Published on *Applied Numerical Mathematics* (2015), 90, 111-131.
Publisher's version available at http://dx.doi.org/10.1016/j.apnum.2014.10.007

A mesh simplification strategy for a spatial regression analysis over the cortical surface of the brain

Franco Dassi, Bree Ettinger, Simona Perotto and Laura M. Sangalli

MOX– Modellistica e Calcolo Scientifico
Dipartimento di Matematica "F. Brioschi"
Politecnico di Milano
via Bonardi 9, 20133 Milano, Italy

Abstract

We present a new mesh simplification technique developed for a statistical analysis of a large data set distributed on a generic complex surface, topologically equivalent to a sphere. In particular, we focus on an application to cortical surface thickness data. The aim of this approach is to produce a simplified mesh which does not distort the original data distribution so that the statistical estimates computed over the new mesh exhibits good inferential properties. To do this, we propose an iterative technique that, for each iteration, contracts the edge of the mesh with the lowest value of a cost function. This cost function takes into account both the geometry of the surface and the distribution of the data on it. After the data are associated with the simplified mesh, they are analyzed via a spatial regression model for non-planar domains. In particular, we resort to a penalized regression method that first conformally maps the simplified cortical surface mesh into a planar region. Then, existing planar spatial smoothing techniques are extended to non-planar domains by suitably including the flattening phase. The effectiveness of the entire process is numerically demonstrated via a simulation study and an application to cortical surface thickness data.

Keywords: Iterative edge contraction, conformal flattening maps, regression analysis, statistical analysis of complex data, cortical surface thickness data

1 Introduction and motivation

In this paper, we develop a technique to analyze large data sets lying on complicated two-dimensional manifolds. In particular, we are interested in analyzing data observed over the cortical surface of the brain, a two-dimensional manifold with many folds and creases, constituting the outermost part of the brain. The data of interest are the hemodynamic signals associated with neural activity on the cerebral cortex, or the measurements of the cerebral cortex thickness (i.e., the thickness of grey matter tissue). From a medical viewpoint, the study of these data is of relative importance to better understand brain functions and the underlying mechanics of brain diseases. For instance, the thickness of the cerebral cortex changes over time and is linked, in the medical literature, to the pathology of many neurological disorders such as autism, Alzheimer's disease and schizophrenia [19]. Cortical surface data are obtained from reconstructions of the output of various types of magnetic resonance imaging (MRI) (see, e.g.,[5]). Figure 1 shows an example of thickness data studied in [3] and [4]. On the left, a cortical surface mesh is provided, while, on the right, we have the corresponding thickness measurements at each node of the mesh represented as a color map, obtained by linearly interpolating the measurements at the mesh nodes. Due to the folded nature of the cerebral cortex, the mesh generation process is a complex multistep procedure that results in a very large data set (often more than 10^6 nodes). Moreover, these data sets are usually characterized by noise in both the node locations and the data measurements. Advanced methods for modeling data spatially distributed over these manifolds are consequently required.

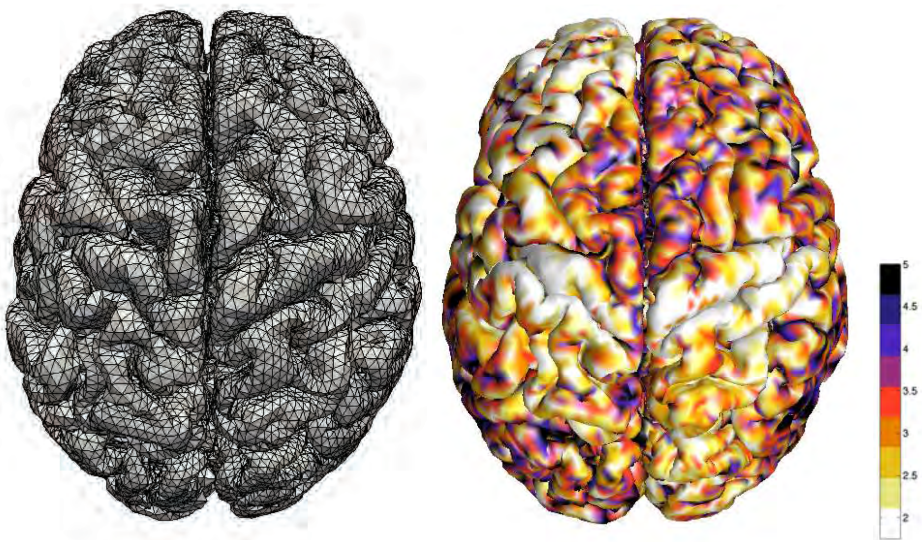

Figure 1: Example of a cortical thickness data set: a cortical surface mesh with 40962 nodes (left); color map of the cortical thickness (right). The data and the Matlab code used to build the color map are available at `http://www.stat.wisc.edu/~mchung/softwares/hk/hk.html`

We propose an efficient technique to analyze large noisy data sets associated with triangular meshes of complicated non-planar geometries. To do this, we couple a mesh simplification technique with a spatial regression method for analyzing data on non-planar domains. The motivation for the simplification procedure is to reduce the computational effort associated with the statistical analysis of the large data sets that are typical in cortical surface applications. The proposed simplification procedure is designed specifically for producing a mesh that does not distort the original data distribution and is optimal for a statistical analysis of the data. In particular, through an iterative procedure, we take into account both the geometry of the mesh and the data distribution over it. The simplified geometry is generated in a way such that the analysis of the data associated with it should have statistical estimates with good inferential properties. For the data analysis, we resort to the Spatial Regression model for Non-Planar domains (SR-NP) developed in [9]. The SR-NP approach smooths the noisy data by minimizing a sum of squared error functional with a roughness penalty term involving the Laplace-Beltrami operator associated with the non-planar domain. The estimation problem on the surface is then appropriately recast over a planar domain via a conformal map. In the planar domain, existing spatial smoothing techniques are generalized by suitably taking into account the flattening of the domain. Notice that mapping to a planar domain would also allow for a statistical analysis across patients, similar to mapping to a reference brain [4]. In fact, via the SR-NP method, patient-specific estimates can all be mapped to a common planar domain where, after suitable registration among patients, comparisons across patients can be made. Nevertheless, the development of full inferential and uncertainty quantification tools for these population studies is outside the scope of this current paper. However, the mesh simplification proposed in this paper lays a foundation for these tools. The original application for the SR-NP method was modeling hemodynamic forces on the carotid artery (or on any manifold topologically equivalent to a cylinder). Since the cortical surface can be represented by a topological sphere, the conformal map has to be modified accordingly. To accomplish this, we implement a modified version of the conformal map suggested in [1]. The modification we introduce provides robust results when flattening some of the undesirable triangulations generated by the segmentation and extraction procedures [5].

Alternative approaches proposed in the literature chose different methods for containing the computational cost associated with the analysis of large cortical surface data sets. The nearest neighbor averaging technique developed in [12] is an iterative technique that smooths the variable of interest observed at each vertex of the mesh by suitably averaging this value with the ones observed at the neighboring vertices. The averaging process is repeated several times to create a smoothing effect. Although this technique is practical for smoothing data over the cortical surface, more sophisticated methods have been developed to build inferential tools that measure the uncertainty of the resulting estimates. For example, a recent method proposed in [19] identifies the mesh with a weighted graph. Then, the data associated with the mesh is smoothed by tuning the local support around each vertex of the graph via a graph Laplacian. Another example of a smoothing technique for neuroimaging applications is the Iterative Heat Kernel (IHK) smoothing introduced in [4]. This geodesic distance based kernel smoothing method solves the Laplace-

Beltrami eigenvalue problem directly on the surface to construct a basis for the heat kernel on the cortical surface. Then, a finite number of these basis functions are used in the expansion of the heat kernel. In particular, a smoothing window is defined around each data point. The size of the smoothing window is identified by a parameter called the bandwidth. Finally, the number of terms used in the Fourier series expansion of the heat kernel is properly adjusted via an iterative algorithm.

We note that both the SR-NP method and IHK smoothing employ the Laplace-Beltrami operator of the cortical surface, however in very different ways. In the SR-NP method, the Laplace-Beltrami operator is used to control the roughness of the solution, while IHK smoothing resorts to the Laplace-Beltrami operator to create a basis for the heat kernel directly on the cortical surface. As a second relevant difference, IHK smoothing is not currently designed to include space varying covariates. On the contrary, the method we propose has the desired inferential tools as well as the ability to include space varying covariates. In Sections 4-5 we numerically demonstrate the effectiveness of our method to highlight its very good performance and comparative advantages with respect to alternative methods.

The paper is organized as follows. Section 2 describes the mesh simplification strategy and explains how the original data points are associated with the simplified mesh. Section 3 gives the details of the SR-NP method and introduces a new flattening map for cortical surfaces. Section 4 is devoted to simulation studies, while Section 5 applies the proposed procedure to cortical surface data. Finally, Section 6 draws some conclusions and states future research directions.

2 The mesh simplification strategy

A cortical surface mesh is usually composed by a large number of vertices resulting in a high computational cost for the subsequent statistical analysis. The idea is to reduce this drawback via a surface mesh simplification process.

Consider a triangulated surface Γ_h embedded in $\mathbb{R}^3$ where a scalar data value z has been observed at each node of the mesh via the values z_j, for $j = 1, \dots, n$. For example, the scalar data in Figure 1 (right), are the values of the cortical surface thickness measured at each node of the mesh on the left. Hence, the original data locations coincide with the nodes of the original mesh. Our goal is to build a new mesh Γ'_h with m vertices, where $m \ll n$, while properly associating the original scalar data values with this new mesh. Due to the highly folded nature of the brain, this association is really involved. The proposed simplification method carefully tracks the origin of the data and correctly associates it with the new mesh. For instance, in the sulci of the brain, the data is correctly associated with the side of the sulcus it comes from instead of with the closest side (see Figure 2).

Surface mesh simplification has received a lot attention in the literature. Several different strategies have been presented to achieve this goal. They can be categorized as follows:

- VERTEX DECIMATION: this algorithm iteratively removes a vertex of the

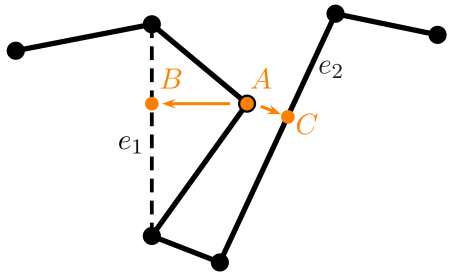

Figure 2: Cross-section of an original mesh (solid lines); the new mesh (dashed line) replaces the two segments sharing the vertex A with the segment e_1. In the new configuration, the correct projection of the data point A is B and not C, even though C is closer to A.

mesh and all the adjacent faces by looking at the distance from the vertex to the average plane identified by its neighborhood; then, the resulting hole is remeshed (see [28] for more details).

