)$ defined on a spherical triangle τ can be expressed as a linear combination of the SBB basis functions $\mathbf{B}^{\tau}(\mathbf{x}) = (B^{\tau}_{\xi}(\mathbf{x}), \xi \in \mathcal{M})^{\top}$ using the coefficients $\boldsymbol{\gamma}^{\tau} = (\gamma^{\tau}_{\xi}, \xi \in \mathcal{M})^{\top}$. Specifically, for any point $\mathbf{x} \in \tau$, we have $s(\mathbf{x}) = \boldsymbol{\gamma}^{\tau\top} \mathbf{B}^{\tau}(\mathbf{x})$. Then, the energy functional $\mathcal{E}(s)$ in (3) can be rewritten as

$$\mathcal{E}(s) = \sum_{\tau \in \triangle} \int_{\tau} \left\{ \diamondsuit(\boldsymbol{\gamma}^{\tau\top} \mathrm{B}^{\tau}(\mathbf{x})) \right\}^{\otimes 2} \mathrm{d}\mu = \sum_{\tau \in \triangle} \int_{\tau} \boldsymbol{\gamma}^{\tau\top} \left\{ \diamondsuit(\mathrm{B}^{\tau}(\mathbf{x})) \right\}^{\otimes 2} \boldsymbol{\gamma}^{\tau} \mathrm{d}\mu, \tag{5}$$

where $\diamondsuit f := (D^{\alpha} f_p, |\alpha| = 2)$ is the second order derivative vector of $f \in \mathcal{S}$ given in (2), $\mathbf{x}^{\otimes 2} = \mathbf{x}\mathbf{x}^{\top}$.

Since the spline coefficients of $s \in \mathcal{C}^r(\triangle)$ need to satisfy some smoothness conditions across each interior face of $\triangle$, a smoothness constraint matrix $\mathbf{M}$ can be used to impose the required smoothness. Specifically, $\mathbf{M}\boldsymbol{\gamma} = \mathbf{0}$ is imposed as a smoothness condition to ensure that the estimated function is rth differentiable at all edges of $\triangle$. In doing so, the smoothness of the TSSS estimator can be controlled and achieved for different values of r. For the spline function $\mathbf{B}(\mathbf{x})^{\top} \boldsymbol{\gamma}$ we have $\mathcal{E}(\mathbf{B}^{\top} \boldsymbol{\gamma}) = \boldsymbol{\gamma}^{\top} \mathbf{P} \boldsymbol{\gamma}$, where $\mathbf{P}$ is a $\{N(d+1)(d+2)/2\} \times \{N(d+1)(d+2)/2\}$ block diagonal penalty matrix and can be obtained from (5).

Thus, one can estimate the spline coefficients via minimising

$$\widehat{\boldsymbol{\gamma}} = \arg\min_{\boldsymbol{\gamma}} \sum_{i=1}^{n} \{Y_i - \mathbf{B}(\mathbf{X}_i)^\top \boldsymbol{\gamma}\}^2 + \lambda \boldsymbol{\gamma}^\top \mathbf{P} \boldsymbol{\gamma}, \text{ subject to } \mathbf{M}\boldsymbol{\gamma} = \mathbf{0}.$$

Then, we can obtain the TSSS estimator by $\widehat{m}_\lambda(\mathbf{x}) = \mathbf{B}^\top(\mathbf{x})\widehat{\boldsymbol{\gamma}}$.

3. Theoretical results

To discuss the asymptotics of the TSSS estimator, we first introduce some technical notation. Let X_n, $n \geq 1$, be a sequence of random variables. We denote $X_n = O_p(a_n)$ if $\lim_{c\to\infty} \limsup_{n\to\infty} P(|X_n| \geq ca_n) = 0$. Similarly, we write $X_n = o_p(a_n)$ if $\lim_{n\to\infty} P(|X_n| \geq ca_n) = 0$ holds for any positive constant c. In addition, we denote $a_n \asymp b_n$ if there exist two positive constants c and C such that $c|a_n| \leq |b_n| \leq C|a_n|$. These notation are useful for discussing the rate of convergence and consistency of the TSSS estimator below.

For any measurable functions f, f_1, f_2 defined over the closure of $\Omega \subseteq \mathbb{S}^2$, we define the theoretical and empirical inner products: $\langle f_1, f_2\rangle_\Omega = \mathrm{E}\{f_1(\mathbf{X})f_2(\mathbf{X})\}$, $\langle f_1, f_2\rangle_{n,\Omega} = n^{-1}\sum_{i=1}^n f_1(\mathbf{X}_i)f_2(\mathbf{X}_i)$, respectively, with induced norms $\|f\|_\Omega = \{\langle f, f\rangle_\Omega\}^{1/2}$ and $\|f\|_{n,\Omega} = \{\langle f, f\rangle_{n,\Omega}\}^{1/2}$. We define the supremum norm $\|f\|_{\infty,\Omega} = \sup_{\mathbf{x}\in\Omega} |f(\mathbf{x})|$ and the L^q-norm $\|f\|_{L^q(\Omega)} = \{\int_\Omega |f(\mathbf{x})|^q d\mathbf{x}\}^{1/q}$. Furthermore, denote $\langle f_1, f_2\rangle_\mathcal{E} = \sum_{\tau\in\triangle} \int_\tau (\diamondsuit f_{1,p})^\top \diamondsuit f_{2,p} \mathrm{d}\mu$, where $\diamondsuit f = (D^\alpha f_p, |\alpha| = 2)^\top$ refers to the second-order partial derivatives of f.

Lemmas 3.1–3.2 presented below provide the relationships between the inner products and norms defined above. We first make some technical assumptions.

(C1) The density function of $\mathbf{X}$, $f_\mathbf{X}(\cdot)$, is bounded away from zero and infinity over Ω.
(C2) For any spherical triangle τ within $\triangle$, it is contained in a spherical cap of radius 1/6. The minimal determining set has a constant $\lambda = 3$; see Supplementary Material Section S2.5. Assume that triangulation $\triangle$ is ϱ-quasi-uniform, that is, there exists a positive constant ϱ such that $|\triangle|/\rho_\tau \leq \varrho$, for all $\tau \in \triangle$.
(C3) The number of triangles N and sample size n satisfy $N \asymp n^\omega$ for some $C > 0$ and $\omega \in (0, 1)$.

Assumptions (C1)–(C3) are standard in the nonparametric literature (Huang 2003; Lai and Wang 2013; Yu et al. 2020). Assumption (C2) suggests the use of more uniform triangulations with smaller shape parameters (Baramidze and Lai 2011). Assumption (C3) can be handled via triangulation adjustment. Taking into account the fixed area of domain Ω, we can interchange N and $|\triangle|^{-2}$ in all results.

Lemma 3.1: *Let* $\{B_\xi\}_{\xi\in\mathcal{M}}$ *be the basis for* $\mathcal{S}_d^r(\triangle)$. *Let* $p_1 = \sum_{\xi\in\mathcal{M}} \gamma_\xi^{(1)} B_\xi$, $p_2 = \sum_{\xi\in\mathcal{M}} \gamma_\xi^{(2)} B_\xi$. *Under Assumptions* (C1) and (C3), *we have*

$$R_n = \sup_{p_1,p_2\in\mathcal{S}_d^r(\triangle)} \left| \frac{\langle p_1, p_2\rangle_{n,\Omega} - \langle p_1, p_2\rangle_\Omega}{\|p_1\|_\Omega \|p_2\|_\Omega} \right| = O_p\left\{(N\log n)^{1/2} n^{-1/2}\right\}.$$

Lemma 3.2: *Under Assumptions* (C1)–(C3), *we have*

$$V_n = \sup_{p \in S_d^r(\triangle)} \left\{ \frac{\|p\|_{\infty,\Omega}}{\|p\|_{n,\Omega}}, \|p\|_{n,\Omega} \neq 0 \right\} = O_p(\sqrt{N}),$$

$$\bar{V}_n = \sup_{p \in S_d^r(\triangle)} \left\{ \frac{\|p\|_{\mathcal{E}}}{\|p\|_{n,\Omega}}, \|p\|_{n,\Omega} \neq 0 \right\} = O_p(N).$$

3.1. Convergence rate of TSSS estimator

Before presenting the convergence results of the TSSS estimator, it is important to discuss the construction of the spherical Sobolev space and its related norms. The convergence rate of the estimator depends on the smoothness of the underlying regression function, and understanding the properties of the spherical Sobolev space is essential for understanding this relationship.

For an open set $\bar{\Omega} \subseteq \mathbb{R}^2$, let $W_q^\ell(\bar{\Omega})$ denote the classical Sobolev space as in Section 1.6 of Lai and Schumaker (2007), and $\|f\|_{W_q^\ell(\bar{\Omega})} = \sup_{|\alpha|=\ell} \|D^\alpha f\|_{L^q(\bar{\Omega})}$ denote the classical Sobolev norm. To work on an appropriate space for functions defined on $\Omega \subseteq \mathbb{S}^2$, we consider the following Sobolev space on the sphere $W_q^\ell(\Omega) := \{f : (\alpha_j^* f) \circ \phi_j^{-1} \in W_q^\ell(\bar{\Omega}_j), \forall j\}$, $\ell > 0$, where $f : \Omega \to \mathbb{R}$ is a spherical function; $\phi_j^{-1} : \mathbb{R}^2 \to \mathbb{S}^2$ is a smooth mapping; $\bar{\Omega}_j \subseteq \mathbb{R}^2$ is an open set and the support of ϕ_j^{-1}; and $\alpha_j^* : \mathbb{S}^2 \to \mathbb{R}$ is a mapping to indicate the partition of $\mathbb{S}^2$. Thus, $(\alpha_j^* f)$ and ϕ_j^{-1} bridge $\mathbb{R}^2$ and $\mathbb{R}$, and induce *spherical Sobolev norm* $\|f\|_{\ell,q,\Omega} = \sum_j \|(\alpha_j^* f) \circ \phi_j^{-1}\|_{W_q^\ell(\bar{\Omega}_j)}$ for $\ell > 0$. In addition, *spherical Sobolev seminorm* is defined as $|f|_{\ell,q,\Omega} = \sum_{|\alpha|=\ell} \|D^\alpha f_{\ell-1}\|_{L^q(\Omega)}$, where $\|D^\alpha f_{\ell-1}\|_{L^q(\Omega)}$ is the L^q-norm of the extended trivariate function $f_{\ell-1}$ restricted to Ω. When $\ell = 0$, the seminorm $|f|_{0,q,\Omega}$ reduces to the L^q-norm $\|f\|_{L^q(\Omega)}$.

We introduce two additional assumptions.

(C4) The spherical mean function $m \in W_q^{\ell+1}(\Omega)$ for some integer $0 \leq \ell \leq d, \ell = d(\text{mod}2)$.

(C4) The noise ϵ satisfies $\lim_{a\to\infty} \mathrm{E}\{\epsilon^2 I(\epsilon > a)\} = 0$, and $\mathrm{E}|\epsilon_i^{2+\eta}| \leq \nu_\eta$, for some constant $\eta \in (0, \infty)$, where $\nu_\eta \in (0, \infty)$ is a constant depending on η. The standard deviation function $\sigma(\mathbf{x})$ is continuous, and $0 < c_\sigma \leq \inf_{\mathbf{x}\in\Omega} \sigma(\mathbf{x}) \leq \sup_{\mathbf{x}\in\Omega} \sigma(\mathbf{x}) \leq C_\sigma < \infty$.

(C3') The number of triangles N and sample size n satisfy $N \asymp n^\omega$ for some $C > 0$ and $0 < \omega < \eta/(\eta + 2)$, where η is a positive and finite constant that is required to bound the higher order moments of noise ϵ as in (C5).

Assumption (C4) specifies the degree of smoothness required for m (Baramidze and Lai 2004, 2011). By characterising the smoothness of the function using the Sobolev space and its norms, we can obtain the convergence rate of the estimator for different smoothness levels of the regression function. It is worth noting that one of the challenges of using homogeneous spherical splines is that spline spaces of even and odd degrees share only the zero function due to the homogeneity of the basis. To address this limitation, our proposed method employs a simultaneous use of two spaces, $\mathcal{S}_d^r(\triangle)$ and $\mathcal{S}_{d-1}^r(\triangle)$, with one

degree even and the other odd. This approach ensures that, regardless of the parity of the unknown ℓ, the assumption can be satisfied. Assumption (C5) places constraints on the behaviour of the noise and the standard deviation function, which helps to establish the asymptotic normality of the TSSS estimator (Huang 2003; Lai and Wang 2013).