- VERTEX CLUSTERING: in this case a bounding box is placed around the original mesh and it is divided via an octree algorithm; then, in each cell of the octree, the vertices are clustered together into a single vertex and the faces of the mesh are suitably updated (see [26] and [18]).

- ITERATIVE EDGE CONTRACTION: the mesh is iteratively simplified by contracting the edges (see [14, 25] and [15]); an extension of this strategy is proposed in [10], where an additional contraction is made between any two vertices which are too close to one another and not necessary connected by an edge. The validity of an edge contraction can be based on other properties of the mesh, for instance, on the preservation of a homeomorphism in some lower dimensional environment as in [6].

We propose a mesh simplification process based on this last approach. In more detail, we develop an iterative technique such that, for each iteration, we contract an edge of the mesh Γ_h via a properly defined edge cost function. The strategies currently available in the literature usually take into account only the geometric aspects of the simplification process, i.e., they find a new mesh Γ_h' with fewer elements that approximates the mesh Γ_h in a best way possible. Here, we aim at enriching the geometric criterion with data information. In particular, the novelty of the proposed simplification strategy is twofold: associating the scalar data values of the original mesh with the new mesh Γ_h', and considering their displacement and distribution on the simplified mesh during the contraction process. To do this, we drive the simplification process via an edge cost function that takes into account both the geometric fitting of the domain and the association of the data points with the new mesh. In particular, for the data association, we analyze the displacement of the data points from their original positions to their new locations on Γ_h' as well as the evenness of the resulting data distribution over Γ_h'. Our aim is to control these two data properties to ensure quality statistical estimates with good inferential properties. The proposed algorithm has been developed for closed surfaces with genus zero (i.e, with no holes), but it can be extended to high genus or open surfaces by properly accounting for the edges in the neighborhood of the hole or boundary during the contraction process.

To describe the simplification process, we introduce the following notation. During the iterative contraction process, we consider an edge e with endpoints v_1 and v_2. Then, we replace the vertices v_1 and v_2 with a single new vertex v^*, which, a priori, may coincide with the end points v_1 and v_2 (see Figure 3). In general, we say that the edge e is contracted into the vertex v^*. For each contraction, we define the following sets:

- $\mathcal{T}_{\text{edge}}$, the set of triangles connected to edge e, i.e., the set of triangles that have either v_1 or v_2 as a vertex (the patch of elements in Figure 3, left).
- $\mathcal{T}_{\text{new}}$, the set of triangles in $\mathcal{T}_{\text{edge}}$ after the contraction (the patch of elements in Figure 3, right).

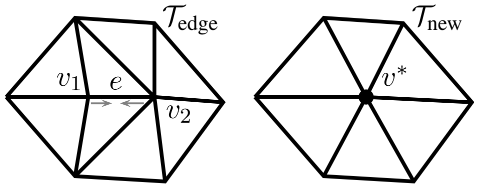

Figure 3: Contraction of the edge e into the vertex v^*.

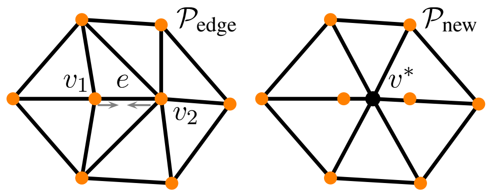

Figure 4: Data point sets during the contraction process (in this case $\mathcal{P}_{\text{orig}} \equiv \mathcal{P}_{\text{edge}}$).

Finally, we denote by $\mathcal{P}_{\text{orig}}$ the original location of the data points, while the set of all the data points projected on the triangles in $\mathcal{T}_{\text{edge}}$ is denoted by $\mathcal{P}_{\text{edge}}$. After the contraction, the data points are projected onto the triangles in $\mathcal{T}_{\text{new}}$ and denoted by $\mathcal{P}_{\text{new}}$ (see Figure 4). Now, the data points do not necessarily coincide with the mesh nodes.

2.1 Preliminary geometric considerations

The contraction of a generic edge e can lead to undesired topological artifacts. In the initial mesh, the triangles are oriented so that the corresponding normals are pointing outward with respect to the surface. When an edge is contracted may produce an inverted triangle, i.e., a triangle whose normal points inward. The inward normal changes the orientation of the triangle yielding a triangle with negative area (see Figure 5).

The situation becomes even more complicated on geometries with folds such as the cortical surface. In certain configurations, the location of v^* can also create a self-intersection of the mesh as shown in Figure 6. To overcome these problems, we have developed a series of specific tests that control the undesired features. In particular, to prevent

- the inversion of triangles, we check the normals of the triangles constituting $\mathcal{T}_{\text{new}}$. After the contraction, these normals may change direction and orientation. The angle between the corresponding normals before and after the contraction has to be strictly less than $\pi/2$;
- the self-intersection of the newly generated triangles with neighboring elements, we resort to a series of triangle-triangle intersection tests developed in [22].

If the contraction of the edge e into the vertex v^* passes the two tests above, we refer to the edge e as a *valid edge*.

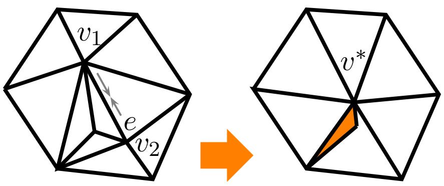

Figure 5: Example of a contraction of an edge that produces an inverted triangle: when the edge e on the left is contracted into the vertex v^*, the inverted colored triangle on the right is generated.

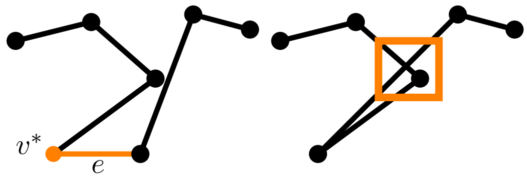

Figure 6: Example of self-intersection due to the nature of the sulci. The algorithm tries to contract the edge e into the node v^* (left), but this operation yields a self intersection (right).

2.2 The edge cost function

In order to select the contraction to perform during a given iteration of the mesh simplification procedure, we introduce the notion of *contraction cost*. This value takes into account both the geometric approximation of the mesh and the association of the original data with the new mesh. Hence, we define the contraction cost function to be

$$c(e, v^*) := \alpha c_{\text{geo}}(e, v^*) + \beta c_{\text{data}}(e, v^*),$$

where e is a generic edge of the mesh and v^* is the node that replaces the edge e, while $c_{\text{geo}}(e, v^*)$ and $c_{\text{data}}(e, v^*)$ represent the geometric cost function and the data cost function, respectively. In particular, $c_{\text{geo}}(e, v^*)$ is a function that associates with the edge e and the vertex v^* a positive real number that measures the loss of geometric accuracy produced by the contraction of e into v^*. Similarly, $c_{\text{data}}(e, v^*)$

measures the loss of good properties for the subsequent statistical analysis in terms of the displacement and distribution of the data points over the new mesh. The weights $\alpha, \beta \in \mathbb{R}^+$ balance each function's contribution to the overall contraction cost (possible choices for α and β are given in Section 2.3). The goal of the next two sections is to explain how to compute $c_{\text{geo}}(e, v^*)$ and $c_{\text{data}}(e, v^*)$, respectively.

2.2.1 The geometric cost function

In order to approximate the geometry of the mesh, we use the theory provided in [10]. Here, we briefly recall its basic concepts and explain how we exploit them.

Each vertex v of the mesh can be seen as the intersection of a set of planes. The error of a new vertex v^* with respect to these planes can be defined as the sum of squared distances to these planes, i.e., as

$$\sum_{\varrho \in \pi_v} (\varrho^t v^*)^2 \,, \tag{1}$$

where $\varrho = [a\, b\, c\, d]^t$ represents a generic plane in $\mathbb{R}^3$ defined by the equation $ax + by + cz + d = 0$, with $a^2 + b^2 + c^2 = 1$, and π_v is the set of planes identified by the triangles of the mesh sharing the vertex v. Note that, here, the vertex v^* is assigned to a vector in $\mathbb{R}^4$ where the last component is one, in order to properly define the scalar product $\varrho^t v^*$. Definition (1) leads us to introduce, for a generic vertex v, the symmetric matrix $Q_v := \sum_{\varrho \in \pi_v} \varrho\varrho^t \in \mathbb{R}^{4\times 4}$. Consequently, given the edge e with vertices v_1 and v_2, and the associated matrices Q_{v_1} and Q_{v_2}, we can define the symmetric matrix $Q_e := Q_{v_1} + Q_{v_2}$. Thus, for a generic vertex v^*, the quantity

$$v^{*t} Q_e v^* \,, \tag{2}$$

can be assumed to estimate the loss of geometric accuracy due to the contraction of the edge e into the node v^*. Following [10], during the mesh simplification process, for each edge e, we consider four possible different locations of the point v^*:

$$v_1 \,, \quad v_2 \,, \quad (v_1 + v_2)/2 \,, \text{ and } v_{\text{opt}},$$

where v_{opt} is the optimal position that minimizes the quantity (2). We consider three different configurations besides v_{opt} since this optimal position does not necessarily exist or it may produce an undesired configuration (see Section 2.1). Thus, for a valid edge e and an optimal location for v^*, we define the geometric cost for contracting the edge e into the node v^* by the quantity

$$c_{\text{geo}}(e, v^*) := v^{*t} Q_e v^* \,. \tag{3}$$

2.2.2 The data cost function

The actual novelty of the proposed algorithm lies in incorporating the data points into the simplification process. For the statistical analysis that follows the mesh simplification, it is crucial to properly take into account the association of the original data with the new mesh. Thus, to reduce the error with respect to using the original mesh, we attempt to control the displacement of the data locations when they are projected onto the new mesh. Another crucial property the new mesh needs to produce quality statistical estimates with good inferential properties is an equidistribution of the original data points over Γ'_h, i.e., to produce estimates that are robust and characterized by low bias (i.e., low systematic errors) each triangle should contain, a priori, the same quantity of information, independently of the size of the triangle. In order to evaluate the effectiveness of an edge contraction in the current mesh Γ'_h with respect to the resulting data associations, we consider

a) the *displacement* of the data points, i.e., the distance between the projected data locations and their original locations;

b) the *equidistribution* of the data points over the triangles of the new mesh Γ'_h, i.e., each triangle of Γ'_h should be associated with about the same number of data points. Equidistribution of the data should ensure that the quality of the inferences provided by the statistical estimates is uniform over the entire mesh.

To take care of both these aspects, we introduce two suitable cost functions, one for each desired feature. Thus the total data cost function is defined as

$$c_{\text{data}}(e, v^*) := \beta_1 c_{\text{disp}}(e,\, v^*) + \beta_2 c_{\text{equi}}(e,\, v^*),$$

where β_1 and β_2 are positive real numbers that properly weight the contribution of the data point displacement cost function, $c_{\text{disp}}(e,\, v^*)$, and of data distribution cost function, $c_{\text{equi}}(e,\, v^*)$, respectively. These two cost functions are rigorously defined below.

Before dealing with these two features of the mesh simplification process, let us make some further considerations about the data projection phase of the process. Although, the data points are orthogonally projected onto the simplified mesh Γ'_h, this projection is not straightforward on complicated surfaces such as the highly folded cortical surface. For example, in Figure 2, the correct location for the point A on the new mesh is the point B on the edge e_1 and not the point C on the edge e_2. Specifically, each point of Γ_h can be projected onto the new mesh Γ'_h in one of the following ways:

- to the face of a triangle of Γ'_h;
- to an edge between two triangles of Γ'_h;
- to a vertex of Γ'_h.

After the projection procedure, the data points are associated with their projection on Γ'_h.

Data displacement function When the edge e is contracted into the point v^*, we define the corresponding displacement cost function as

$$c_{\text{disp}}(e, v^*) := \max_{(p,q)\in\mathcal{P}_{\text{new}}\times\mathcal{P}_{\text{orig}}} \|p - q\|, \tag{4}$$

with $\|\cdot\|$ the Euclidean norm, which measures the maximum Euclidean distance between the orthogonally projected locations of the data points $\mathcal{P}_{\text{new}}$, and their original locations $\mathcal{P}_{\text{orig}}$ (see Figure 7). By minimizing the displacement of the data associations during the contraction process, we are able to reduce the error between the statistical estimates that use the original data points on Γ_h and the estimates based on the data points associated with the simplified mesh Γ'_h. Of course, this minimization step is properly constrained to avoid any incorrect associations such as the one discussed in Figure 2.

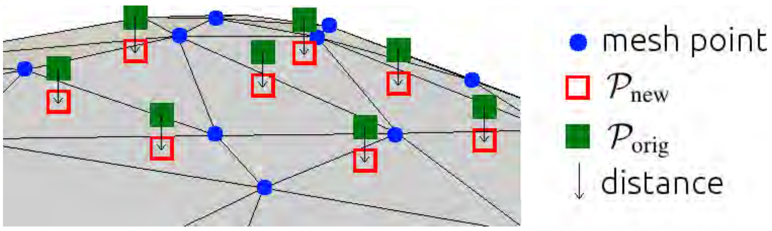

Figure 7: Surface during the simplification process: the distance between the projected locations of the data points, $\mathcal{P}_{\text{new}}$, and their original locations, $\mathcal{P}_{\text{orig}}$, is illustrated.

Data distribution function Our goal at the end of the simplification process is to obtain a new mesh where each element contains approximately the same number of data points, i.e., to equidistribute the data points in the new mesh. For this purpose, during the simplification process, we define the quantity of information associated with each triangle T via the number

$$N_T := n_f + \frac{1}{2}n_e + \frac{1}{\#(\mathcal{T}_{v_1})}n_1 + \frac{1}{\#(\mathcal{T}_{v_2})}n_2 + \frac{1}{\#(\mathcal{T}_{v_3})}n_3\,, \tag{5}$$

where n_f and n_e denote the number of data points associated with the face and the edges of the triangle T, respectively. For $j = 1, 2, 3$, n_j is the number of data points associated with v_j, the j-th vertex of T, $\mathcal{T}_{v_j}$ is the patch of elements associated with v_j and $\#(\mathcal{T}_{v_j})$ denotes the cardinality of the patch $\mathcal{T}_{v_j}$.

Moving from (5), we denote by $\overline{N}$ the mean value of N_T over the entire mesh before the current iteration of the simplification process takes place. Then, when contracting the edge e into v^*, we compute the quantity N_T for all the triangles in $\mathcal{T}_{\text{new}}$ and we evaluate the following distribution cost function

$$c_{\text{equi}}(e, v^*) := \frac{1}{\#(\mathcal{T}_{\text{new}})}\left(\sum_{T\in\mathcal{T}_{\text{new}}} (N_T - \overline{N})^2\right)\,.$$

For a contraction of the edge e into the vertex v^*, this value measures the variation in the distribution of the number of data points associated with triangles in $\mathcal{T}_{\text{new}}$ with respect to $\overline{N}$. Minimizing this variation should yield an even distribution of the data locations over the triangles in the final mesh. Notice that $c_{\text{equi}}(e,\, v^*)$ can also be expressed as

$$c_{\text{equi}}(e,\, v^*) := \frac{1}{\#(\mathcal{T}_{\text{new}})} \left(\sum_{T \in \mathcal{T}_{\text{new}}} (N_T - \overline{N}_{\text{cont}})^2 + (\overline{N}_{\text{cont}} - \overline{N})^2 \right), \tag{6}$$

where $\overline{N}_{\text{cont}}$ is the mean value of N_T over $\mathcal{T}_{\text{new}}$. By minimizing (6), we are reducing the variance of the distribution of the number of data points associated with each triangle in the patch $\mathcal{T}_{\text{new}}$, via the first term. While, with the second term, we are lowering the difference between the mean number of data associated with each triangle in $\mathcal{T}_{\text{new}}$ and the corresponding mean value computed over the entire mesh before the contraction. Moreover, after the contraction, we add a further check on each triangle of $\mathcal{T}_{\text{new}}$ to assure that the contraction does not produce an empty triangle, i.e., a triangle with no data associations.

The employment of $c_{\text{equi}}(e,\, v^*)$ during the contraction process, allows us to even out the uncertainty of the statistical estimates over the entire mesh Γ'_h. This increases the quality of the inferences provided by the statistical estimates. In more detail, in the presence of data evenly distributed throughout the mesh, the resulting pointwise confidence intervals for the estimates will all have about the same size. This means that the quality of the estimates will be uniform over the entire mesh, i.e, no region of the mesh will have a better estimate to the solution than other regions. The corresponding hypothesis tests will all have about the same power, allowing for consistent conclusions to be drawn over the whole mesh. This is extremely important for cortical surface applications where the interest lies in finding areas of activation or evidence of disease. The same level of uncertainty everywhere is necessary to produce clear and interpretable results.

2.3 Combination of the geometric and the data cost functions

The cost functions in (3), (4) and (6) may have different ranges depending on the data and the geometry. So, we normalize these three functions by their respective maxima. No change in notation is employed in the following for these normalized quantities.

Now, for each valid contraction of the edge e into the vertex v^*, we compute the cost $c(e, v^*)$ as the linear combination

$$c(e,\, v^*) = \alpha c_{\text{geo}}(e,\, v^*) + \beta_1 c_{\text{disp}}(e,\, v^*) + \beta_2 c_{\text{equi}}(e,\, v^*), \tag{7}$$

applied to the normalized values of the three cost functions, and thus we obtain the overall contraction cost for the edge e and the vertex v^*. A low value of $c(e,\, v^*)$ means that the contraction will yield a good geometric approximation to the original geometry, where the data points are close to their original locations and evenly distributed throughout the triangles of the new mesh. On the contrary, a high value of $c(e,\, v^*)$ means that the contraction will produce a bad approximation of the original surface, or the projected data points are too far from their original locations or there might be triangles with too many or too few data points associated with them. As a consequence, if we iteratively remove the edge of the mesh characterized by the lowest cost, we obtain a new mesh with all the desired properties.

The algorithm is straightforward. We have implemented a dynamic data structure that, for each triangle of the current mesh, stores a valid edge of the triangle which minimizes the value (7). Moving from this data structure, we iteratively contract the edge with the lowest cost until we reach the desired number of nodes. After each contraction the data structure is properly updated. This *ad hoc* data structure orders the edges so that we get the minimum in a constant and reduced time. Actually, the most computationally expensive operation is updating the data structure, since the operation of removal and insertion has a cost which is $O(\log(N))$, where N is the number of elements already stored in the data structure. For an initial mesh with n nodes and a fixed threshold of m nodes, where $m \ll n$, then the simplification algorithm can be outlined as in Algorithm 1.

Throughout the paper, we apply two simplification strategies. We compare the proposed approach that controls both the geometry and data with a more traditional strategy that only utilizes the geometry. In particular, we denote by

- `Data+Geo` the simplification obtained by equally weighting the geometric, the displacement and the distribution cost functions ($\alpha = \beta_1 = \beta_2 = 1/3$ in (7));
- `OnlyGeo` the simplification driven only by the geometric information $c_{\text{geo}}(e, v^*)$ ($\alpha = 1$ and $\beta_1 = \beta_2 = 0$ in (7)).

The choice for the weights in the `Data+Geo` approach is, of coarse, not unique and a more rigorous investigation of this choice is the object of future work. We could make a different choice to give more importance to the geometry or to the data, depending on the manifold or the application we are dealing with.

Algorithm 1 ITERATIVE MESH SIMPLIFICATION ALGORITHM

```
read the original mesh
create the data structure
while the number of nodes of Γ'_h > m do
   find the cheapest valid edge e
   contract the edge e
   update the data structure
end while
```

Let us exemplify the differences between the `Data+Geo` and `OnlyGeo` approaches on the pawn geometry in Figure 8 (left), which originally consists of 2527 nodes. We show the simplified meshes obtained with $m = 1000$, via the

Figure 8: Simplification of a pawn. Original geometry with $n = 2527$ nodes (left), mesh simplified to $m = 1000$ nodes via the `Data+Geo` (center) and the `OnlyGeo` (right) approaches. The color map shows the quantity (5) for each triangle of the mesh.