Define $s_{\lambda,m}$ and $s_{\lambda,\epsilon}$ as penalised spline estimators based on $\{m(\mathbf{X}_i)\}_{i=1}^n$ and $\{\sigma(\mathbf{X}_i)\epsilon_i\}_{i=1}^n$, respectively. Let $\mathbb{B} = \sum_{i=1}^n \mathbf{B}(\mathbf{X}_i)\mathbf{B}(\mathbf{X}_i)^\top$, then

$$s_{\lambda,m}(\mathbf{x}) = \mathbf{B}(\mathbf{x})^\top \boldsymbol{\gamma}_{\lambda,m}, \boldsymbol{\gamma}_{\lambda,m} = (\mathbb{B} + \lambda \mathbf{P})^{-1} \sum_{i=1}^n \mathbf{B}(\mathbf{X}_i) m(\mathbf{X}_i), \tag{6}$$

$$s_{\lambda,\epsilon}(\mathbf{x}) = \mathbf{B}(\mathbf{x})^\top \boldsymbol{\gamma}_{\lambda,\epsilon}, \boldsymbol{\gamma}_{\lambda,\epsilon} = (\mathbb{B} + \lambda \mathbf{P})^{-1} \sum_{i=1}^n \mathbf{B}(\mathbf{X}_i) \sigma(\mathbf{X}_i) \epsilon_i. \tag{7}$$

To evaluate the bias and variance of the proposed estimator, we decompose the estimation error as follows: $\widehat{m}_\lambda(\mathbf{x}) - m(\mathbf{x}) = \{s_{\lambda,m}(\mathbf{x}) - m(\mathbf{x})\} + s_{\lambda,\epsilon}(\mathbf{x})$, where the first term represents the size of bias and the second term represents the size of variance in the estimation. The following propositions provide bounds for the order of the bias and variance terms.

Proposition 3.1: *Under Assumptions (C1)–(C4), if $d \geq 3r + 2$ and $\triangle$ is a spherical triangulation such that $|\triangle| < 1/6$, then we have the following uniform convergence rate:* $\|s_{\lambda,m} - m\|_{\infty,\Omega} = O_p\{\lambda n^{-1} N^{3/2} |m|_{2,\infty,\Omega} + (1 + \lambda n^{-1} N^{5/2}) N^{-(\ell+1)/2} |m|_{\ell+1,\infty,\Omega}\}$.

Proposition 3.2: *Under Assumptions* (C1), (C2), (C3') *and* (C5), $\|s_{\lambda,\epsilon}\|_{L^2(\Omega)} = O_p(n^{-1/2} N^{1/2})$ *and* $\|s_{\lambda,\epsilon}\|_{\infty,\Omega} = O_p\{n^{-1/2}(N \log n)^{1/2} + \lambda n^{-3/2} N^3\}$.

Combining Propositions 3.1 and 3.2, we can obtain the following convergence rates of the TSSS estimator in terms of both the L^2 and supremum norms in Theorem 3.1. These convergence rates are important for assessing the performance of the TSSS estimator under different smoothness assumptions and for guiding the choice of the triangulation and spline basis functions.

Theorem 3.1: *Under Assumptions* (C1), (C2), (C3'), (C4) *and* (C5), *we have*

$$\|\widehat{m}_\lambda - m\|_{L^2(\Omega)} = O_p\left\{\frac{\lambda N^{3/2}}{n} |m|_{2,\infty,\Omega} + \left(1 + \frac{\lambda N^{5/2}}{n}\right) N^{-(\ell+1)/2} |m|_{\ell+1,\infty,\Omega} + \sqrt{\frac{N}{n}}\right\},$$

$$\|\widehat{m}_\lambda - m\|_{\infty,\Omega} = O_p\left\{\frac{\lambda N^{3/2}}{n} |m|_{2,\infty,\Omega} + \left(1 + \frac{\lambda N^{5/2}}{n}\right) N^{-(\ell+1)/2} |m|_{\ell+1,\infty,\Omega} + \sqrt{\frac{N \log n}{n}} + \frac{\lambda N^3}{n^{3/2}}\right\}.$$

According to Theorem 3.1, the convergence rate of the TSSS estimators depends on the sample size, the triangulation size and the roughness penalty parameter, as well as the characteristics of the estimated underlying signal. The first and third terms in the order of $\|\widehat{m}_\lambda - m\|_{L^2(\Omega)}$ and $\|\widehat{m}_\lambda - m\|_{\infty,\Omega}$ show the bias brought by the roughness penalty. When the tuning parameter is small enough, the second term in the order of $\|\widehat{m}_\lambda - m\|_{L^2(\Omega)}$

and $\|\widehat{m}_\lambda - m\|_{\infty,\Omega}$ represent the bias when approximating an arbitrary function m by a spherical spline. The last two terms are the estimation variance generated from random noise.

Theorem 3.1 also provides a guideline on how to choose triangulation. As the sample size increases, a finer triangulation can be considered for a more accurate estimation of the mean function. In addition, a finer triangulation is needed when there are rapid changes in the mean function or when the domain is highly curved or complex. A more detailed discussion of triangulation selection is given in Section 4.

It can be demonstrated that the TSSS estimator achieves the optimal convergence rate under certain regularity conditions. As indicated in Stone (1982), the optimal global approximation rate for a nonparametric estimator can be expressed as follows: for the L^q-norm ($0 < q < \infty$) of the approximation error, the rate is n^{-r^*}, where $r^* = p^*/(2p^* + d^*)$, for a p^*-times differentiable function of a d^*-dimensional measurement variable; for $q = \infty$, the optimal convergence rate becomes $(n^{-1}\log n)^{r^*}$. In the case of our problem, where $\mathbf{x} \in \Omega \subseteq \mathbb{S}^2$, we have $d^* = 2$. Furthermore, the mean function $m \in W_q^{\ell+1}(\triangle)$ implies that it is $p^* = (\ell + 1)$-times differentiable. Therefore, the optimal global approximation rate is $n^{-(\ell+1)/(2\ell+4)}$ when $0 < q < \infty$, and $(n/\log n)^{-(\ell+1)/(2\ell+4)}$ when $q = \infty$. When $\lambda = 0$, the TSSS achieves the optimal convergence rate in the L^2-norm when the number of triangles satisfies condition $N \asymp n^{1/(\ell+2)}$, and achieves the optimal convergence rate in the supremum norm when $N \asymp (n/\log n)^{1/(\ell+2)}$. When $\lambda > 0$, the optimal convergence rate in the L^2-norm is achieved under the conditions $\lambda = O\{n^{\ell/(2\ell+4)}\}$ and $N \asymp n^{1/(\ell+2)}$. Furthermore, when $\lambda = o\{n^{\ell/(2\ell+4)}\}$ and $N \asymp (n/\log n)^{1/(\ell+2)}$, the TSSS estimator achieves the optimal convergence rate in the supremum norm.

3.2. Asymptotic normality

To derive the asymptotic distribution of the TSSS estimator, we further assume the following:

(C3") The number of triangles N and sample size n satisfy $N \asymp n^\omega$ for some $C > 0$ and $1/(\ell + 2) < \omega < \eta/(\eta + 2)$, where η is a positive and finite constant that is required to bound the higher order moments of noise ϵ, see details in (C5).

(C6) The roughness parameter satisfies $\lambda = o(n^{1/2}N^{-1} \wedge nN^{-2})$.

Assumption (C3") extends Assumption (C3') in Section 3.1 by imposing a lower bound on the number of spherical triangles, which depends on the order of differentiation of the underlying mean function. This extension aligns with similar assumptions discussed in the literature for univariate and bivariate cases (Li and Ruppert 2008; Lai and Wang 2013). Meanwhile, Assumption (C6) requires a smaller smoothing parameter λ, effectively reducing bias by undersmoothing.

Theorem 3.2 below states the asymptotic normality of the proposed TSSS estimator.

Theorem 3.2: *Under Assumptions* (C1), (C2), (C3"), (C4), (C5) *and* (C6), *as* $n \to \infty$, *for each* $\mathbf{x} \in \Omega$, $[\mathbf{Var}\{\widehat{m}_\lambda(\mathbf{x})|\mathbb{X}\}]^{-1/2}\{\widehat{m}_\lambda(\mathbf{x}) - m(\mathbf{x})\} \xrightarrow{d} N(0,1)$, *where* $\mathbb{X}$ *is the collection of observed* $\mathbf{X}_1, \ldots, \mathbf{X}_n$.

Theoretically, the above asymptotic distribution result can be used to construct asymptotic confidence intervals. However, it is challenging to obtain the exact form of the standard error for general TSSS estimators due to the characteristics of the trivariate spline basis functions. To address this issue, we propose using a wild bootstrap method (Mammen 1993; Hall and Horowitz 2013) to estimate standard errors and provide PCIs, as outlined in Algorithm 1. In this algorithm, δ_i has a mean zero and enforces ϵ_i^* to have a mean zero. In addition, δ_i introduces a controlled degree of randomness to the bootstrap residuals, thereby enhancing the convergence rate as demonstrated by Mammen (1993) in their Table 1.

Furthermore, based on the bootstrap method, we construct PCIs for the mean function $m(\mathbf{x})$. In particular, an asymptotic $100(1-\alpha)\%$ PCI is given by

$$\widehat{m}_\lambda(\mathbf{x}) \pm z_{1-\alpha/2}\widehat{s}_B(\mathbf{x}), \tag{8}$$

where $z_{1-\alpha/2}$ is the $(1-\alpha/2)$th quantile of the standard normal distribution, $\widehat{s}_B(\mathbf{x})$ is the estimated sample standard error of the mean estimator computed from the B bootstrap estimators of the mean function $m(\mathbf{x})$; see Step 5a of Algorithm 1. Alternatively, we can also compute the PCIs for $m(\mathbf{x})$ based on bootstrap percentiles:

$$(L_{\alpha/2}(\mathbf{x}), U_{\alpha/2}(\mathbf{x})), \tag{9}$$

where $L_{\ell,\alpha/2}(\mathbf{x})$ and $U_{\ell,\alpha/2}(\mathbf{x})$ are the $\{100 \times (\alpha/2)\}$th and $\{100 \times (1-\alpha/2)\}$th percentiles of the bootstrap estimators of $m(\mathbf{x})$; see Step 5b of Algorithm 1.

In Algorithm 1, since the bias of $\widehat{m}_\lambda(\mathbf{x})$ is not adjusted, the proposed PCI formulas (8) and (9) generally do not produce adequate confidence intervals for for $m(\mathbf{x})$ unless the bias of $\widehat{m}_\lambda(\mathbf{x})$ is negligible relative to its variance. In practice, one can make the bias negligible by selecting a finer triangulation or larger polynomial degree d during the spline approximation, which helps reduce the bias to a negligible level. The average coverage percentage of the bootstrap PCIs is evaluated and reported in the simulation studies detailed in Section 5.

4. Implementation details

The selection of triangulation, spline basis, and penalty parameters is a critical step in the proposed TSSS method. In this section, we discuss the strategies for selecting these parameters systematically and effectively.

4.1. Selection of triangulation

Triangulation, which constructs a mesh that accurately captures the shape of the domain, is a critical step of the proposed TSSS method. Several factors influence the goodness level of triangulation required to accurately estimate the regression function. First, the complexity of the domain plays a significant role in determining the fineness of triangulation. Highly curved or complex domains typically require finer triangulation to capture the underlying pattern of the function accurately. However, constructing a suitable triangulation for a complex domain can be challenging and generating a fine triangulation can be computationally complicated and time-consuming. Usually, triangulation algorithms automatically generate the interior vertices of the triangulation. Our triangulation algorithm is able to take any given vertices (both interior and boundary vertices) to form a good triangulation.

Algorithm 1: Bootstrap estimation of TSSS estimator's standard error and PCIs for $m(\mathbf{x})$.

Input: $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, significance level α. **Output:** The standard deviation $\widehat{s}_B(\mathbf{x})$ of TSSS estimator $\widehat{m}_\lambda(\mathbf{x})$; and $100(1-\alpha)\%$ PCI for $m(\mathbf{x})$.

Step 1: Generate TSSS estimator $\widehat{m}$, and calculate residuals $\{\widehat{\epsilon}_i = Y_i - \widehat{m}_\lambda(\mathbf{X}_i)\}_{i=1}^n$.

Step 2: Generate bootstrap samples of residuals $\{\epsilon_i^* = \delta_i \widehat{\epsilon}_i\}_{i=1}^n$, where

$$\delta_i = \begin{cases} \frac{1-\sqrt{5}}{2} & amp; w.p. \frac{5+\sqrt{5}}{10} \\ \frac{1+\sqrt{5}}{2} & amp; w.p. \frac{5-\sqrt{5}}{10}, \end{cases}$$

and define $Y_i^* = \widehat{Y}_i + \epsilon_i^*$.

Step 3: Estimate TSSS estimator $\widehat{m}^*$ from $\{(\mathbf{X}_i, Y_i^*\}_{i=1}^n$.

Step 4: Repeat Step 2 and Step 3 for B times, and denote the TSSS estimators using bootstrap samples as $\{\widehat{m}_b^*\}_{b=1}^B$. Calculate the standard deviation of $\widehat{m}$ at $\mathbf{x} \in \mathbf{\Omega}$ as

$$\widehat{s}_B(\mathbf{x}) = \left[\sum_{b=1}^{B} \frac{1}{B} \{\widehat{m}_b^*(\mathbf{x}) - \overline{m}^*(\mathbf{x})\}^2 \right]^{1/2},$$

where $\overline{m}^*(\mathbf{x}) = B^{-1} \sum_{b=1}^B \widehat{m}^*(\mathbf{x})$.

Step 5a: Generate $100(1-\alpha)\%$ PCI for $m(\mathbf{x})$ by $\widehat{m}_\lambda(\mathbf{x}) \pm z_{1-\alpha/2} \widehat{s}_B(\mathbf{x})$.