`Data+Geo` (center) and the `OnlyGeo` (right) approaches. Both the choices preserve the shape of the pawn. However, the results are really different in terms of the data distribution. In particular, the color map shows the quantity (5) for each triangle of the mesh. We see that, by including the data distribution, we are able to generate a mesh that has an even distribution of the number of data points throughout the whole mesh while avoiding triangles with no associated data (compared to the many empty (green) triangles produced by the `OnlyGeo` approach on the right). As shown in Sections 4-5, this property ensures statistical estimates with very good inferential properties.

3 Spatial regression models for non-planar domains

After setting the geometric procedure, in this section, we generalize the SR-NP model proposed in [9] to the case of surfaces topologically equivalent to a sphere. Figure 9 illustrates how we integrate the mesh simplification process with the SR-NP method.

We modify the method for its first application to domains that are topologically equivalent to a sphere. The SR-NP method is a generalization of the penalized least square estimation technique proposed in [27] for planar domains. This method conformally maps the non-planar domain to a planar region. Then, the penalty term employed in the planar case is modified to properly include the original shape of the domain. In this paper, we introduce a new conformal map to deal

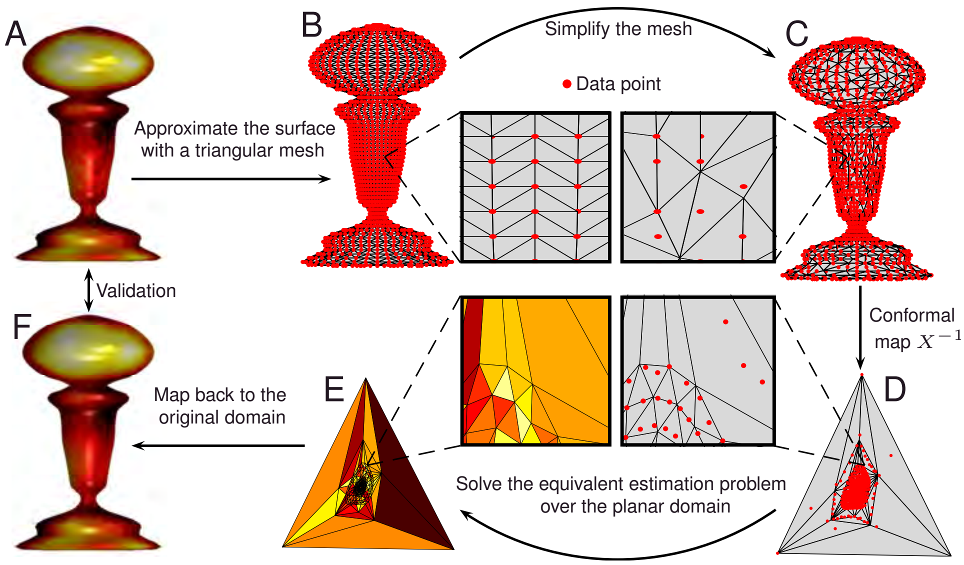

Figure 9: Sketch of the whole procedure: A: simulated data over the pawn geometry; B: original triangular mesh of the pawn in A where each vertex coincides with a data point (see the zoom where the data points are identified via the red dots); C: simplified mesh with 1000 nodes yielded via the method described in Section 2. As the zoom highlights, the data points are not necessarily associated with the mesh nodes; D: planar triangular mesh generated via a conformal flattening map applied to the simplified mesh in C. The data points are projected onto the planar space by evaluating the conformal map at each data point on the geometry shown in C; E: solution of an equivalent estimation problem solved on the planar domain in D; F: the solution to the estimation problem is mapped back to the original manifold and the whole process is finally validated.

with manifolds that are topologically equivalent to a sphere such as the cortical surface. In particular, we move from the conformal map proposed in [1], which has been specifically developed for cortical surfaces. We properly modify this map, realizing that it occasionally fails when applied to configurations characterized by obtuse triangles. Obtuse triangles are commonly created by the automatic meshing procedures for cortical surfaces. The proposed modification affects only the obtuse triangles in the mesh and reduces to the original method proposed in [1] when applied to acute triangles. Furthermore, in the original application of the SR-NP model, the data is assumed to occur only at the nodes of the mesh. On the contrary, after the mesh simplification procedure the data does not necessarily occur at the nodes. Consequently, we properly adapt the SR-NP approach to include this change.

3.1 The SR-NP model

Consider n data locations $\{\mathbf{x}_j = (x_{1j}, x_{2j}, x_{3j}) : j = 1, \dots, n\}$, lying on a non-planar domain Γ that is a uniformly regular surface embedded in $\mathbb{R}^3$. At each location a scalar data value, z, is observed via the value z_j. We assume the following model for the data:

$$z_j = f(\mathbf{x}_j) + \epsilon_j, \tag{8}$$

for $j = 1, \dots, n$, where ϵ_j are independent observational errors with zero mean and constant variance σ^2, while f is a twice continuously differentiable real-valued function defined on the surface domain Γ. Of course, f is the quantity we aim at approximating. In practice, Γ will be approximated by a triangular mesh Γ_h, and successively by the simplified mesh Γ_h', while the original data locations will be approximated by their locations on Γ_h'.

To estimate f, the following penalized sum of squared error functional is minimized:

$$J_{\Gamma,\lambda}(f) = \sum_{j=1}^{n} (z_j - f(\mathbf{x}_j))^2 + \lambda \int_\Gamma (\Delta_\Gamma f(\mathbf{x}))^2 \, d\Gamma, \tag{9}$$

where Δ_Γ is the Laplace-Beltrami operator associated with the surface Γ (see, e.g., [7]). The Laplace-Beltrami operator is a generalization of the standard Laplacian to the case of functions defined on surfaces in Euclidean spaces. Being related to the local curvature of f on Γ, the Laplace-Beltrami operator in the penalty controls the roughness of the solution f. Thus, the functional $J_{\Gamma,\lambda}$ balances the fidelity of the estimate to the data via the sum of the squared errors and the roughness of the solution via the penalty term. The smoothness parameter $\lambda > 0$ adjusts this trade-off. For the planar model setting, methods for choosing the optimal value of the smoothness parameter λ have been discussed in the literature and include the Akaikes Information Criterion (AIC), the Bayesian Information Criterion (BIC) and the Generalized Cross-Validation (GCV) criterion (see [11], [24], [13] and references therein). Here, we resort to a GCV approach.

To solve the estimation problem in (9), we first recast it over a planar domain via a conformal map. For a non-planar domain Γ that is a Riemannian surface embedded in $\mathbb{R}^3$, the Riemann Mapping Theorem ensures that there exists a conformal map from Γ to the unit sphere, the Euclidean plane or the unit disk. Hence, it is possible to define a uniformly regular and continuously differentiable map

$$\begin{aligned} X : \Omega &\to \Gamma \\ \mathbf{u} = (u_1, u_2) &\mapsto \mathbf{x} = (x_1, x_2, x_3), \end{aligned} \tag{10}$$

where Ω is an open, convex and bounded set in $\mathbb{R}^2$ whose boundary $\partial\Omega$ is piecewise C^∞. These types of conformal maps are unique up to dilations, rotations and translations [17]). In particular, the map X is conformal, if

$$\|X_{u_1}(\mathbf{u})\| = \|X_{u_2}(\mathbf{u})\| \text{ and } \langle X_{u_1}(\mathbf{u}), X_{u_2}(\mathbf{u})\rangle = 0,$$

for any $\mathbf{u} \in \Omega$, where $X_{u_1}(\mathbf{u})$ and $X_{u_2}(\mathbf{u})$ are the column vectors of the first order partial derivatives of X with respect to u_1 and u_2, respectively while $\langle \cdot, \cdot \rangle$ denotes the Euclidean scalar product of two vectors with the associated norm $\| \cdot \|$. The (space-dependent) metric tensor is defined as

$$G(\mathbf{u}) := \begin{pmatrix} \|X_{u_1}(\mathbf{u})\|^2 & \langle X_{u_1}(\mathbf{u}), X_{u_2}(\mathbf{u}) \rangle \\ \langle X_{u_2}(\mathbf{u}), X_{u_1}(\mathbf{u}) \rangle & \|X_{u_2}(\mathbf{u})\|^2 \end{pmatrix}.$$

Let $\mathcal{W}(\mathbf{u}) := \sqrt{\det(G(\mathbf{u}))}$, and denote by $G^{-1}(\mathbf{u})$ the inverse of $G(\mathbf{u})$. Then, for $f \circ X \in \mathcal{C}^2(\Omega)$, the Γ-gradient of f is given by

$$\nabla_\Gamma f(\mathbf{x}) = \nabla X(\mathbf{u})\, G^{-1}(\mathbf{u}) \nabla f(X(\mathbf{u})), \tag{11}$$

while the Laplace-Beltrami operator associated with the surface Γ can be written in terms of the map X as

$$\Delta_\Gamma f(\mathbf{x}) = \frac{1}{\mathcal{W}(\mathbf{u})} \mathrm{div}(\mathbf{K} \nabla f(X(\mathbf{u}))), \tag{12}$$

for any $\mathbf{u} \in \Omega$, where $\mathbf{K}(\mathbf{u}) = \mathcal{W}(\mathbf{u})\, G^{-1}(\mathbf{u})$ is a symmetric positive definite matrix and the divergence and gradient operators for planar domains are denoted by div and ∇, respectively (see [7] for more details).

The representation of the Laplace-Beltrami operator in (12) highlights that an estimation problem equivalent to (9) can be properly rewritten over the planar domain Ω via the map X. In more detail, we find a function $f \circ X$ defined on Ω that minimizes

$$J_{\Omega,\lambda}(f \circ X) = \sum_{j=1}^{n} \big(z_j - f(X(\mathbf{u}_j))\big)^2 + \lambda \int_\Omega \frac{1}{\mathcal{W}(\mathbf{u})} \Big(\mathrm{div}(\mathbf{K} \nabla f(X(\mathbf{u}))) \Big)^2 d\Omega, \tag{13}$$

where Ω is the domain in $\mathbb{R}^2$ obtained via the flattening of the cortical surface. The existence and the uniqueness of a solution to the estimation problem in (13), is established in the functional space

$$H^2_{n0,\mathbf{K}}(\Omega) = \big\{ g \in H^2(\Omega) \ : \ \mathbf{K} \nabla g \cdot n = 0 \text{ on } \partial\Omega \big\}$$

which consists of functions in $H^2(\Omega)$ whose co-normal derivative is identically equal to zero on the boundary of Ω, $\partial\Omega$. We note that $H^2_{n0,\mathbf{K}}(\Omega) \subset H^2(\Omega)$ is a modification of the standard Sobolev space $H^2(\Omega)$ ([21]).

Let $\boldsymbol{z} = (z_1, \ldots, z_n)^t$ be the vector collecting the observed data values in (8). For any function g defined on Γ, such that $g \circ X$ is defined on Ω, we denote the column vector of evaluations of the function g at the n data locations $\mathbf{x}_j$ by

$$\mathbf{g}_n = \Big(g(\mathbf{x}_1), \ldots, g(\mathbf{x}_n) \Big)^t = \Big(g(X(\mathbf{u}_1)), \ldots, g(X(\mathbf{u}_n)) \Big)^t, \tag{14}$$

with $X(\mathbf{u}_j) = \mathbf{x}_j$. To ease the notation, in the following we omit the dependence on $\mathbf{u}$. In [9] it is shown that the estimator $\hat{f} \circ X$ that minimizes (13) over $H^2_{n0,\mathbf{K}}(\Omega)$ satisfies the relation

$$\boldsymbol{\mu}_n^t \boldsymbol{z} = \boldsymbol{\mu}_n^t \hat{\mathbf{f}}_n + \lambda \int_\Omega \frac{1}{\mathcal{W}} \Big(\operatorname{div} \left(\mathbf{K} \nabla (\mu \circ X) \right) \Big) \Big(\operatorname{div} \Big(\mathbf{K} \nabla (\hat{f} \circ X) \Big) \Big) d\Omega, \quad (15)$$

for any μ defined on Γ such that $\mu \circ X \in H^2_{n0,\mathbf{K}}(\Omega)$, with $\boldsymbol{\mu}_n$ and $\hat{\mathbf{f}}_n$ defined according to (14). Moreover, for a fixed X, the estimator $\hat{f} \circ X$ is unique.

3.2 The conformal map

The goal of this section is to properly define the map X. The map employed in [9] to flatten arteries with an aneurysm can not be directly applied for the flattening of the cortical surface. For this purpose, we propose a modification of the approach developed in [1]. This method assumes the cortical surface can be approximated by a topological sphere and then uses a result from complex analysis that identifies a topological sphere minus a point with the complex plane. When we consider a triangular mesh of the topological sphere, this corresponds to mapping the mesh minus a fixed triangle into the image of the fixed triangle in the complex plane. However, this image in the complex plane is not very good for visualization purposes. For this reason in [1] the flattened triangulation in the complex plane is subsequently mapped to the unit sphere via the inverse stereographic projection. For the SR-NP method, we only need to use the first part of this transformation to map the cortical surface to the complex plane. We note that this map, as it is proposed, does not work for any triangulation. In particular, it can produce a triangulation in the plane with overlapping triangles. The map relies on a cotangent formula that breaks down for certain configurations involving obtuse triangles. We suggest a modification of this map that only affects the obtuse angles and can be used to flatten any triangulations without generating overlapping elements.

First, we introduce the map as proposed in [1]. For a smooth two-dimensional closed manifold Γ with genus zero embedded in $\mathbb{R}^3$, the conformal coordinates u_1 and u_2 of the planar domain Ω in (10) can be defined by a map which depends on a single point p on Γ. In particular, we can map the surface Γ without the point p to the complex plane $\mathbb{C}$, i.e., the inverse map of X in (10) is now given by $X^{-1} : \Gamma \backslash \{p\} \to \mathbb{C}$. Thus, after assuming that the cortical surface is topologically equivalent to a sphere, we can identify the unit sphere without the north pole with the complex plane via stereographic projection. In particular, we have decided to identify the complex plane with $\mathbb{R}^2$ to get the planar domain suited for our estimation problem. To this end, we start by finding the inverse map $X^{-1} = u_1 + iu_2$ by solving the following partial differential equation

$$\Delta_\Gamma X^{-1} = \left(\frac{\partial}{\partial \nu_1} - i \frac{\partial}{\partial \nu_2} \right) \delta_p, \quad (16)$$

where δ_p is the Dirac delta function at the point p, i is the imaginary unit, and ν_1 and ν_2 are local coordinates defined in a neighborhood of p. The Laplace-Beltrami problem (16) on $\Gamma\backslash\{p\}$ is completed with full homogeneous Neumann boundary conditions.

3.2.1 Approximation of the conformal flattening map

In practice, we approximate (16) on a discretization of Γ, i.e., on the simplified triangulated surface Γ'_h yielded by the mesh simplification procedure described in Section 2. For this purpose, we first introduce a suitable functional setting. Let $V(\Gamma'_h) = H^1(\Gamma'_h)$ define the Sobolev space of the functions defined on Γ'_h which are in $L^2(\Gamma'_h)$ together with all their first order partial derivatives [8]). Let $V_h(\Gamma'_h) \subset V(\Gamma'_h)$ be the finite dimensional discrete space of the piecewise linear functions defined on Γ'_h. We denote by $\{\psi_j\}$ a Lagrangian basis for $V_h(\Gamma'_h)$, such that $\psi_j(v_l) = \delta_{jl}$ for any vertex v_l of Γ'_h, where δ_{jl} is the Kronecker delta symbol.

First, let us approximate the right-hand side of (16). Let g be a generic smooth function in a neighborhood of p. Then, we have

$$\iint_{\Gamma'_h} g\left(\frac{\partial}{\partial\nu_1} - i\frac{\partial}{\partial\nu_2}\right)\delta_p\, d\Gamma'_h = -\left(\frac{\partial g}{\partial\nu_1} - i\frac{\partial g}{\partial\nu_2}\right)\bigg|_p. \tag{17}$$

In particular, if $g \in V_h(\Gamma'_h)$, we can compute the quantities in (17) by the values of g at the vertices of the triangle $\triangle_{ABC}$ (i.e., the triangle with vertices $A,\ B,\ C$) that contains the point p (see [1]). Now, we choose the ν_1- and ν_2-axes so that A and B lie along the ν_1-axis and the positive ν_2-axis points towards C. Then, we can easily compute

$$\frac{\partial g}{\partial\nu_1} = \frac{g(B) - g(A)}{\|B - A\|} \qquad \text{and} \qquad \frac{\partial g}{\partial\nu_2} = \frac{g(C) - g(C_\perp)}{\|C - C_\perp\|},$$

where $C_\perp$ is the orthogonal projection of C on the edge AB. By exploiting the linearity of g together with the orthogonality relation $\langle C - C_\perp,\ B - A\rangle = 0$, from (17) we obtain

$$\begin{aligned}\iint_{\Gamma'_h} & g\left(\frac{\partial}{\partial\nu_1} - i\frac{\partial}{\partial\nu_2}\right)\delta_p\, d\Gamma'_h \\ &= \frac{g(A) - g(B)}{\|B - A\|} + i\frac{g(C) - (g(A) + \Theta(g(B) - g(A)))}{\|C - C_\perp\|},\end{aligned} \tag{18}$$

where

$$\Theta = \frac{\langle C - A, B - A\rangle}{\|B - A\|^2}.$$

Thus, we have a closed form for the right-hand side of (16) in the discrete setting.

During the mesh simplification process in Section 2, we fix the triangle $\triangle_{ABC}$ containing the point p, i.e., we add the requirement that the simplified mesh must preserve the triangle $\triangle_{ABC}$. In particular, this triangle will coincide with the element removed from the triangulated surface before the flattening, i.e., the image of $\triangle_{ABC}$ via the map X^{-1} will identify the domain $\Omega \subset \mathbb{R}^2$. Since during the flattening phase the interior of the triangle $\triangle_{ABC}$ is removed, the only data associated with it occurs at the vertices A, B, and C so that no data is lost. Moreover, fixing the same triangle in the original mesh for each simplification standardizes the flattening procedure for more appropriate comparisons. Figure 10 shows the fixed triangle for the pawn geometry in Figure 8, specifically, for the original mesh (left), for the `Data+Geo` simplification (center) and for the `OnlyGeo` simplification (right). For the cortical surface mesh, we fix the triangle whose barycenter is closest to the center of mass of the original vertices of the mesh.

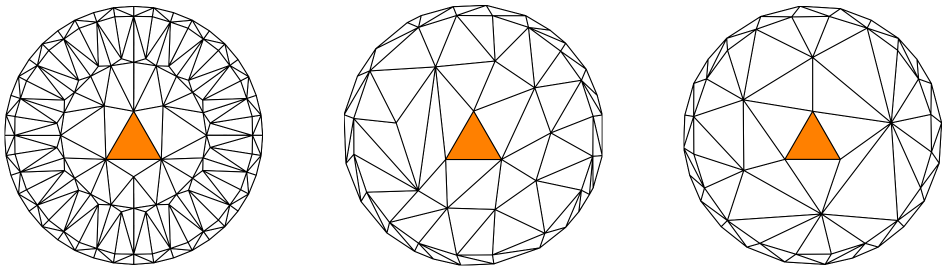

Figure 10: The fixed triangle selected for the pawn simplifications is located on the bottom of the pawn and does not change during the simplification process: the original triangulation Γ'_h (left), the 1000 node mesh yielded by the `Data+Geo` (center), and the `OnlyGeo` (right) simplifications.

Now, let us come back to the approximation of problem (16). It is well-known that X^{-1} is the minimizer of the functional

$$\frac{1}{2} \iint_{\Gamma'_h} \left(\|\nabla_{\Gamma'_h} X^{-1}\|^2 + 2X^{-1} \left(\frac{\partial}{\partial \nu_1} - i \frac{\partial}{\partial \nu_2} \right) \delta_p \right) d\Gamma'_h, \tag{19}$$

where $\nabla_{\Gamma'_h}$ is defined analogously to ∇_Γ in (11). Using this definition and by exploiting (17), it can be shown that X^{-1} satisfies (16) if and only if, for all smooth test functions g, we have

$$\iint_{\Gamma'_h} \nabla_{\Gamma'_h} X^{-1} \cdot \nabla_{\Gamma'_h} g \, d\Gamma'_h = \left. \left(\frac{\partial g}{\partial \nu_1} - i \frac{\partial g}{\partial \nu_2} \right) \right|_p . \tag{20}$$

Thus, an approximation to the conformal map is found by finding $X^{-1} \in V_h(\Gamma'_h)$ such that (20) holds for any $g \in V_h(\Gamma'_h)$. In particular, since (20) is linear in g, it is enough to guarantee the condition (20) for any basis function $\psi_k \in V_h(\Gamma'_h)$ after expanding X^{-1} in terms of the basis $\{\psi_j\}$. Hence, we are led to solve a linear system that finds a complex number $X^{-1}(v_j) = u_{1j} + i\, u_{2j}$ for each vertex v_j of the simplified mesh Γ'_h, with $j = 1, \ldots, m$, and such that

$$\sum_{j=1}^{m} X^{-1}(v_j) \iint_{\Gamma'_h} \nabla_{\Gamma'_h} \psi_j \cdot \nabla_{\Gamma'_h} \psi_k \, d\Gamma'_h = \left(\frac{\partial \psi_k}{\partial \nu_1} - i \frac{\partial \psi_k}{\partial \nu_2} \right) \Big|_p , \tag{21}$$

for $k = 1, \ldots, m$. Notice that (u_{1j}, u_{2j}) identifies the location of the vertex v_j of Γ'_h in the corresponding flattened mesh denoted in the following by Ω'_h.