Step 5b: Generate $100(1-\alpha)\%$ PCI for $m(\mathbf{x})$ by $(L_{\alpha/2}(\mathbf{x}), U_{\alpha/2}(\mathbf{x}))$, where $L_{\alpha/2}(\mathbf{x})$ and $U_{\alpha/2}(\mathbf{x})$ are the $100(\alpha/2)$th and the $100(1-\alpha/2)$th percentiles among $\{\widehat{m}_b^*\}_{b=1}^B$, respectively.

Second, the characteristic of the underlying function is an essential consideration when selecting the goodness level of triangulation. A function with rapid changes or highly localised features may require a finer triangulation to capture these features accurately. However, to establish a function-dependent triangulation, it is often necessary to have knowledge of the true signal at a substantial number of locations across the domain of interest, which may not be readily available in practical applications.

Finally, the sample size n can also influence the appropriate goodness level of triangulation. As n increases, more data points become available and practitioners can consider finer triangulation and/or increase the degree of spline space to obtain a more accurate estimation of the mean function. A practical rule of thumb we suggest is to ensure that, on average, each triangle contains at least $\{(d+2)(d+1)/2\}$ data points to support reliable estimation.

Ultimately, the selection of triangulation should balance accuracy with computational efficiency. Although it is essential to construct a suitable triangulation for surface-based data analysis, practitioners can have some flexibility without sacrificing estimation performance. Assumption (C3) in Section 3 indicates that the choice of triangulation has a minimal effect on the performance of the TSSS estimator as long as the triangulation is fine enough to capture the underlying pattern. The TSSS estimator can accurately estimate the regression function even with relatively coarse triangulations, provided that the regression

function does not oscillate much, e.g. a linear polynomial. More complicated regression functions require more complicated triangulation to approximate them well. Therefore, practitioners need to choose an appropriate level of triangulation that accurately captures the underlying pattern of the function without being too computationally expensive if the regression to be estimated is not very complicated.

In practice, practitioners can make use R packages 'Triangulation' (Wang and Lai 2019) and 'INLA' (Lindgren and Rue 2015), and MATLAB algorithm 'DistMesh' (Persson and Strang 2004) for 2D planar patches treating the boundary of a spherical domain as the boundary of a planar domain $\subseteq [0, \pi] \times [0, 2\pi)$, while ensuring boundary points that wrap around the sphere are identical, see triangulation examples constructed using 'Triangulation' in Simulation Study 2 in Section 5.2. Practitioners can also utilise the Python package 'Differentiable Surface Triangulation' developed in Rakotosaona et al. (2021) for surface-based domains. Another simple approach is to create a uniform triangulation throughout the sphere and select only the spherical triangles that contain observations while ensuring that the collection of triangles forms a triangulation that satisfies the definition of triangulation in Lai and Schumaker (2007). Examples of such triangulations are illustrated in Section 6.

4.2. Selection of spline basis functions and the roughness penalty parameter

Compared to the selection of triangulation, choosing the spline basis is generally easier. The parameters for the spline space $\mathcal{S}_d^r(\triangle)$, namely d and r, can be predetermined or selected by the user. As demonstrated in Simulation Study 1 in Section 5.1, a higher degree of spline basis d leads to a more flexible estimator, but may also result in overfitting. On the other hand, a lower degree of spline basis may lead to underfitting. In practice, the choice of d and r is closely tied to the intended interpretation of the estimated function. If the goal is to enhance the signal-to-noise ratio for visualisation or to suggest a simple parametric model, then a slightly oversmoothed function with a subjectively chosen parameter may be appropriate. However, if the focus is on accurately estimating the regression function and preserving local structures, then a slightly under-smoothed function may be more suitable.

In practice, we select the degree of spline basis d and the smoothing penalty parameter λ based on cross-validation (CV), such as K-fold CV (Lai and Wang 2013; Mu et al. 2018), which is a widely used technique for model selection. In K-fold CV, the data is randomly partitioned into K folds, with each fold used once for validation while the remaining folds train the model. The CV score is then computed as $\mathbf{CV}(d, \lambda) = \sum_{i=1}^{n} \{Y_i - \widehat{m}_\lambda^{-k[i]}(\mathbf{X}_i)\}^2$, where $k[i]$ is the index of the fold that contains the ith observation and $\widehat{m}_\lambda^{-k[i]}$ is the estimate of m given λ and the data without the $k[i]$th fold. The value of d and λ that minimises $\mathrm{CV}(d, \lambda)$ is then selected. We adopt the 5-fold CV approach throughout our numerical studies.

5. Simulation studies

This section presents an empirical evaluation of the proposed TSSS method and compares its performance with three alternative methods: ridge kernel regression (Kernel) (Cao et al. 2013), thin plate splines on the ordinary sphere (TPSOS) (Wahba 1981, 1982), and

tensor product splines (Lyche and Schumaker 2000; Wood 2006) using spherical coordinates as input (Tensor-Sphere) through three simulation studies. Simulation Study 1 investigates the performance of the TSSS for functions defined on the whole sphere and explores the effects of different triangulations and smoothing parameters. Simulation Study 2 examines the performance of the TSSS in addressing the 'leakage' problem and handling complex domains. Simulation Study 3 focuses on the performance of the proposed bootstrap variance estimator for the TSSS estimator.

We assess the performance of our simulation under various signal-to-noise ratios (SNR). When considering a fixed standard deviation, we define SNR as the ratio of the sample variance of $\{m(\mathbf{x}_j)\}_{j=1}^{N_g}$ to σ^2 (Ruppert 2002), where $\{\mathbf{x}_j\}_{j=1}^{N_g}$ are grid points defined on domain $\boldsymbol{\Omega}$. For the varying standard deviation function $\sigma(\mathbf{x})$, we define the SNR as follows:

$$\frac{\int_{\boldsymbol{\Omega}} \widehat{\mathrm{Var}}(m)d\mathbf{x}}{\int_{\boldsymbol{\Omega}} \sigma(\mathbf{x})^2 d\mathbf{x}} = \frac{A_{\boldsymbol{\Omega}} \widehat{\mathrm{Var}}(m)}{\int_{\boldsymbol{\Omega}} \sigma(\mathbf{x})^2 d\mathbf{x}} \approx \frac{\widehat{\mathrm{Var}}(m)}{N_g^{-1} \sum_{j=1}^{N_g} \sigma(\mathbf{x}_j)^2},$$

where $\widehat{\mathrm{Var}}(m)$ represents the sample variance of $\{m(\mathbf{x}_j)\}_{j=1}^{N_g}$, and $A_{\boldsymbol{\Omega}}$ denotes the area of $\boldsymbol{\Omega}$. In general, the SNRs in the simulation studies vary from 2 to 8; see Table S.1 in Supplementary Material for the specific SNRs in each simulation setting.

Throughout all numerical experiments, the TSSS method is implemented in MATLAB, and the smoothness parameter r is fixed to be one while the penalty parameter λ is selected by K-fold CV introduced in Section 4.2. The Kernel, TPSOS, and Tensor-Sphere are conducted using the R package 'mgcv' (Wood 2003, 2017) based on the functions `magic()`, `s()` and `te()`, respectively, along with `gam()`, and λ is selected by generalised cross-validation (GCV) for the competing methods. In addition, for TPSOS, we choose $m = 2$ to penalise the second derivative smoothness. For both TPSOS and Tensor-Sphere, we fix the dimension for the smooth term as $k = 100$ and $k = 64$, respectively. These dimensions are selected to align closely with the dimension utilised in the TSSS method to ensure a fair comparison.

5.1. Simulation study 1: functions on whole sphere

In this simulation, we generate n design points $\mathbf{X}_i = (X_{i1}, X_{i2}, X_{i3})^\top \in \mathbb{S}^2$ with $\|\mathbf{X}_i\| = 1$, for $i = 1, \ldots, n$. We generate the response variable Y_i according to the following model:

$$Y_i = m(\mathbf{X}_i) + \sigma_1(\mathbf{X}_i)\epsilon_i, \epsilon_i \overset{i.i.d}{\sim} N(0,1), \quad i = 1, \ldots, n, \tag{10}$$

where $m(\mathbf{x}) : \mathbb{S}^2 \to \mathbb{R}$ represents the mean function, and we consider two mean functions: (i) $m_1(\mathbf{x}) = -2 + 1/2\{x_1^2 + \exp(2x_2^3) + \exp(2x_3^2) + 10x_1x_2x_3\}$; (ii) $m_2(\mathbf{x}) = 2.5\{(x_1 - 1)(x_2 - 1)x_3^2\} - 3$. The illustration of m_1 and m_2 is presented in Figure 4(a,b). We generate $\mathbf{X}_i$'s on $n = 400$, 900, and 2500 random locations, i.e. irregularly spaced locations, on a unit sphere for training and evaluate the performance of the estimators at $N_g = 10{,}201$ grid points $\{\mathbf{x}_j\}_{j=1}^{N_g}$ on the sphere for each simulation setting, which is fixed across all 100 replications. We consider both the constant standard deviation $\sigma_1(\mathbf{x}) \equiv \sigma$ ($\sigma = 0.50, 0.75$), and the heterogeneous standard deviation function $\sigma_1(\mathbf{x})$, which varies in different spatial locations. Specifically, we define $\sigma_1(\mathbf{x}) = c_\sigma\{1 - (x_1^2 + x_2^2 + 1.5x_3^2)/10\}$,

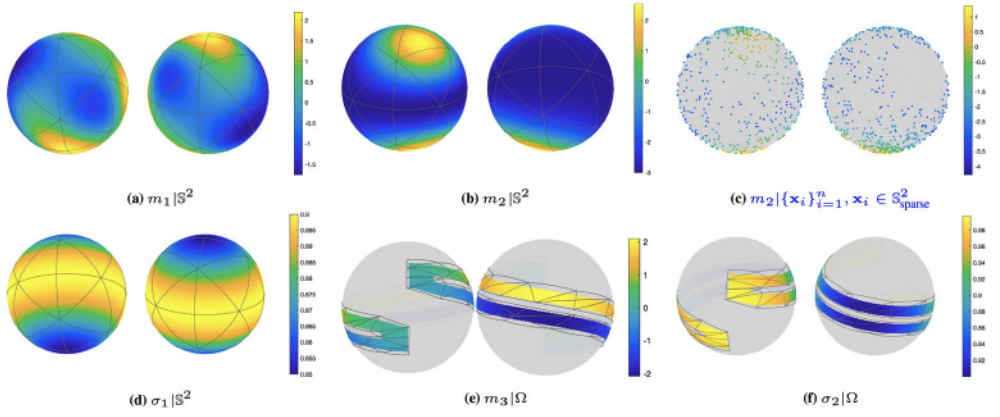

Figure 4. Plots of the true mean and standard deviation functions in simulation studies, shown from two viewpoints for each: (a) $m_1|\mathbb{S}^2$ with triangulation ($N = 32$); (b) $m_2|\mathbb{S}^2$ with triangulation ($N = 32$); (c) m_2 evaluated on sparse training locations $\{\mathbf{x}_i\}_{i=1}^n$ that are constrained to observable region $x_{i1} + x_{i2} < 1.2$; (d) $\sigma_1|\mathbb{S}^2$ with $c_\sigma = 1$; and (e) $m_3|\Omega$ with triangulation ($N = 58$); and (f) $\sigma_2|\Omega$ with $c_\sigma = 1$.

where c_σ takes values of 0.50 and 0.75. A visual representation of the standard deviation function $\sigma_1(\mathbf{x})$ can be found in Figure 4(c). In addition, we consider a case where sparse training locations $\{\mathbf{x}_i\}_{i=1}^n$ are constrained to the region $x_{i1} + x_{i2} < 1.2$ on the sphere, denoted as $\mathbb{S}^2_{\text{sparse}} = \{(x_{i1}, x_{i2}, x_{i3}) \in \mathbb{S}^2 : x_{i1} + x_{i2} < 1.2\}$. The mean function is given by $m_2(\mathbf{x})$, and we refer to this scenario as $m_2|\mathbb{S}^2_{\text{sparse}}$; see Figure 4(c) for a sample set of sparse training locations $\{\mathbf{x}_i\}_{i=1}^n$.

5.1.1. Comparative analysis of TSSS with competing methods

In this section, we conduct a comparative analysis between the proposed TSSS method and the Kernel and TPSOS methods. To evaluate the performance of different estimators, we use two metrics: the predicted mean squared error (PMSE) and the training mean squared error (TMSE). The PMSE is calculated on a set of grid points $\{\mathbf{x}_j\}_{j=1}^{N_g}$ on the sphere, and is defined as $N_g^{-1}\sum_{j=1}^{N_g}\{m(\mathbf{x}_j) - \widehat{m}(\mathbf{x}_j)\}^2$, where m and $\widehat{m}$ are the true and generic estimated functions. TMSE is calculated at the training data locations $\{\mathbf{X}_i\}_{i=1}^n$ and is defined as $n^{-1}\sum_{i=1}^{n}\{m(\mathbf{X}_i) - \widehat{m}(\mathbf{X}_i)\}^2$. In addition, we report the dimension of the design parameters ('Dim') and the average computation time per replication ('Time'), measured in seconds.