Let D denote the stiffness matrix in (21). The components of the stiffness matrix,

$$D_{jk} = \iint_{\Gamma'_h} \nabla_{\Gamma'_h} \psi_j \cdot \nabla_{\Gamma'_h} \psi_k \, d\Gamma'_h,$$

can be computed by resorting to a well-known cotangent formula. This formula is based on the conformal invariance of the energy functional

$$E(X^{-1}) = \frac{1}{2} \iint_{\Gamma'_h} \|\nabla_{\Gamma'_h} X^{-1}\|^2 \, d\Gamma'_h, \tag{22}$$

with respect to conformal changes of domain metric (see [23] for more details). For a triangle $T_1 \in \Gamma'_h$, the energy reduces to

$$E(X^{-1})\big|_{T_1} = \frac{1}{4} \sum_{j=1}^{3} \cot \theta_j \|\tilde{e}_j\|^2_{\Omega'_h}, \tag{23}$$

where θ_j is an angle of T_1 while $\|\tilde{e}_j\|_{\Omega'_h}$ denotes the length of the edge opposite to θ_j in the corresponding triangle $\widetilde{T}_1 \in \Omega'_h$ (see Figure 11). As a consequence, we can define the energy associated with the map X^{-1} as the sum of the energy of each triangle in the mesh, i.e.,

$$E(X^{-1}) = \sum_{T \in \Gamma'_h} E(X^{-1})\big|_T = \frac{1}{4} \sum_{\tilde{e}_j \in \Omega'_h} (\cot \theta_j + \cot \theta_k) \, \|\tilde{e}_j\|^2_{\Omega'_h} \tag{24}$$

where $\tilde{e}_j$ is the generic edge of Ω'_h with end points $\tilde{v}_j$ and $\tilde{v}_k$, while θ_j and θ_k denote the angles opposite e_j in the two adjacent elements (see Figure 11). The energy thus coincides with a weighted sum of edge lengths. From (24), a generic component of the stiffness matrix D can be computed as

$$D_{jk} = -\frac{1}{2} \left(\cot \theta_j + \cot \theta_k \right), \tag{25}$$

if v_j and v_k are connected by an edge and zero otherwise ([16]). Moreover, the diagonal entries of D are defined by $D_{jj} = -\sum_{k \neq j} D_{jk}$, since we have $\sum_j D_{jk} = 0$. Then, to find the planar coordinates u_{1j} and u_{2j} for each vertex of the mesh Γ'_h, we define vectors $a, b \in \mathbb{R}^m$ with components $a_k = \left(\frac{\partial \psi_k}{\partial \nu_1}(p) \right)$ and $b_k = \left(\frac{\partial \psi_k}{\partial \nu_2}(p) \right)$,

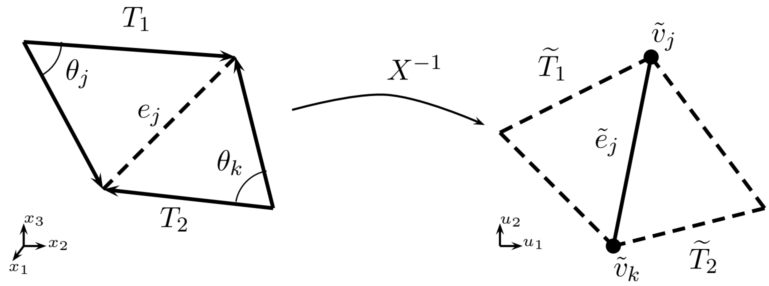

Figure 11: Quantities involved in the energy definition.

respectively, for $k = 1, \ldots, m$. Via (18), we obtain

$$a - ib := \begin{cases} 0 & \text{if } v_k \notin \{A,\ B,\ C\}, \\ \frac{-1}{\|B-A\|} + i\frac{1-\Theta}{\|C-C_\perp\|} & \text{if } v_k = A, \\ \frac{1}{\|B-A\|} + i\frac{\Theta}{\|C-C_\perp\|} & \text{if } v_k = B, \\ i\frac{-1}{\|C-C_\perp\|} & \text{if } v_k = C, \end{cases}$$

so that the conformal coordinates $u_1 = (u_{11},\ u_{12}, \ldots,\ u_{1m})^t$ and $u_2 = (u_{21},\ u_{22}, \ldots,\ u_{2m})^t$ defining X^{-1} are the solutions to the linear systems

$$Du_1 = a \quad \text{and} \quad Du_2 = -b, \tag{26}$$

respectively. We remark that the choice made for the boundary conditions completing problem (16) yields a singular stiffness matrix D. However, since both a and b belong to the orthogonal complement, $\ker(D)^\perp$, of $\ker(D)$, both the linear systems in (26) are solvable. In particular, since D restricted to $\ker(D)^\perp$ is symmetric positive definite, we solve these systems via the conjugate gradient method.

A critical issue for this formulation is that, for a mesh with obtuse angles like the ones involved cortical surface applications, the cotangent weights in (25) may be negative ([2]). As a consequence, the orientation of the edges around a vertex can change, resulting in overlapping triangles in the planar domain. Moving from (24), we can heuristically consider the energy $E(X^{-1})$ as the amount of tension each triangle places on the edges or, likewise, the vertices. With this interpretation, a negative cotangent weight works as a repelling force that pushes the vertex away from the triangle instead of pulling it towards itself. Figure 12 (top) illustrates the problem. In particular, the vectors $\mathbf{h}_1,\ \mathbf{h}_2$ and $\mathbf{h}_3$ in Figure 12 (top) show the directions that the triangles $T_1 = \triangle_{v_1 v_2 v_3}$, $T_2 = \triangle_{v_1 v_3 v_4}$ and $T_3 = \triangle_{v_1 v_4 v_5}$ act on v_1, respectively. The cotangent weight formula applied to the triangle T_1 pushes the vertex v_1 towards the triangle T_3 causing T_1 to overlap T_2 and T_3 in the planar domain.

To alleviate the problems due to the negative cotangent weights, we consider the absolute value of the cotangents in (25) when computing the off-diagonal ele-

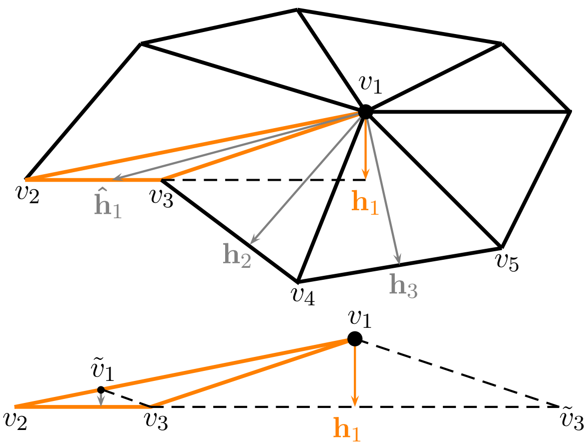

Figure 12: In this configuration the obtuse triangle highlighted in orange has a negative weight that flips the order of the edges when using (25) (top). Using the modification in (27), the stiffness matrix uses the acute triangle $\triangle_{v_1 v_2 \tilde{v}_3}$ (or equivalently $\triangle_{\tilde{v}_1 v_2 v_3}$) instead of $\triangle_{v_1 v_2 v_3}$ (bottom).

ments D_{jk}, i.e., now

$$D_{jk} = -\frac{1}{2}\left(|\cot\theta_j| + |\cot\theta_k|\right), \tag{27}$$

while the diagonal entries are computed so that the property $\sum_j D_{jk} = 0$ holds. Notice that the new formula (27) only affects the weights related to obtuse angles while reducing to formula (25) in the presence of acute angles. In such a way, each triangle is forced to exert an attractive force on its vertices. For the configuration in Figure 12 (top), the modification in (27) produces for the triangle T_1 the new direction $\hat{\mathbf{h}}_1$ instead of $\mathbf{h}_1$ and pulls the vertex v_1 in the proper direction. Formula (27) effectively substitutes the obtuse angles with their supplementary angles. Hence, the formula in (27) uses the triangle $T_4 = \triangle_{v_1 v_2 \tilde{v}_3}$ instead of using T_1 (see Figure 12 (bottom)). Note that T_4 and T_1 both force v_1 in the direction of $\mathbf{h}_1$. However, the components of the corresponding stiffness matrix depend only on the angles involved and thus similar triangles have the same elemental stiffness. Now, due to that fact that the edge with vertices v_2 and v_3 is a constraint for this configuration, $T_5 = \triangle_{\tilde{v}_1 v_2 v_3}$, the triangle similar to T_4 that has $v_2 v_3$ as an edge, will be used instead of T_4. Hence, the contribution for v_1 from T_1 calculated via (27) pulls the triangle T_1 in the direction of $\hat{\mathbf{h}}_1$ as shown in Figure 12 (top). This preserves the orientation of the triangles around v_1 and prevents T_1 from overlapping neighboring triangles in the planar domain. The idea of using the absolute value of the cotangent weights is also implemented in [20]. Figure 13 shows the benefits of employing formula (27) instead of the standard one (25).