Due to similar results, we only report the results in the heterogeneous standard deviation setting in Table 1 and defer the results in the constant standard deviation setting to the Supplementary Material. Table 1 reports the estimation results for $m_1|\mathbb{S}^2$, $m_2|\mathbb{S}^2$ and $m_2|\mathbb{S}^2_{\text{sparse}}$ when the errors are heterogeneous with varying standard deviation function: $\sigma_1(\mathbf{x}) = c_\sigma\{1 - (x_1^2 + x_2^2 + 1.5x_3^2)/10\}$. For TSSS, we utilise the triangulation mesh with $N = 32$ triangles, illustrated in Figure 4, along the spherical spline basis based on either a fixed degree ($d = 3$) or a degree selected via CV (d_{CV}), as described in Section 4.2.

In Table 1, it is evident that at a lower noise level ($\sigma = 0.5$), TSSS with degree $d = 3$ outperforms others with the smallest PMSE and TMSE across all sample sizes, suggesting its precision robustness. For a higher noise level ($\sigma = 0.75$), the TSSS with CV-selected

Table 1. Simulation Study 1 results for different estimation methods on mean functions $m_1|\mathbb{S}^2$, $m_2|\mathbb{S}^2$ and $m_2|\mathbb{S}^2_{\text{sparse}}$ with varying standard deviation function $\sigma_1(\mathbf{x})$: the average (and standard deviations) of predicted mean squared error (PMSE), training mean squared error (TMSE), dimension of the design parameters (Dim), and computation time per iteration in seconds (Time).

$m_1|\mathbb{S}^2$

		$n = 400$						$n = 900$						$n = 2500$					
c_σ	Method	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)
0.5	Kernel	22.61	(4.6)	19.71	(3.9)	400	4901	10.29	(2.1)	9.95	(2.1)	900	8325	5.28	(0.8)	5.17	(0.7)	2500	23207
	TPSOS	25.59	(4.7)	23.11	(4.2)	100	2	13.33	(2.2)	12.72	(2.1)	100	4	6.26	(1.0)	6.12	(0.9)	100	8
	Tensor-Sphere	50.41	(11.4)	38.64	(6.7)	64	0	39.18	(4.5)	33.70	(4.0)	64	0	35.97	(1.9)	32.20	(2.3)	64	0
	TSSS, $d = 3$	21.78	(4.5)	19.43	(3.9)	57	3	12.97	(2.0)	12.15	(1.9)	57	3	7.85	(0.9)	7.58	(0.8)	57	3
	TSSS, d_{CV}	23.07	(6.0)	20.44	(4.8)	102	19	11.19	(2.4)	10.58	(2.3)	129	24	4.56	(0.9)	4.48	(0.8)	151	41
0.75	Kernel	47.27	(9.9)	41.51	(8.4)	400	4893	20.04	(4.7)	19.52	(4.6)	900	8252	8.89	(1.8)	8.74	(1.7)	2500	23583
	TPSOS	45.72	(8.9)	42.57	(8.5)	100	2	24.55	(4.4)	23.75	(4.2)	100	3	11.50	(1.9)	11.32	(1.9)	100	8
	Tensor-Sphere	74.20	(13.1)	60.24	(10.0)	64	0	52.10	(6.7)	45.51	(5.7)	64	0	41.24	(2.7)	37.08	(2.9)	64	0
	TSSS, $d = 3$	39.82	(9.3)	36.23	(8.1)	57	3	22.72	(4.3)	21.50	(4.1)	57	3	11.94	(2.0)	11.63	(1.9)	57	3
	TSSS, d_{CV}	39.66	(10.4)	36.08	(8.8)	110	19	21.19	(4.8)	20.32	(4.7)	115	23	9.92	(1.9)	9.74	(1.8)	132	41

$m_2|\mathbb{S}^2$

		$n = 400$						$n = 900$						$n = 2500$					
c_σ	Method	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)
0.5	Kernel	21.26	(5.1)	18.72	(4.3)	400	4356	8.60	(2.3)	8.33	(2.3)	900	9771	3.11	(0.8)	3.07	(0.8)	2500	26619
	TPSOS	22.18	(4.7)	20.82	(4.3)	100	9	11.57	(2.4)	11.12	(2.2)	100	19	5.42	(0.9)	5.29	(0.9)	100	39
	Tensor-Sphere	25.68	(6.9)	19.94	(4.5)	64	0	12.81	(3.4)	10.40	(2.3)	64	0	5.48	(1.0)	4.72	(0.8)	64	0
	TSSS, $d = 3$	16.73	(4.2)	15.37	(3.7)	57	3	8.61	(2.0)	8.27	(1.9)	57	3	3.59	(0.9)	3.55	(0.9)	57	4
	TSSS, d_{CV}	18.19	(5.6)	16.69	(4.9)	86	21	8.91	(2.1)	8.57	(2.0)	71	29	3.42	(0.9)	3.39	(0.8)	60	49
0.75	Kernel	46.94	(10.9)	41.54	(9.3)	400	4355	19.30	(5.3)	18.70	(5.3)	900	9763	6.99	(1.9)	6.90	(1.8)	2500	26642
	TPSOS	40.41	(9.4)	38.67	(8.8)	100	8	21.39	(4.7)	20.74	(4.5)	100	18	9.94	(1.8)	9.76	(1.8)	100	40
	Tensor-Sphere	47.97	(12.6)	38.93	(9.1)	64	0	24.51	(6.4)	20.41	(4.7)	64	0	10.42	(2.1)	9.04	(1.8)	64	0
	TSSS, $d = 3$	36.40	(9.3)	33.58	(8.2)	57	3	19.04	(4.5)	18.30	(4.3)	57	3	7.96	(2.0)	7.88	(2.0)	57	4
	TSSS, d_{CV}	34.77	(10.1)	32.51	(8.9)	80	21	16.47	(4.7)	15.92	(4.5)	64	28	6.76	(1.8)	6.71	(1.8)	61	48

$m_2|\mathbb{S}^2_{\text{sparse}}$

		$n = 400$						$n = 900$						$n = 2500$					
σ	Method	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)
0.5	Kernel	26.04	(4.6)	17.71	(3.3)	400	2390	10.99	(3.6)	7.92	(1.8)	900	5468	4.04	(1.9)	2.99	(0.8)	2500	15984
	TPSOS	21.05	(5.1)	20.17	(4.7)	100	1	10.67	(2.2)	10.47	(1.9)	100	4	4.74	(0.9)	4.71	(0.9)	100	6
	Tensor-Ssphere	26.79	(7.5)	19.80	(4.5)	64	0	12.37	(2.9)	10.04	(2.0)	64	0	5.49	(1.2)	4.67	(0.9)	64	0
	TSSS, $d = 3$	17.73	(4.7)	14.88	(3.6)	57	2	9.55	(2.9)	7.84	(1.8)	57	2	4.33	(1.3)	3.50	(0.8)	57	2
	TSSS, d_{CV}	18.49	(5.0)	15.80	(4.3)	80	13	9.59	(2.6)	8.18	(1.7)	75	13	3.81	(1.0)	3.37	(0.8)	67	18
0.75	Kernel	50.40	(9.8)	38.54	(7.3)	400	2390	23.51	(6.1)	17.63	(4.1)	900	5542	8.93	(3.4)	6.70	(1.7)	2500	16198
	TPSOS	38.49	(10.1)	37.48	(9.5)	100	2	19.84	(4.4)	19.61	(3.9)	100	4	8.98	(1.8)	8.98	(1.8)	100	6
	Tensor-Sphere	48.83	(13.1)	38.58	(9.4)	64	0	23.39	(5.6)	19.65	(4.1)	64	0	10.36	(2.3)	8.97	(1.8)	64	0
	TSSS, $d = 3$	38.44	(10.2)	32.41	(7.9)	57	2	21.17	(6.6)	17.36	(4.0)	57	2	9.61	(2.8)	7.75	(1.8)	57	2
	TSSS, d_{CV}	36.87	(10.1)	32.44	(8.8)	87	13	16.70	(5.2)	14.92	(4.0)	66	12	7.21	(1.9)	6.49	(1.6)	61	17

The results for TSSS are based on a triangulation with $N = 32$ triangles and spline basis functions with either a fixed degree $d = 3$ or a CV-selected degree d_{CV}. TPSOS and Tensor-Sphere have dimensions $k = 100$ and $k = 64$, respectively. Kernel method results have only 30 iterations instead of 100 due to computation inefficiency. A factor of 10^3 scales the reported average (and standard deviations) of PMSEs and TMSEs. Results for $m_1|\mathbb{S}^2$ and $m_2|\mathbb{S}^2$ with constant standard deviations can be found in Table S.2 in Supplementary Material.

degree d_{CV} consistently exhibits strong performance, often achieving or competing for the lowest errors, reflecting its effective handling of increased variability. Conversely, the Tensor-Sphere shows relatively poor performance, with the highest PMSE and TMSE values under both noise conditions, marking it as the least accurate. The kernel and TPSOS generally achieve moderate error rates, neither excelling nor performing the poorest across the different scenarios. These patterns remain consistent as the sample size increases from 400 to 2500, indicating that the relative performance of these methods is stable across different amounts of data.

Regarding computational efficiency, as indicated in Table 1, the Kernel method is the most time consuming among the methods considered. Comparatively, the Tensor-Sphere stands out due to its minimal computational time. The TSSS with a fixed degree d and the TPSOS exhibit similar computational performance. The TSSS with a CV-selected degree d exhibits a slightly slower computational pace, as anticipated, but remains significantly faster, by hundreds or thousands of times, than the Kernel method. Therefore, TSSS achieves a favourable balance between estimation accuracy and computational efficiency for functions defined in the entire sphere.

Lastly, we observe that the performance of TSSS under the sparse scenario, $m_2|\mathbb{S}^2_{\text{sparse}}$, are very similar to those of the non-sparse scenario, demonstrating the stability of the TSSS estimators against sparsity of observed locations. The advantages of estimation accuracy and computational efficiency of TSSS still hold with sparse training locations, as long as the trend of mean functions are relatively smooth in the unobserved sub-domains.

5.1.2. Impact of d and Δ, and CV for d and λ on TSSS estimation

This section investigates the effect of degree d and triangulation $\triangle$ on the estimation performance of the TSSS. We also investigate the behaviour of CV-selected d, λ, and their impact on TSSS. Specifically, we first investigate the impact of d and $\triangle$ on TSSS under different σ and n settings. In addition, we fix $\triangle$ and select the optimal combination of d and the penalty parameter λ through the CV method and study the behaviour of CV-selected d_{CV} compared to fixed d's.

The results in Figure 5 reveal that different combinations of d and triangulation can lead to varied TSSS performance. Firstly, when d is small, a finer triangulation improves PMSE; see $d = 2$. In contrast, for larger d, a finer triangulation may not necessarily improve the PMSE, as observed in the cases of $d = 3, 4, 5$. This is because a larger degree d and the number of triangles N can lead to over-parameterisation of the mean function. Secondly, when d is too small for a fixed triangulation, the mean functions may not be fully represented, leading to underfitting. Thirdly, as the sample size n increases, both d and the fineness of the triangulation may need to be adjusted to achieve optimal estimation performance. When noise is small, the estimated mean function can better approximate the true mean function, allowing a higher degree of spline basis d and a finer triangulation to be used without overfitting. However, the choice of d and N should still be balanced with computational efficiency, as a very high d and a very large N can lead to high computational costs. In general, a careful trade-off between accuracy and computational efficiency must be considered when choosing the appropriate d and N for a given sample size and noise level. The reader is referred to Section 4 for a detailed discussion.

Next, we assess the performance of the TSSS estimator employing spline basis functions with the selected degree d_{CV}. As before, we consider simulation settings with various

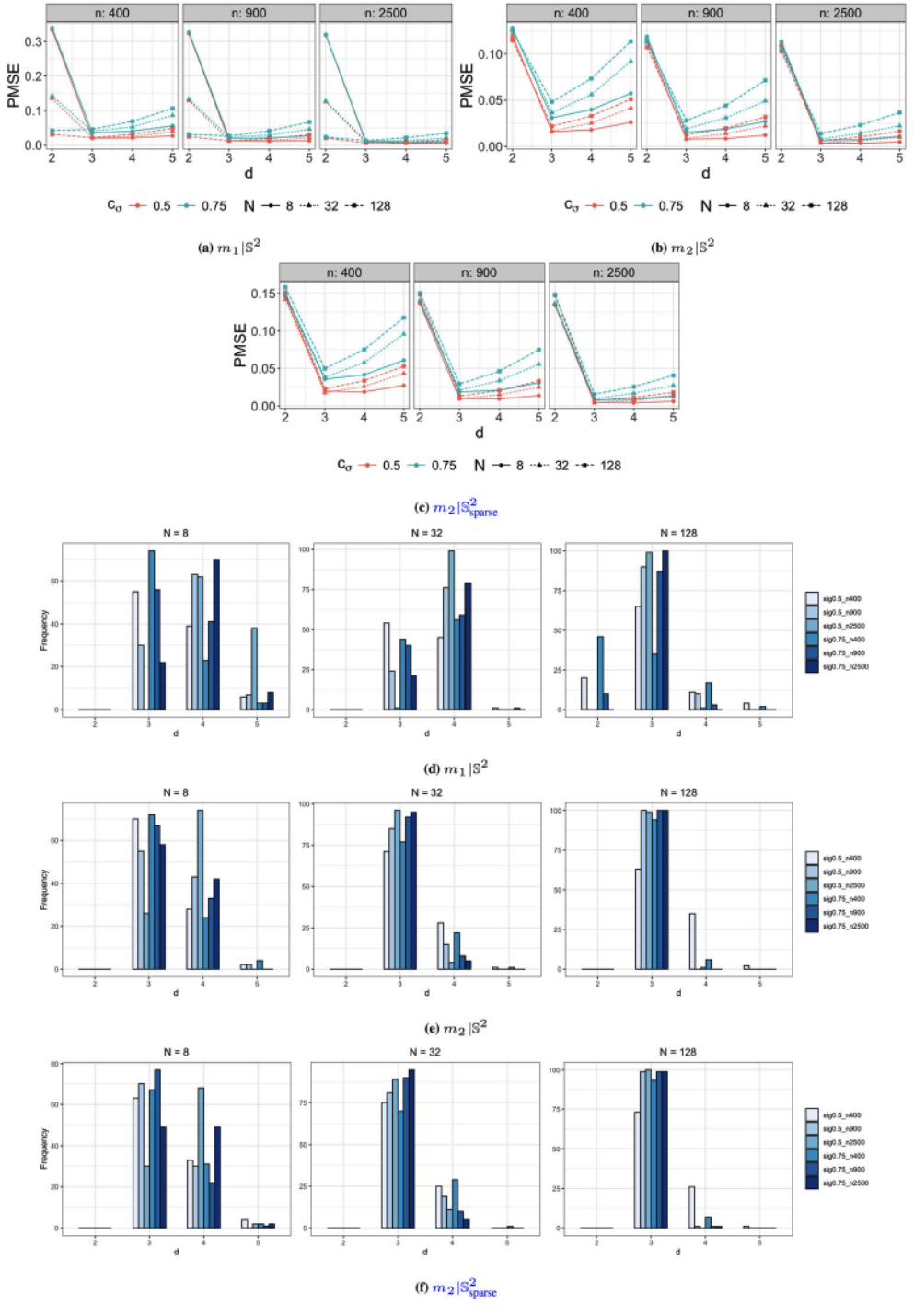

Figure 5. (a)–(c): Performance of TSSS for $m_1|\mathbb{S}^2$, $m_2|\mathbb{S}^2$ and $m_2|\mathbb{S}^2_{\text{sparse}}$ in Simulation Study 1 with different n, d, c_σ, and N; (d)–(f): Bar graphs of d_{CV} under different combinations of $c_\sigma = 0.50, 0.75$ (sig), $n = 400, 900, 2500$, and $N = 8, 32, 128$ in Simulation Study 1. The results for $m_1|\mathbb{S}^2$ and $m_2|\mathbb{S}^2$ with constant standard deviations can be found in Figures S.1 and S.2 of Supplementary Material. (a) $m_1|\mathbb{S}^2$. (b) $m_2|\mathbb{S}^2$. (c) $m_2|\mathbb{S}^2_{\text{sparse}}$. (d) $m_1|\mathbb{S}^2$. (e) $m_2|\mathbb{S}^2$ and (f) $m_2|\mathbb{S}^2_{\text{sparse}}$.

combinations of parameters: $n = 400, 900, 2500$; $d = 2, 3, 4, 5$; $c_\sigma = 0.50, 0.75$; and $N = 8$, 32, 128, applied to $m_1|\mathbb{S}^2$, $m_2|\mathbb{S}^2$ and $m_2|\mathbb{S}^2_{\text{sparse}}$. For each setting, we simulate $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$ and compute the TSSS estimator using d_{CV}. This procedure is repeated 100 times for each setting. The bar graphs in Figure 5 and columns '$f_{d_{\text{CV}}}$' in Table 2 illustrate the empirical frequency of d_{CV} in various settings. Table 2 also presents the PMSE of the TSSS estimators for $d = 2, 3, 4, 5$ and d_{CV}. From the table and bar graphs, for fixed n, σ, as N increases, smaller d_{CV}'s are selected. This is consistent with our previous discussion on the balance between d and N. In addition, when n and N are fixed, as σ increases, a smaller d_{CV} tends to be selected. This trend is reasonable because a higher d tends to pick up more noise, which does not benefit the overall fitting. Furthermore, compared to estimators with fixed $d = 2, 3, 4, 5$, the estimator with d_{CV} is among the best in terms of PMSE. Therefore, the proposed procedure for selecting d and λ by CV is validated. The results of the sparse scenario, $m_2|\mathbb{S}^2_{\text{sparse}}$, closely resembles non-sparse scenarios, demonstrating the stability of the TSSS estimators against sparsity of training locations.

5.2. Simulation study 2: functions on irregularly shaped spherical patches

In this simulation study, we generate the response variable Y_i from the following model:

$$Y_i = m_3(\mathbf{X}_i) + \sigma_2(\mathbf{X}_i)\epsilon_i, \epsilon_i \overset{i.i.d}{\sim} N(0,1), \quad i = 1, \ldots, n, \tag{11}$$

with mean function $m_3 : \boldsymbol{\Omega} \to \mathbb{R}$ and varying standard deviation function $\sigma_2(\mathbf{x}) = 1 - (1.5(x_1+1)^2 + x_2^2 + x_3^2)/30$ defined on irregularly shaped spherical patch $\boldsymbol{\Omega}$. Specifically, for $\mathbf{x} \in \boldsymbol{\Omega}$, we define $m_3(\mathbf{x}) = \widetilde{m}_3(\mathbf{x}')$, where $\widetilde{m}_3(\mathbf{x}) = (0.08\pi + 0.84 + x_1)I(x_1 \geq -0.84, x_3 > 0) - (0.08\pi + 0.84 + x_1)I(x_1 \geq -0.84, x_3 \leq 0) - 0.16 \arctan\{x_3/(0.84+x_1)\}$ $I(x_1 < -0.84)$; the spherical coordinates of $\mathbf{x}$, (θ, ϕ), and the spherical coordinates of $\mathbf{x}'$, (θ', ϕ'), satisfy $\theta' = \theta, \phi' = \phi + \theta/6$. This choice of m_3 allows us to test the performance of the methods for functions with complex patterns on non-standard spherical domains. See Figure 4 for the mean and standard deviation functions. We generate $\mathbf{X}_i$'s at $n = 400, 900$ and 2500 random locations in the complex domains for training. The SNRs of Simulation Study 2 are reported in Table S.1.

For TSSS, since m_3 is not wildly oscillating, we select a spline basis based on $d = 2$ and a triangulation mesh with 56 triangles. The triangulation is constructed by mapping the spherical domain to the planar domain, triangulating the planar domain using `TriMesh` from R package 'Triangulation' (Wang and Lai 2019), and mapping the planar triangulation back to spherical coordinates. This procedure would require the user to provide boundary points and interior holes' boundary points and adjust the identical boundary points that wrap around the longitude or co-latitude. This practice is suitable and recommended for simple domains due to the nice properties of `TriMesh` function. Refer to Figure 4(d) for the triangulation.

Then, we calculate the PMSE by evaluating the performance on 11,986 grid points for $m_3|\boldsymbol{\Omega}$, with results shown in Table 3. The table shows that TSSS, both with a fixed degree ($d = 2$) and a cross-validation-selected degree (d_{CV}), outperforms TPSOS, Kernel and Tensor-Sphere in terms of both PMSE and TMSE across all settings. TSSS also demonstrates significant computational benefits over the Kernel approach. It achieves a

Table 2. Performance comparison of TSSS Estimators for $m_1|\mathbb{S}^2$ and $m_2|\mathbb{S}^2$ in Simulation Study 1: d_{CV} vs. fixed degree $d = 2, 3, 4, 5$.

$m_1|\mathbb{S}^2$

		$N = 8$						$N = 32$						$N = 128$					
		$n = 400$		$n = 900$		$n = 2500$		$n = 400$		$n = 900$		$n = 2500$		$n = 400$		$n = 900$		$n = 2500$	
c_σ	d	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$
0.5	2	333.35	0	322.90	0	318.50	0	134.50	0	129.03	0	125.59	0	30.84	20	24.63	0	20.80	0
		(7.82)		(2.73)		(1.29)		(4.93)		(2.21)				(0.88)		(4.45)		(1.55)	(0.66)
	3	20.48	55	12.43	30	8.20	0	21.78	54	12.97	24	7.85	1	21.44	65	12.21	90	5.96	99
		(3.78)		(1.81)		(0.60)		(4.48)		(1.99)		(0.89)		(4.33)		(1.99)		(0.93)	
	4	20.95	39	11.24	63	5.21	62	24.05	45	13.11	76	6.01	99	31.36	11	18.62	10	9.62	1
		(5.50)		(2.18)		(0.93)		(5.18)		(2.15)		(0.94)		(5.37)		(2.31)		(1.13)	
	5	26.74	6	12.57	7	5.15	38	39.22	1	20.74	0	9.28	0	48.33	4	30.09	0	15.37	0
		(5.59)		(2.31)		(0.95)		(7.09)		(2.91)		(1.28)		(7.18)		(3.23)		(1.52)	
	d_{CV}	21.79	–	12.04	–	5.29	–	23.07	–	11.19	–	4.56	–	23.95	–	11.34	–	4.79	–
		(4.95)		(2.36)		(0.96)		(6.02)		(2.44)		(0.85)		(5.83)		(2.16)		(0.89)	
0.75	2	339.35	0	326.40	0	319.74	0	142.69	0	133.00	0	127.03	0	42.30	46	30.63	10	23.20	0
		(9.91)		(4.08)		(1.72)		(8.59)		(3.82)		(1.44)		(8.87)		(3.19)		(1.29)	
	3	34.93	74	20.03	56	11.10	22	39.82	44	22.72	40	11.94	21	45.79	35	26.61	87	13.11	100
		(7.84)		(3.72)		(1.29)		(9.25)		(4.32)		(2.04)		(9.54)		(4.45)		(2.06)	
	4	40.56	23	20.79	41	9.63	70	52.58	56	29.05	59	13.39	79	68.78	17	41.31	3	21.46	0
		(11.16)		(4.75)		(1.95)		(11.34)		(4.80)		(2.10)		(11.94)		(5.18)		(2.55)	
	5	55.64	3	26.37	3	10.76	8	86.11	0	46.00	1	20.72	0	106.21	2	66.79	0	34.26	0
		(12.20)		(4.90)		(2.08)		(15.65)		(6.46)		(2.87)		(16.06)		(7.26)		(3.41)	
	d_{CV}	36.85	–	20.78	–	10.12	–	39.66	–	21.19	–	9.92	–	41.01	–	20.26	–	8.99	–
		(9.13)		(4.19)		(1.94)		(10.42)		(4.76)		(1.90)		(10.56)		(5.06)		(1.66)	

$m_2|\mathbb{S}^2$

		$N = 8$						$N = 32$						$N = 128$					
		$n = 400$		$n = 900$		$n = 2500$		$n = 400$		$n = 900$		$n = 2500$		$n = 400$		$n = 900$		$n = 2500$	
c_σ	d	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$	PMSE	$f_{d_{CV}}$
0.5	2	120.16	0	115.00	0	112.18	0	118.35	0	112.43	0	109.05	0	114.51	0	107.31	0	102.80	0
		(3.87)		(1.91)		(0.86)		(4.72)		(2.10)		(0.92)		(5.25)		(2.26)		(1.04)	
	3	15.94	70	7.86	55	3.70	26	16.73	71	8.61	85	3.59	96	21.77	63	12.59	100	6.19	99
		(4.27)		(1.89)		(0.62)		(4.19)		(2.02)		(0.91)		(4.51)		(2.25)		(0.99)	
	4	18.09	28	8.52	43	3.37	74	25.22	28	13.85	15	6.39	4	32.94	35	19.75	0	10.27	1
		(5.51)		(2.12)		(0.83)		(5.37)		(2.35)		(1.01)		(5.70)		(2.64)		(1.23)	
	5	25.98	2	12.14	2	4.89	0	41.55	1	22.04	0	9.92	0	50.97	2	31.96	0	16.39	0
		(5.52)		(2.35)		(0.99)		(7.75)		(3.10)		(1.38)		(7.79)		(3.65)		(1.65)	
	d_{CV}	17.01	–	8.43	–	3.47	–	18.19	–	8.91	–	3.42	–	18.83	–	8.64	–	4.03	–
		(4.80)		(2.31)		(0.84)		(5.62)		(2.07)		(0.86)		(5.14)		(2.03)		(0.79)	
0.75	2	128.22	0	118.76	0	113.52	0	126.73	0	116.68	0	110.67	0	124.75	0	112.93	0	105.09	0
		(8.28)		(3.51)		(1.49)		(8.64)		(3.67)		(1.58)		(9.64)		(4.02)		(1.77)	
	3	30.97	72	15.18	67	6.48	58	36.40	77	19.04	92	7.96	95	48.14	94	28.09	100	13.86	100
		(9.17)		(3.99)		(1.34)		(9.27)		(4.46)		(2.04)		(10.13)		(5.00)		(2.22)	
	4	39.77	24	18.76	33	7.41	42	56.02	22	30.96	8	14.31	5	73.37	6	44.18	0	23.00	0
		(12.39)		(4.70)		(1.91)		(11.99)		(5.25)		(2.28)		(12.85)		(5.89)		(2.77)	
	5	57.41	4	27.01	0	10.92	0	92.09	1	49.10	0	22.17	0	113.51	0	71.49	0	36.71	0
		(12.36)		(5.26)		(2.22)		(17.14)		(6.95)		(3.10)		(17.50)		(8.16)		(3.71)	
	d_{CV}	34.18	–	16.17	–	6.69	–	34.77	–	16.47	–	6.76	–	32.07	–	17.43	–	8.60	–
		(10.06)		(4.61)		(1.62)		(10.12)		(4.71)		(1.80)		(8.26)		(4.13)		(1.70)	

The frequency of selecting d_{CV} among 100 replications is denoted as $f_{d_{CV}}$. The reported average (and standard deviations) of PMSEs are scaled by a factor of 10^3. Additional results for $m_1|\mathbb{S}^2$ and $m_2|\mathbb{S}^2$ based on the constant standard deviation function setting can be found in Table S.3 in Supplementary Material.

reasonable computation efficiency that is comparable to TPSOS, especially for larger sample sizes. Although the Tensor-Sphere method frequently records minimal computational times, at times as low as zero seconds, it falls short of TSSS in terms of estimation accuracy. The success of TSSS is attributed to its ability to address the 'leakage' problem by adopting 'domain-aware' splines.