As a last step of the flattening procedure the map X^{-1} is evaluated at the data points $\mathbf{w}_j = X^{-1}(\mathbf{x}_j)$ with $j = 1, \ldots, n$, that have been projected from Γ_h to Γ'_h

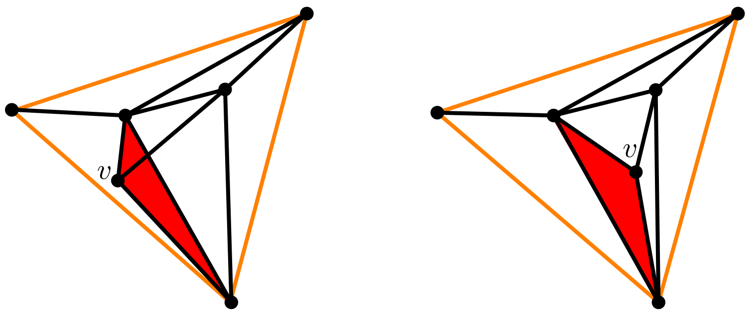

Figure 13: Flattening via formula (25) (left) and via formula (27) (right)

via the simplification process to obtain their planar locations in Ω'_h.

3.3 A finite element approximation for the estimation problem

To get a finite element approximation of the estimation problem (15), we properly reformulate this problem by introducing an auxiliary function. Essentially, we aim at reducing the regularity assumption of $\hat{f} \circ X$. The introduction of the unknown auxiliary function leads us to rewrite (15) as a system of coupled equations: find $(\hat{f} \circ X, \gamma \circ X) \in (H^1_{n0,\mathbf{K}}(\Omega) \cap C^0(\bar{\Omega})) \times H^1(\Omega)$ such that

$$\begin{aligned} \boldsymbol{\mu}_n^t \hat{\mathbf{f}}_n - \lambda \int_\Omega \mathbf{K}\nabla(\mu \circ X) \cdot \nabla(\gamma \circ X) d\Omega &= \boldsymbol{\mu}_n^t \boldsymbol{z} \\ \int_\Omega (\xi \circ X)(\gamma \circ X)\mathcal{W} d\Omega + \int_\Omega \nabla(\xi \circ X)\mathbf{K}\nabla(\hat{f} \circ X) d\Omega &= 0, \end{aligned} \tag{28}$$

for any $(\mu \circ X, \xi \circ X) \in (H^1_{n0,\mathbf{K}}(\Omega) \cap C^0(\bar{\Omega})) \times H^1(\Omega)$, where $H^1_{n0,\mathbf{K}}(\Omega)$ consists of functions in $H^1(\Omega)$ whose co-normal derivative is identically equal to zero on $\partial\Omega$. The regularity of the problem guarantees the solution, $\hat{f} \circ X$, still belongs to $H^2_{n0,\mathbf{K}}(\Omega)$.

Now, analogously to Section 3.2.1, we introduce the linear finite element space $V_h(\Omega'_h)$ associated with the flattening Ω'_h of the simplified surface Γ'_h. We recall that the dimension of the space $V_h(\Omega'_h)$ coincides with the number of nodes of the flattened mesh Ω'_h. Then, we can state the discrete counterpart of the estimation problem (28), which leads us to find $(\hat{f} \circ X, \gamma \circ X) \in V_h(\Omega'_h) \times V_h(\Omega'_h)$ such that (28) holds for any $(\mu \circ X, \xi \circ X) \in V_h(\Omega'_h) \times V_h(\Omega'_h)$, where the integrals are now computed over Ω'_h. To provide an algebraic counterpart of the discrete formulation, we introduce the mass and stiffness finite element matrices given by

$$\boldsymbol{R}_0 = \int_{\Omega'_h} \tilde{\boldsymbol{\psi}}\tilde{\boldsymbol{\psi}}^t \, \mathcal{W} d\Omega'_h \in \mathbb{R}^{m \times m} \qquad \text{and} \qquad \boldsymbol{R}_1 = \int_{\Omega'_h} \nabla\tilde{\boldsymbol{\psi}}^t \boldsymbol{K} \nabla\tilde{\boldsymbol{\psi}} \, d\Omega'_h \in \mathbb{R}^{m \times m},$$

respectively, where $\tilde{\boldsymbol{\psi}} = (\tilde{\psi}_1, \ldots, \tilde{\psi}_m)^t$ is the column vector of the m finite element basis functions for $V_h(\Omega'_h)$. That is, for any $g \in V_h(\Omega'_h)$

$$g(\cdot) = \sum_{j=1}^{m} g(\tilde{v}_j)\tilde{\psi}_j(\cdot) = \mathbf{g}^t\tilde{\boldsymbol{\psi}}(\cdot), \quad \text{where} \quad \mathbf{g} = (g(\tilde{v}_1), \ldots, g(\tilde{v}_m))^t \in \mathbb{R}^m \quad (29)$$

is the column vector of evaluations of g at the m nodes $\tilde{v}_j$ of the mesh Ω'_h. Define

$$\tilde{\Psi} = \begin{bmatrix} \tilde{\boldsymbol{\psi}}^t(\mathbf{w}_1) \\ \vdots \\ \tilde{\boldsymbol{\psi}}^t(\mathbf{w}_n) \end{bmatrix} \in \mathbb{R}^{n\times m},$$

to be the matrix of the m basis functions evaluated at the n data locations. Extending the arguments detailed in [9] to the case where the data does not necessarily occur at the vertices of the mesh, the discrete counterpart of the estimation problem (28) reduces to finding the pair of coefficient vectors $(\hat{\boldsymbol{f}}, \gamma) \in \mathbb{R}^m \times \mathbb{R}^m$ such that, for any $(\boldsymbol{\mu}, \boldsymbol{\xi}) \in \mathbb{R}^m \times \mathbb{R}^m$, we have

$$\begin{cases} \boldsymbol{\mu}^t\tilde{\Psi}^t\tilde{\Psi}\hat{\boldsymbol{f}} - \lambda\boldsymbol{\mu}^t\boldsymbol{R}_1\gamma = \boldsymbol{\mu}^t\tilde{\Psi}^t\boldsymbol{z} \\ \boldsymbol{\xi}^t\boldsymbol{R}_0\gamma + \boldsymbol{\xi}^t\boldsymbol{R}_1\hat{\boldsymbol{f}} = \mathbf{0}, \end{cases}$$

with $\tilde{\Psi}\hat{\mathbf{f}} = \hat{\mathbf{f}}_n$ and $\tilde{\Psi}\boldsymbol{\mu} = \boldsymbol{\mu}_n$ where $\mathbf{0} \in \mathbb{R}^m$ denotes the null vector, $\hat{\boldsymbol{f}}$, $\boldsymbol{\gamma}$, $\boldsymbol{\mu}$, $\boldsymbol{\xi}$ are defined according to (29) and $\boldsymbol{z}$ is as defined in (28). Then, the estimator $\hat{f} \circ X \in V_h(\Omega'_h)$ that solves the discrete counterpart of the estimation problem is given by $\hat{f} \circ X = \hat{\boldsymbol{f}}^t\tilde{\boldsymbol{\psi}}$, where $\hat{\boldsymbol{f}}$ satisfies

$$\begin{bmatrix} -\tilde{\Psi}^t\tilde{\Psi} & \lambda\boldsymbol{R}_1 \\ \lambda\boldsymbol{R}_1 & \lambda\boldsymbol{R}_0 \end{bmatrix} \begin{bmatrix} \hat{\boldsymbol{f}} \\ \gamma \end{bmatrix} = \begin{bmatrix} -\tilde{\Psi}^t\boldsymbol{z} \\ \mathbf{0} \end{bmatrix}, \tag{30}$$

and $\boldsymbol{\gamma}$ is the component vector associated with the auxiliary function γ employed in (28). Moreover, for a given X, $\hat{f} \circ X$ is uniquely determined. From (30) it follows that

$$\hat{\boldsymbol{f}} = \left(\tilde{\Psi}^t\tilde{\Psi} + \lambda\boldsymbol{R}_1\boldsymbol{R}_0^{-1}\boldsymbol{R}_1\right)^{-1}\tilde{\Psi}^t\boldsymbol{z}. \tag{31}$$

Notice that the estimate $\hat{\boldsymbol{f}}$ is linear in the observed data and has a typical penalized regression form (see, e.g., [24]). Thus, classical inferential tools can be applied, such as approximate confidence bands for f and approximate prediction intervals for new data locations. Moreover, (31) yields a closed form for a Generalized-Cross-Validation (GCV) criterion that can be used to select the smoothing parameter λ. We refer to [9] for more details.

Remark 3.1 *In many neuroimaging applications it could be extremely interesting to include covariate information in the model. For instance, when studying hemodynamic signals over the cortical surface in response to a stimulus, it would be interesting to take into account the thickness of the cortical surface at each location as a covariate since the thickness of the cortical surface may indeed influence* *the size of the hemodynamic signal. The covariate inclusion leads, in general, to a more in-depth analysis by preventing the compounding of the results with other information that is varying along with the quantity of interest. Through a semi-parametric framework, the model presented in this paper can also be extended to include space-varying covariate information following [9].*

4 Simulation studies

In this section, we show the good performance of the proposed technique on the pawn geometry introduced in Figure 8. In particular, our goal is to verify that the mesh simplification procedure described in Section 2 produces a mesh that can lead to good statistical estimates, comparable with the ones on the original mesh. For this purpose, we also compare the proposed approach with the Iterative Heat Kernel (IHK) smoothing developed specifically for neuroimaging applications in [4]. The mesh simplification procedure as well as the simulations presented in this paper were run with Matlab 7.12.0 on 2 GHz Intel Core i7 processor in a MacBook Pro with a 256GB Solid State hard drive.

The IHK method works directly on the mesh without any flattening. To do this, a Laplace-Beltrami eigenvalue problem is solved directly on the surface Γ, i.e., ordered eigenvalues $0 = \lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \cdots$ and the corresponding eigenfunctions $\phi_0, \phi_1, \phi_2, \ldots$ are found by solving the eigenvalue problem $-\Delta_\Gamma \phi_j = \lambda_j \phi_j$ on Γ. Thus, a heat kernel with bandwidth B is constructed from the eigenvalue-eigenfunction pairs $\{(\lambda_j, \phi_j)\}$ as

$$K_B(p,q) = \sum_{j=0}^{\infty} e^{-\lambda_j B} \phi_j(p)\phi_j(q),$$

where p and q are two generic points on Γ. The heat kernel smoothing of the quantity of interest z_j in (8), is thus given by $K_B * z_j = \sum_{l=0}^{\infty} e^{-\lambda_l B} \beta_l(\mathbf{x}_j)\phi_l(\mathbf{x}_j)$, where $\beta_l(\mathbf{x}_j) = \langle z_j, \phi_l(\mathbf{x}_j)\rangle$. In practice, only k eigenvalue-eigenfunction pairs are selected via an iterative residual fitting algorithm. For a fixed bandwidth, the level of smoothing is determined by an optimal number of eigenfunctions selected via the F-test criterion outlined in [4].