Table 3. Simulation Study 2 results for different estimation methods on mean function $m_3|\Omega$ with varying standard deviation function $\sigma_2(\mathbf{x})$: the average (and standard deviations) of predicted mean squared error (PMSE), training mean squared error (TMSE), dimension of the design parameters (Dim), and computation time per iteration in seconds (Time).

		$m_3\|\Omega$															
		$n = 400$						$n = 900$						$n = 2500$			
c_σ	Method	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		*Dim*	Time(s)	PMSE		TMSE		Dim	Time(s)
0.5	Kernel	141.48	(31.6)	132.00	(33.7)	400	1928	108.22	(44.3)	107.12	(45.5)	900	4978	51.16	(7.0)	52.35	(7.2)	2500	21692
	TPSOS	95.68	(12.8)	64.73	(7.0)	100	10	53.36	(5.3)	44.31	(4.2)	100	20	31.64	(1.5)	31.16	(1.7)	100	43
	Tensor-Sphere	58.26	(7.0)	51.14	(5.6)	64	0	42.73	(4.0)	40.03	(4.8)	64	0	30.01	(1.5)	30.64	(1.6)	64	0
	TSSS, $d = 2$	17.69	(5.3)	16.02	(4.4)	45	25	9.72	(2.5)	9.08	(2.1)	45	25	4.94	(0.8)	4.81	(0.8)	45	25
	TSSS, d_{CV}	18.08	(6.0)	16.25	(4.3)	100	263	9.88	(2.2)	9.30	(2.1)	85	325	4.69	(0.9)	4.53	(0.9)	132	270
0.75	Kernel	167.81	(13.7)	157.96	(18.6)	400	1928	133.84	(35.4)	132.22	(35.3)	900	4948	82.62	(41.6)	84.45	(43.3)	2500	21301
	TPSOS	139.64	(17.0)	107.79	(12.8)	100	10	78.65	(8.6)	68.12	(7.7)	100	21	41.77	(2.9)	41.15	(2.8)	100	44
	Tensor-Sphere	79.56	(12.9)	71.57	(10.3)	64	0	55.80	(5.3)	54.42	(5.5)	64	0	36.30	(3.1)	37.01	(3.1)	64	0
	TSSS, $d = 2$	32.27	(10.8)	29.72	(9.2)	45	25	17.17	(4.7)	16.30	(4.4)	45	25	8.24	(1.8)	8.03	(1.8)	45	25
	TSSS, d_{CV}	30.42	(9.3)	28.39	(8.0)	54	341	17.62	(5.4)	16.75	(4.9)	63	333	8.66	(1.9)	8.46	(1.8)	68	359

TSSS settings: $d = 2$ or a CV-selected d_{CV}, $N = 58$, $r = 1$. TPSOS and Tensor-Sphere have dimensions $k = 100$ and $k = 64$, respectively. Kernel method results have only 30 iterations instead of 100 due to computation inefficiency. A factor of 10^3 scales the reported average (and standard deviations) of PMSEs and TMSEs. Additional results for mean functions $m_3|\Omega$ with constant standard deviation function can be found in Table S.4 in Supplementary Material.

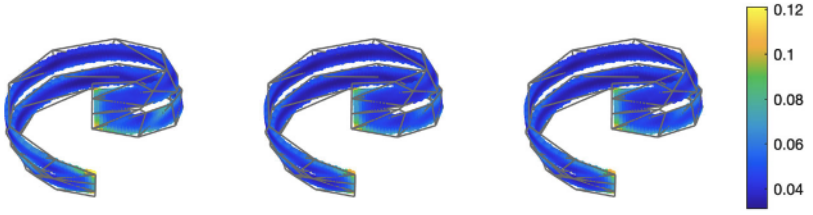

Figure 6. Standard error function plots for $m_3|\Omega$ in Simulation Study 3: from left to right, SE^{MC}, $\text{SE}^{\text{Boot}}_{\text{mean}}$, and $\text{SE}^{\text{Boot}}_{\text{median}}$.

5.3. Simulation study 3: stability of TSSS estimator

In this simulation study, we verify the stability of the TSSS estimator using a bootstrap procedure and study the reliability of PCIs of mean functions. We perform a total of R Monte Carlo (MC) replications. In each replication, we simulate observations $\{(\mathbf{X}_i, Y_i)\}_{i=1}^n$, using the same data generation settings as those used in Simulation Study 2 in Section 5.2.

Firstly, we study the bootstrap standard errors of TSSS estimators. We fix $\mathbf{X}_i$'s on a grid with $n = 3065$, $d = 3$, and $c_\sigma = 0.5$. The bootstrap standard errors (SE) of TSSS estimator are generated through Algorithm 1 using $B = 100$ wild bootstrap samples $\{\text{SE}^{\text{Boot}}_{rep}\}_{rep=1}^R$, where $R = 200$ is the number of MC replications. We report the mean and median of $\{\text{SE}^{\text{Boot}}_{rep}\}_{rep=1}^R$, denoted as $\text{SE}^{\text{Boot}}_{\text{mean}}$ and $\text{SE}^{\text{Boot}}_{\text{median}}$, respectively. The Monte Carlo standard error SE^{MC} from the R TSSS estimators is calculated from the MC replications and serves as the ground truth. We then compare SE^{MC} to $\text{SE}^{\text{Boot}}_{\text{mean}}$ and $\text{SE}^{\text{Boot}}_{\text{median}}$ in Figure 6. The similarity between the bootstrap SE estimator and the Monte Carlo SE estimator demonstrates the stability of the TSSS estimator.

Next, we investigate the coverage rate of the proposed bootstrap-based PCIs in Algorithm 1. We consider randomly sampled locations $\mathbf{X}_i$'s with sample sizes $n = 900$, 2500 and $c_\sigma = 0.5, 0.75$ with mean function $\mu_3|\Omega$. We report the average coverage probabilities of the proposed PCIs using $B = 100$ wild bootstrap samples $\{\text{SE}^{\text{Boot}}_{rep}\}_{rep=1}^{100}$ through Algorithm 1 over 11,986 test grid locations. Table 4 reports the average coverage rates and

Table 4. Empirical coverage rates of 99% PCIs with the average widths (in parenthesis) in Simulation Study 3.

n	$\sigma = 0.50$	$\sigma = 0.75$
900	0.974 (0.490)	0.975 (0.734)
2500	0.978 (0.335)	0.980 (0.503)

widths of 99% PCIs defined in (8) under various scenarios. Across different σ, the coverage rate is stable, with the widths of PCIs increasing with higher σ. This demonstrates the stability of the proposed PCI. As the number of locations n increases, the coverage rates of PCIs improve, indicating the desired asymptotic properties of accurate uncertainty quantification of underlying mean functions. The mean coverage rate is reasonably close to the nominal confidence level 99%. The results of PCIs defined in (9) closely resemble that of (8) and are omitted here.

6. Data applications

6.1. Cortical surface fMRI data

In our first application, we apply the proposed TSSS method to estimate the mean neural activity of the brain using data from the Human Connectome Project (HCP) (Van Essen et al. 2013). Specifically, we use the cs-fMRI data from the Motor task study of the HCP 500-subject data release. In this study, participants are asked to tap their left or right fingers, squeeze their left or right foot, or move their tongue following visual cues. During the experiment, two runs of task fMRI scans were collected. One run was collected by scanning the brain from left to right (LR), and the other run was from right to left (RL) (Woolrich et al. 2001).

In this study, we use the RL scan run of a randomly sampled subject (Subject #100307) registered to the low-resolution Conte 69 standard mesh, which contains approximately 32,000 vertices per hemisphere (Fischl 2012; Jenkinson et al. 2012; Van Essen et al. 2012; Glasser et al. 2013). In addition, to evaluate the performance of TSSS in both the resting and task states, the first frame in the subject's resting state ($t = 0$s) and the 20th frame after left-hand tapping is onset ($t \approx 15$s) are studied. For each frame, we perform a 10-fold CV procedure to examine the prediction accuracy. For visualisation, we map the predicted values from the sphere to the mid-thickness brain surface, using the R package 'ciftiTools' (Pham et al. 2022).

For the TSSS method, the triangulation is constructed by first obtaining a uniform triangulation of the whole sphere. Next, we select only the spherical triangles that contain observations and adjust the boundary points to obtain a smoother domain shape. This step can be more complex as it requires careful consideration of the underlying data distribution and geometry of the domain. However, in our experience, the triangulation works well even without the adjustment step, as long as there are no triangles containing extremely sparse observations. Two settings are studied for TSSS: (i) $r = 1, d = d_{\text{CV}} = 5$ with a modest fine triangulation $N = 473$, as shown in Figure 7; and (ii) $r = 1, d = 3$ with a fine triangulation $N = 1913$ for each hemisphere. Here we adopted cross validation to select optimal degree d_{CV} in (i), where moderate triangulation is utilised. With finer triangulation in (ii),

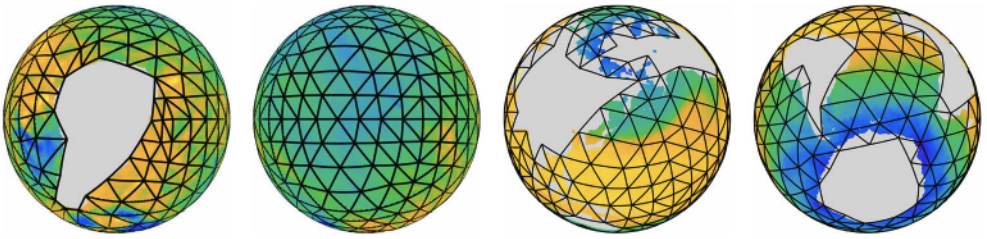

Figure 7. Triangulation for (a) Human Right Hemisphere from two viewpoints ($N = 473$); and (b) the ocean surface from two viewpoints ($N = 380$).

Table 5. The mean (and standard deviation) of CV-RMSPE in the HCP data application based on TPSOS, Tensor-Sphere and TSSS.

Left Hemisphere				Right Hemisphere			
Frame	Method	CV-RMSPE	Dim	Frame	Method	CV-RMSPE	Dim
1	TPSOS	471 (26)	2000	1	TPSOS	421 (11)	2000
1	Tensor-Sphere	528 (26)	2025	1	Tensor-Sphere	468 (15)	2025
1	TSSS (i)	412 (11)	2921	1	TSSS (i)	370 (8)	2921
1	TSSS (ii)	460 (15)	2003	1	TSSS (ii)	409 (10)	2008
20	TPSOS	467 (26)	2000	20	TPSOS	415 (12)	2000
20	Tensor-Sphere	527 (29)	2025	20	Tensor-Sphere	463 (14)	2025
20	TSSS (i)	407 (12)	2921	20	TSSS (i)	365 (9)	2921
20	TSSS (ii)	457 (16)	2003	20	TSSS (ii)	411 (10)	2008

The number of parameters for TPSOS and Tensor-Sphere is set to 2000 and 2025, respectively. The settings for TSSS are as follows: (i) $r = 1, d = d_{CV} = 5, N = 473$, yielding a dimensionality (Dim) of 2921; and (ii) $r = 1$, $d = 3$, $N = 1913$, yielding a dimensionality (Dim) of approximately 2000.

we hope to match the dimensions of TSSS with those of TPSOS and Tensor-Sphere for a fair comparison. Thus, $d = 3$ is used under finer triangulation.

Table 5 reports the mean and standard deviation of the cross-validated root mean squared prediction error (CV-RMSPE), $[N_k^{-1} \sum_{i:k[i]=k} \{Y_i - \widehat{m}^{-k}(\mathbf{X}_i)\}^2]^{-1/2}$, $k = 1, \ldots, 10$, for TSSS, TPSOS, and Tensor-Sphere, where the Kernel method is omitted due to its computation inefficiency. Here, $k[i]$ is the index of the fold that contains the ith observation, N_k is the cardinality of the kth fold, and $\widehat{m}^{-k}$ is the fitted mean function using data that exclude the kth fold. From Table 5, we observe that when the dimensions of TPSOS, TSSS and Tensor-Sphere are similar, around 2000, TSSS demonstrates a slight advantage over TPSOS and a more significant advantage compared to Tensor-Sphere. However, as the dimension of TPSOS increases to around 2900, the advantage of TSSS over both TPSOS and Tensor-Sphere becomes more pronounced. These results hold consistently for both hemispheres and different brain activity periods. Importantly, TSSS offers researchers greater flexibility in achieving their objectives. Depending on the desired result, researchers can adjust the parameters accordingly. Higher values of d and N can be employed to recover detailed signals, while finer triangulation can be used to incorporate prior domain knowledge. Moreover, by tuning the smoothness coefficient r, researchers can control the level of smoothness in the estimated function, aligning it with their expectations. TSSS thus provides a versatile tool for researchers with diverse needs and preferences.

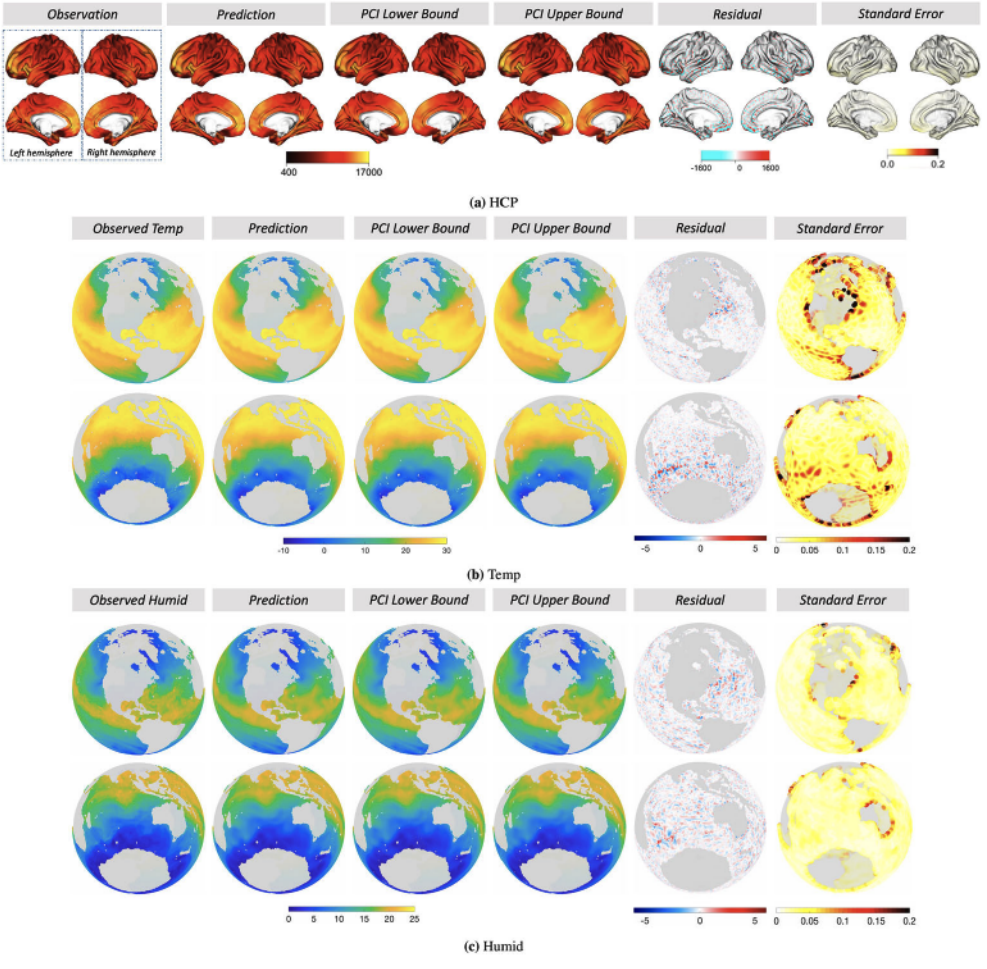

Figure 8. (a) The observed, predicted, lower and upper bound of 99% PCIs, residual values, along with the estimated standard error of the TSSS estimator for the first frame in HCP data application. Rows 1 and 2 depict two different viewpoints for each hemisphere. The TSSS settings used are: $r = 1, d = d_{\text{CV}} = 5$, and $N = 473$; (b) & (c): The observed, predicted lower and upper bound of 99% PCIs, residual values, along with the estimated standard error of the TSSS estimator for temperature and humidity in oceanic atmospheric data application. The TSSS settings used are: $r = 1, d = d_{\text{CV}} = 5$, and $N = 380$. (a) HCP. (b) Temp and (c) Humid.

Furthermore, we employ the bootstrap method outlined in Algorithm 1 to estimate the standard error of the TSSS estimator. To facilitate better visualisation, we present the standard error normalised by the maximum observed value, denoted as $\text{SE}^{\text{Boot}}/\max_i\{Y_i\}$. In Figure 8, we provide visual representations of the observed, predicted, and residual values for the first frame of both hemispheres using TSSS with setting (i). The estimated standard errors are consistently small across the entire domain. In addition, the 99% PCIs for the mean function are illustrated.

Table 6. The mean (and standard deviation) of the CV-RMSPEs in NCEI data application based on TSSS and TPSOS.

Item	Methods	CV-RMSPE	Dim	Item	Methods	CV-RMSPE	Dim
Temp	TPSOS	0.4913 (0.004)	2000	Humid	TPSOS	0.3808 (0.004)	2000
	Tensor-Sphere	0.5667 (0.008)	2025		Tensor-Sphere	0.4992 (0.011)	2025
	TSSS (i)	0.4684 (0.003)	2600		TSSS (i)	0.3522 (0.001)	2600
	TSSS (ii)	0.5171 (0.003)	1947		TSSS (ii)	0.3925 (0.001)	1947

The settings for TPSOS and Tensor-Sphere are 2000 and 2025, respectively. The TSSS settings used are: (i) $r = 1, d = d_{\text{CV}} = 5, N = 380$, yielding a dimensionality (Dim) of 2600; and (ii) $r = 1, d = 3, N = 1513$, yielding a dimensionality (Dim) of 1947.

6.2. Oceanic atmospheric data

In this application, we apply our method to the NOAA Ocean Surface Bundle Climate Data Record (CDR) from the National Centers for Environmental Information (NCEI) (Clayson et al. 2016). In particular, we consider the dataset SEAFLUX-OSB-CDR_V02R00_ATMOS_D20210831_C20211223.nc and focus on the specific humidity (Humid) and air temperature (Temp) on August 31, 2021. There are approximately one million lattice locations on the surface of Earth, among which approximately 485,000 locations are observed.

Again, we apply a 10-fold CV procedure to evaluate the performance of TSSS, TPSOS, and Tensor-Sphere. For TSSS, we use the settings (i) $d = d_{\text{CV}} = 5$, $r = 1$ with a modest fine triangulation $N = 380$, as shown in Figure 7; and (ii) $d = 3, r = 1$ with a fine triangulation $N = 1513$ for the ocean surface. The triangulation is constructed in the same way as described in Section 6.1. Results in Table 6 indicate that TSSS outperforms Tensor-Sphere in both settings and outperforms TPSOS in setting (i). Figure 8 displays the observed, predicted values, residuals of humidity and temperature, along with the estimated standard error of TSSS with setting (i). The residuals do not exhibit patterns, and the standard errors are relatively small with slightly larger values along the domain boundary due to fewer observations in those areas. In addition, the 99% PCIs for the mean function is also illustrated.

7. Summary and discussion

Motivated by ubiquitous surface-based data in various fields, we introduce a novel TSSS estimator for data that reside on complex domains on the sphere or any surface that can be properly mapped to and back from the sphere. By combining penalised splines with triangulation techniques on spherical surfaces, TSSS efficiently handles complex spherical domains while offering enhanced computational efficiency. Firstly, TSSS provides enhanced flexibility in capturing complex spatial patterns, allowing for accurate modelling of intricate variations in the data. By employing the 'domain aware' splines, TSSS avoids leakage or boundary effects, ensuring accurate estimation. This distinguishes TSSS from its competitors (e.g. Tensor-Sphere, TPSOS, and kernel-based methods). Secondly, TSSS offers improved computational efficiency compared to the kernel-based approach, making it more scalable for large-scale datasets. Kernel-based approaches rely on observed locations as kernel centers, which may not efficiently approximate localised or rapidly changing functions. In contrast, TSSS constructs basis functions based on triangulated domains and

avoids the need for excessive basis functions. By adaptively refining the triangulation and utilising spline basis functions with a flexible degree and smoothness, TSSS can better capture the complexity of the underlying signal. Thirdly, we provide rigorous theoretical guarantees and an implementation algorithm for the uncertainty quantification of TSSS. Specifically, Theorem 3.2 demonstrates the asymptotic normality of the TSSS estimator, and Algorithm 1 introduces a wild Bootstrap algorithm to obtain the standard error map of the TSSS estimator. All of these advantages have been demonstrated through extensive numerical experiments.

The studies in this paper also open opportunities for further research. TSSS usually requires well-constructed triangulations that accurately approximate the shape of domains; however, determining whether to connect isolated subdomains can be challenging and may need human input. To overcome this challenge, researchers could develop automated triangulation procedures to construct triangulations for TSSS, allowing easy tuning of user configurations. Moreover, inspired by Basna et al. (2022), there is a potential to explore function-dependent triangulation methods in functional data settings. In particular, random samples of functional data can be used to customise triangulations using data-specific features.

Another interesting future research direction is to advance TSSS to accommodate datasets that possess physical constraints, such as non-negativity (Baramidze and Lai 2018; Kim et al. 2021; Zhang, Zhou, Wang, et al. 2022) and shape constraints (Wang, Xue, et al. 2020; Fang et al. 2022). Incorporating non-negativity constraints into the TSSS framework can broaden the method's applicability to datasets where negative values are physically infeasible or meaningless. Moreover, integrating constraints (e.g. monotonicity, convexity, symmetry, or specific geometric patterns) into the TSSS framework can provide additional regularisation to ensure that the estimated function adheres to predefined physical constraints. TSSS can also be extended to handle longitudinal or time series data observed on spherical domains. Allowing researchers to model the dynamics of surface-based data over time could offer essential insights, such as estimating the temporal change of Earth's atmospheric properties. Furthermore, in functional regression where surface-based imaging serves as functional responses or covariates (Lila and Aston 2020), TSSS can be used as the first smoothing step.

It is also of immense interest to extend the proposed TSSS method to the hypersphere setting. Hypersphere surface data, or 'directional data', are widely observed in cell cycle data analysis (Schafer 1998), the shape analysis of manifold-valued data (Lin et al. 2017), compositional household expenditure (Scealy and Welsh 2017), and gene expression clustering (Ding and Regev 2021). In these studies, hypersphere-valued data $\mathbf{x} \in \mathbb{S}^d$ are considered the response of interest, and their traits carry information. Although our framework focuses more on the estimation of the mean function with $\mathbf{x} \in \mathbb{S}^d$ being the covariates, we can utilise dimension reduction methods from directional data literature to address the high-dimension setting.

In addition to the location variable $\mathbf{x}$, datasets frequently incorporate explanatory variables $\{z_k, k = 1, \ldots, p\}$, which may also contribute to the model. To account for this, we can utilise the generalised additive mean structure introduced in Yu et al. (2020), $g\{\mu(\mathbf{z}, \mathbf{x})\} = \sum_{k=1}^{p} f_k(z_k) + m(\mathbf{x})$. Here, g can be any link function, and $m(\cdot), f_k(\cdot)$ can be estimated nonparametrically depending on the assumption of the function space of

$m(\cdot), f_k(\cdot)$. Another potential approach is the generalised spatially varying coefficient models (Kim and Wang 2021). These modelling frameworks offer researchers the opportunity to explore the relationship between local features and responses of interest that reside on a surface.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

Guannan Wang's research was supported in part by the National Institutes of Health grant 1R01AG085616 and the Simons Foundation collaboration grant #963447. Ming-Jun Lai's research was partially supported by the Simons Foundation collaboration grant #864439. Lily Wang's research was partially supported by the National Institutes of Health grant 1R01AG085616 and the National Science Foundation grant DMS-2426173. The HCP data were provided by the Human Connectome Project, WU-Minn Consortium (Principal Investigators: David Van Essen and Kamil Ugurbil; 1U54MH091657) funded by the 16 NIH Institutes and Centers or Neuroscience Research; and by the McDonnell Center for Systems Neuroscience at Washington University. The Ocean Near Surface Atmospheric Properties CDR was acquired from NOAA's National Centers for Environmental Information (http://www.ncdc.noaa.gov). We thank the Associate Editor and the reviewers for their valuable input, which has significantly contributed to the improvement of this work.

ORCID

Lily Wang http://orcid.org/0000-0001-8432-9986

References

Abrial, P., Moudden, Y., Starck, J.L., Fadili, J., Delabrouille, J., and Nguyen, M. (2008), 'CMB Data Analysis and Sparsity', *Statistical Methodology*, 5, 289–298.

Alfeld, P., Neamtu, M., and Schumaker, L.L. (1996a), 'Bernstein–Bézier Polynomials on Spheres and Sphere-like Surfaces', *Computer Aided Geometric Design*, 13, 333–349.

Alfeld, P., Neamtu, M., and Schumaker, L.L. (1996b), 'Fitting Scattered Data on Sphere-like Surfaces Using Spherical Splines', *Journal of Computational and Applied Mathematics*, 73, 5–43.

Baramidze, V. and M.J. Lai. 2005. 'Error Bounds for Minimal Energy Interpolatory Spherical Splines', in *Approximation Theory XI*, eds. C.K. Chui, M. Neamtu, and L.L. Schumaker. Gatlinburg: Nashboro Press, pp. 25–50.

Baramidze, V., and Lai, M.J. (2011), 'Convergence of Discrete and Penalized Least Squares Spherical Splines', *Journal of Approximation Theory*, 163, 1091–1106.

Baramidze, V., and Lai, M.J. (2018), 'Nonnegative Data Interpolation by Spherical Splines', *Journal of Computational and Applied Mathematics*, 342, 463–477.

Baramidze, V., Lai, M.J., and Shum, C.K. (2006), 'Spherical Splines for Data Interpolation and Fitting', *SIAM Journal on Scientific Computing*, 28, 241–259.

Basna, R., Nassar, H., and Podgórski, K. (2022), 'Data Driven Orthogonal Basis Selection for Functional Data Analysis', *Journal of Multivariate Analysis*, 189, 104868.

Brodoehl, S., Gaser, C., Dahnke, R., Witte, O.W., and Klingner, C.M. (2020), 'Surface-based Analysis Increases the Specificity of Cortical Activation Patterns and Connectivity Results', *Scientific Reports*, 10, 1–13.

Cao, F., Lin, S., Chang, X., and Xu, Z. (2013), 'Learning Rates of Regularized Regression on the Unit Sphere', *Science China Mathematics*, 56, 861–876.

Chang, T., Ko, D., Royer, J.Y., and Lu, J. (2000), 'Regression Techniques in Plate Tectonics', *Statistical Science*, 15, 342–356.

Clayson, C.A., Brown, J., and Program, N.C. (2016), 'NOAA Climate Data Record Ocean Surface Bundle (OSB) Climate Data Record (CDR) of Ocean Heat Fluxes, Version 2', Technical Report, NOAA National Center for Environmental Information.

Cole, M., Murray, K., St-Onge, E., Risk, B., Zhong, J., Schifitto, G., Descoteaux, M., and Zhang, Z. (2021), 'Surface-Based Connectivity Integration: An Atlas-free Approach to Jointly Study Functional and Structural Connectivity', *Human Brain Mapping*, 42, 3481–3499.

De Loera, J., Rambau, J., and Santos, F. (2010), *Triangulations: Structures for Algorithms and Applications* (Vol. 25), Berlin Heidelberg: Springer-Verlag.

Ding, J., and Regev, A. (2021), 'Deep Generative Model Embedding of Single-cell RNA-Seq Profiles on Hyperspheres and Hyperbolic Spaces', *Nature Communications*, 12, 2554.

Eilers, P.H., and Marx, B.D. (2021), *Practical Smoothing: The Joys of P-Splines*, Cambridge: Cambridge University Press.

Fang, Y., Xue, L., Martins-Filho, C., and Yang, L. (2022), 'Robust Estimation of Additive Boundaries with Quantile Regression and Shape Constraints', *Journal of Business & Economic Statistics*, 40, 615–628.

Fischl, B. (2012), 'FreeSurfer', *NeuroImage*, 62, 774–781.

Fischl, B., Sereno, M.I., and Dale, A.M. (1999), 'Cortical Surface-based Analysis: II: Inflation, Flattening, and a Surface-based Coordinate System', *NeuroImage*, 9, 195–207.

Glasser, M.F., Sotiropoulos, S.N., Wilson, J.A., Coalson, T.S., Fischl, B., Andersson, J.L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J.R., et al. (2013), 'The Minimal Preprocessing Pipelines for the Human Connectome Project', *NeuroImage*, 80, 105–124.

Gu, X., Wang, Y., Chan, T.F., Thompson, P.M., and Yau, S.T. (2004), 'Genus Zero Surface Conformal Mapping and Its Application to Brain Surface Mapping', *IEEE Transactions on Medical Imaging*, 23, 949–958.

Hall, P., and Horowitz, J. (2013), 'A Simple Bootstrap Method for Constructing Nonparametric Confidence Bands for Functions', *The Annals of Statistics*, 41, 1892–1921.

Huang, J.Z. (2003), 'Local Asymptotics for Polynomial Spline Regression', *The Annals of Statistics*, 31, 1600–1635.

Jenkinson, M., Beckmann, C.F., Behrens, T.E., Woolrich, M.W., and Smith, S.M. (2012), 'FSL', *NeuroImage*, 62, 782–790.

Kim, M., and Wang, L. (2021), 'Generalized Spatially Varying Coefficient Models', *Journal of Computational and Graphical Statistics*, 30, 1–10.

Kim, M., Wang, L., and Zhou, Y. (2021), 'Spatially Varying Coefficient Models with Sign Preservation of the Coefficient Functions', *Journal of Agricultural, Biological and Environmental Statistics*, 26, 367–386.

Kinoshita, K., and Nakamura, H. (2003), 'Identification of Protein Biochemical Functions by Similarity Search Using the Molecular Surface Database EF-site', *Protein Science*, 12, 1589–1595.

Lai, M.J., and Schumaker, L.L. (2007), *Spline Functions on Triangulations* (1st ed.), Cambridge: Cambridge University Press.

Lai, M., Shum, C., Baramidze, V., and Wenston, P. (2008), 'Triangulated Spherical Splines for Geopotential Reconstruction', *Journal of Geodesy*, 83, 695–708.

Lai, M.J., and Wang, L. (2013), 'Bivariate Penalized Splines for Regression', *Statistica Sinica*, 23, 1399–1417.

Li, Y., and Ruppert, D. (2008), 'On the Asymptotics of Penalized Splines', *Biometrika*, 95, 415–436.

Lila, E., and Aston, J.A. (2020), 'Statistical Analysis of Functions on Surfaces, with An Application to Medical Imaging', *Journal of the American Statistical Association*, 115, 1420–1434.

Lin, L., Zhu, H., and Dunson, D.B. (2017), 'Extrinsic Local Regression on Manifold-valued Data', *Journal of the American Statistical Association*, 112, 1261–1273.

Lindgren, F., and Rue, H. (2015), 'Bayesian Spatial Modelling with R-INLA', *Journal of Statistical Software*, 63, 1–25.

Lyche, T., and Schumaker, L.L. (2000), 'A Multiresolution Tensor Spline Method for Fitting Functions on the Sphere', *SIAM Journal on Scientific Computing*, 22, 724–746.

Mammen, E. (1993), 'Bootstrap and Wild Bootstrap for High Dimensional Linear Models', *The Annals of Statistics*, 21, 255–285.

Mark, D.B., Otfried, C., Marc, V.K., and Mark, O. (2008), *Computational Geometry Algorithms and Applications*, Berlin Heidelberg: Spinger-Verlag.
Marzio, M.D., Panzera, A., and Taylor, C.C. (2019), 'Nonparametric Rotations for Sphere-sphere Regression', *Journal of the American Statistical Association*, 114, 466–476.
Mejia, A.F., Yue, Y., Bolin, D., Lindgren, F., and Lindquist, M.A. (2020), 'A Bayesian General Linear Modeling Approach to Cortical Surface FMRI Data Analysis', *Journal of the American Statistical Association*, 115, 501–520.
Mu, J., Wang, G., and Wang, L. (2018), 'Estimation and Inference in Spatially Varying Coefficient Models', *Environmetrics*, 29, e2485.
Neamtu, M., and Schumaker, L.L. (2004), 'On the Approximation Order of Splines on Spherical Triangulations', *Advances in Computational Mathematics*, 21, 3–20.
Persson, P.O., and Strang, G. (2004), 'A Simple Mesh Generator in MATLAB', *SIAM Review*, 46, 329–345.
Pham, D.D., Muschelli, J., and Mejia, A.F. (2022), 'ciftiTools: A Package for Reading, Writing, Visualizing, and Manipulating CIFTI Files in R', *NeuroImage*, 250, 118877.
Rakotosaona, M.J., Aigerman, N., Mitra, N.J., Ovsjanikov, M., and Guerrero, P. (2021), 'Differentiable Surface Triangulation', *ACM Transactions on Graphics (TOG)*, 40, 1–13.
Ramsay, T. (2002), 'Spline Smoothing over Difficult Regions', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 64, 307–319.
Ruppert, D. (2002), 'Selecting the Number of Knots for Penalized Splines', *Journal of Computational and Graphical Statistics*, 11, 735–757.
Scealy, J.L., and Welsh, A.H. (2017), 'A Directional Mixed Effects Model for Compositional Expenditure Data', *Journal of the American Statistical Association*, 112, 24–36.
Schafer, K. (1998), 'The Cell Cycle: A Review', *Veterinary Pathology*, 35, 461–478.
Seeley, R.T. (1966), 'Spherical Harmonics', *The American Mathematical Monthly*, 73, 115–121.
Smith, A., Lott, N., and Vose, R. (2011), 'The Integrated Surface Database: Recent Developments and Partnerships', *Bulletin of the American Meteorological Society*, 92, 704–708.
Stone, C.J. (1982), 'Optimal Global Rates of Convergence for Nonparametric Regression', *The Annals of Statistics*, 10, 1040–1053.
Sun, H., Hua, Z., Ren, J., Zou, S., Sun, Y., and Chen, Y. (2022), 'Matrix Completion Methods for the Total Electron Content Video Reconstruction', *The Annals of Applied Statistics*, 16, 1333–1358.
Van Essen, D.C., Glasser, M.F., Dierker, D.L., Harwell, J., and Coalson, T. (2012), 'Parcellations and Hemispheric Asymmetries of Human Cerebral Cortex Analyzed on Surface-based Atlases', *Cerebral Cortex*, 22, 2241–2262.
Van Essen, D.C., Smith, S.M., Barch, D.M., Behrens, T.E., Yacoub, E., Ugurbil, K., and Consortium, W.M.H. (2013), 'The WU-Minn Human Connectome Project: An Overview', *NeuroImage*, 80, 62–79.
Wahba, G. (1981), 'Spline Interpolation and Smoothing on the Sphere', *SIAM Journal on Scientific and Statistical Computing*, 2, 5–16.
Wahba, G. (1982), 'Erratum: Spline Interpolation and Smoothing on the Sphere', *SIAM Journal on Scientific and Statistical Computing*, 3, 385–386.
Wang, L., and Lai, M.J. (2019), 'Triangulation', R package version 1.0. Available at https://github.com/funstatpackages/Triangulation.
Wang, L., Wang, G., Lai, M.J., and Gao, L. (2020), 'Efficient Estimation of Partially Linear Models for Data on Complicated Domains by Bivariate Penalized Splines over Triangulations', *Statistica Sinica*, 30, 347–369.
Wang, S., Wang, H., Zhao, Y., Cao, G., and Li, Y. (2023), 'Empirical Likelihood Ratio Tests for Varying Coefficient Geo Models', *Statistica Sinica*, 33, 1–26.
Wang, L., Xue, L., and Yang, L. (2020), 'Estimation of Additive Frontier Functions with Shape Constraints', *Journal of Nonparametric Statistics*, 32, 262–293.
Wood, S.N. (2003), 'Thin Plate Regression Splines', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 65, 95–114.
Wood, S.N. (2006), 'Low-rank Scale-invariant Tensor Product Smooths for Generalized Additive Mixed Models', *Biometrics*, 62, 1025–1036.

Wood, S.N. (2017), 'mgcv', R package version 1.8-23. Available at https://cran.r-project.org/web/packages/mgcv/index.html.

Wood, S.N., Bravington, M.V., and Hedley, S.L. (2008), 'Soap Film Smoothing', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 70, 931–955.

Woolrich, M.W., Ripley, B.D., Brady, M., and Smith, S.M. (2001), 'Temporal Autocorrelation in Univariate Linear Modeling of FMRI Data', *NeuroImage*, 14, 1370–1386.

Yu, S., Wang, G., Wang, L., Liu, C., and Yang, L. (2020), 'Estimation and Inference for Generalized Geoadditive Models', *Journal of the American Statistical Association*, 115, 761–774.

Zhang, T., Zhou, Y., Wang, L., Zhao, K., and Zhu, Z. (2022), 'Estimating 1 km Gridded Daily Air Temperature Using a Spatially Varying Coefficient Model with Sign Preservation', *Remote Sensing of Environment*, 277, 113072.

Zhang, T., Zhou, Y., Zhao, K., Zhu, Z., Asrar, G.R., and Zhao, X. (2022), 'Gap-filling MODIS Daily Aerosol Optical Depth Products by Developing a Spatiotemporal Fitting Algorithm', *GIScience & Remote Sensing*, 59, 762–781.

Zhang, Z., Wu, Y., Xiong, D., Ibrahim, J.G., Srivastava, A., and Zhu, H. (2023), 'LESA: Longitudinal Elastic Shape Analysis of Brain Subcortical Structures', *Journal of the American Statistical Association*, 118, 1–15.