For the simulations based on the IHK method, we use the full mesh of the pawn constituted by 2527 vertices. Note we cannot use the IHK method on a simplified mesh since the method is currently devised to only work on data observed at the vertices of the mesh. The bandwidth B has been heuristically chosen by selecting the one with the best performance after some test runs. In particular, we set $B = 10^{-2.5}$. Then, the optimal number of eigenfunctions is selected via the F-test criterion for each simulation replicate.

For the SR-NP method, we use the two mesh simplification strategies introduced in Section 2.3 with several levels of simplification and show how the resulting estimates compare to the IHK results on the full mesh. The levels of simplification we use are provided by selecting $m = 1000, 1200, 1400, 1600, 1800,$

2000 vertices. The results obtained for the 1000 node simplified meshes, using the `Data+Geo` and the `OnlyGeo` approaches are shown in Figure 8 (center and right, respectively). For the sake of completeness, we also compare the results to the SR-NP method using the original mesh seen in Figure 8 (left). For each simulation replicate over each simplified mesh, the smoothing parameter λ for the SR-NP method is chosen by GCV.

In more detail, first we generate simulated data on the original mesh of the pawn. To do this, we consider fifty test functions of the form

$$f(x_1, x_2, x_3) = a_1 \sin(2\pi x_1) + a_2 \sin(2\pi x_2) + a_3 \sin(2\pi x_3) + 1, \tag{32}$$

with coefficients a_j, for $j = 1, 2, 3$ randomly generated from independent normal distributions with mean one and standard deviation one, where the data locations $\mathbf{x}_j$, for $j = 1, \ldots, n$, coincide with the nodes of the original three-dimensional mesh. Noisy data values z_j are obtained by adding independent normally distributed errors with mean zero and a standard deviation 0.5 to f at each of the data locations, i.e., we have $z_j = f(x_{1j}, x_{2j}, x_{3j}) + \epsilon_j$, for $j = 1, \ldots, n$, with $\epsilon_j \sim N(0, 0.5)$. Figure 14 shows a simulation example, specifically, a sample test function generated by (32) in (a), the corresponding noisy observations z_i in (b), the IHK estimate computed on the original mesh in (c), the SR-NP estimate obtained on the 1000 node meshes yielded by the `Data+Geo` and the `OnlyGeo` simplifications in (d) and (e), respectively. We see that, despite using less than half the nodes, the SR-NP method is able to detect more of the function variation with respect to the IHK approach. This is most evident on the base of the pawn. Furthermore, via the even data distribution shown in Figure 8 (center), the `Data+Geo` simplification maintains an even level of smoothing over the entire pawn. On the contrary, the estimates computed on the simplified mesh obtained via the `OnlyGeo` approach over-smoothes on the top left-hand side of the pawn, as seen in Figure 14 (e).

The superior performance of the SR-NP method combined with the `Data+Geo` simplification is more evident in Table 1 and Figure 15, where more quantitative information can be inferred. In particular, for each mesh simplification and simulation replicate, we compute the Mean Square Error (MSE) of the estimate, i.e., the mean square distance between the true function f and its estimate $\hat{f}$. A lower MSE means a more efficient estimate, characterized by lower bias (i.e., lower systematic errors) and lower variance. In Table 1, we provide the median MSEs computed over the fifty simulation replicates and, within parentheses, the corresponding Inter Quartile Range (IQR) which quantifies the variability of the MSEs over the fifty replicates. This information can be derived also from the box-plots in Figure 15 which illustrate graphically the comparison among the different methods. Table 1 and Figure 15 highlight that, as expected, the SR-NP method with the simplification based on the overall cost function, `Data+Geo`, produces better results than the SR-NP method with the simplification driven only by geometric information, `OnlyGeo`. In particular, for all levels of mesh simplification except for the least simplified mesh (m = 2000), both the median MSEs and the corresponding IQRs

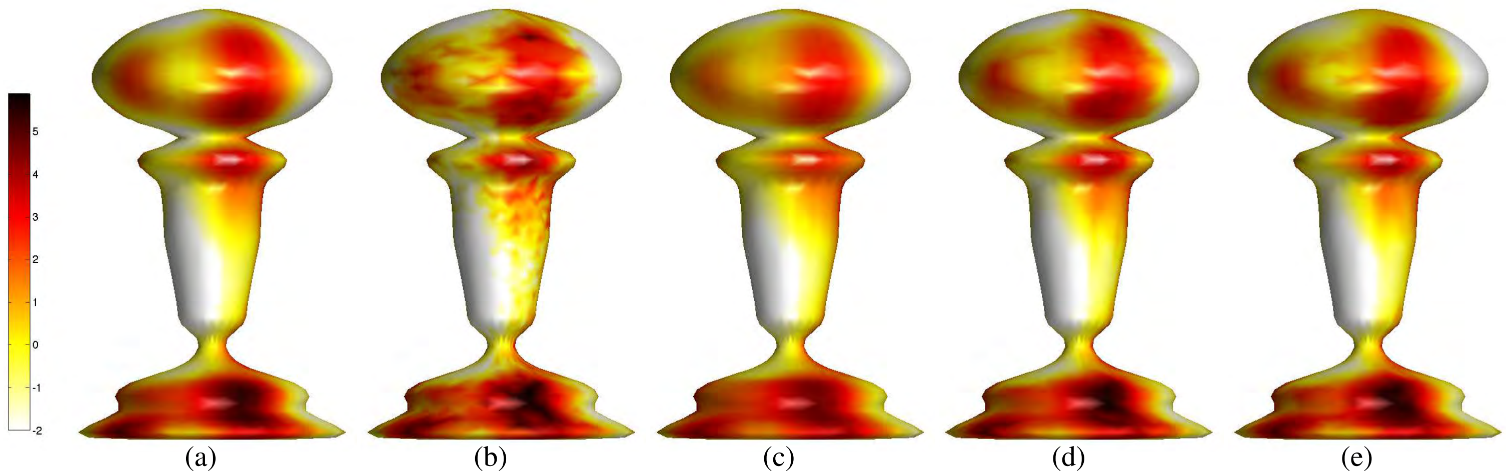

Figure 14: Example of a simulation on the pawn geometry: (a) example of the test function (32), (b) the noisy data z_i as in (8), (c) the IHK estimate computed on the original mesh, (d) the SR-NP estimate on the 1000 node mesh yielded by the `Data+Geo` simplification, (e) the SR-NP estimate on 1000 node mesh provided by the `OnlyGeo` simplification. The color maps are obtained by linear interpolation of the data at $\mathbf{x}_j$ for j=1,...,2527.

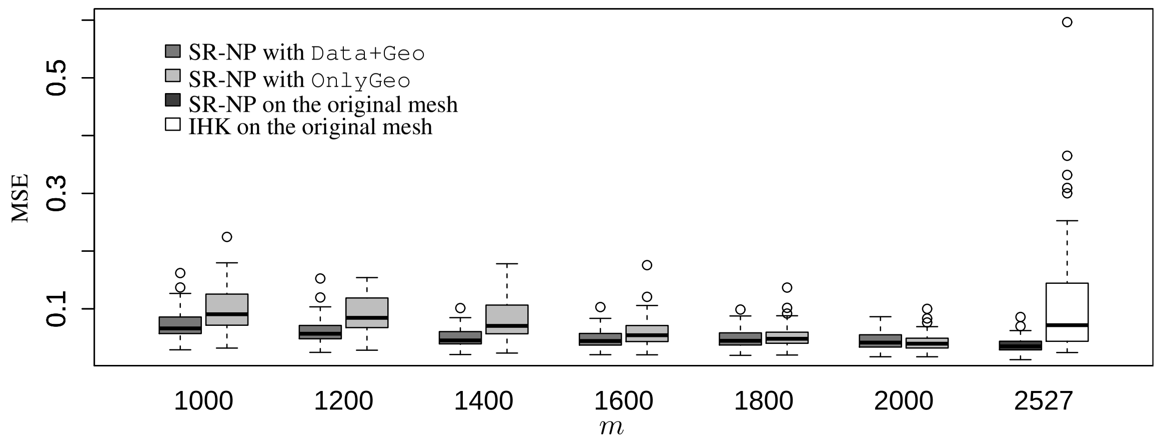

Figure 15: Box-plots of the Mean Square Errors (MSEs) of the estimates over the 50 simulation repetitions over the pawn.

Table 1: Median (IQR) of the Mean Square Errors of the estimates obtained in the 50 simulation repetitions over the pawn and the p-values of pairwise Wilcoxon tests verifying that the MSEs of the SR-NP estimates are significantly lower than those for IHK estimates.

Method	m	Simplification	MSE	p-value
SR-NP	1000	`Data+Geo`	0.0662 (0.0281)	0.0447
		`OnlyGeo`	0.0903 (0.0494)	0.7316
SR-NP	1200	`Data+Geo`	0.0569 (0.0226)	0.0015
		`OnlyGeo`	0.0839 (0.0489)	0.4386
SR-NP	1400	`Data+Geo`	0.0453 (0.0206)	2.207×10^{-8}
		`OnlyGeo`	0.0698 (0.0483)	0.0963
SR-NP	1600	`Data+Geo`	0.0443 (0.0198)	3.100×10^{-9}
		`OnlyGeo`	0.0528 (0.0285)	1.801×10^{-5}
SR-NP	1800	`Data+Geo`	0.0447 (0.0207)	2.762×10^{-9}
		`OnlyGeo`	0.0467 (0.0190)	1.979×10^{-8}
SR-NP	2000	`Data+Geo`	0.0416 (0.0208)	4.139×10^{-10}
		`OnlyGeo`	0.0378 (0.0163)	3.895×10^{-10}
SR-NP	2527		0.0351 (0.0148)	3.895×10^{-10}
IHK	2527		0.0717 (0.0978)	

associated with the `Data+Geo` approach are lower than the `OnlyGeo` ones, corresponding to more accurate and more precise estimates. In the last row of the table, we compare the SR-NP approach with the IHK method on the original mesh. The SR-NP method combined with the simplification strategy driven by both data and geometry controls produces better results in terms of estimates with lower error (the MSEs have a lower median) and more robustness (the MSEs have a smaller IQR). On the other hand, we observe that if we combine the SR-NP method with the simplification strategy driven only by the geometric information, we need a mesh with at least 1400 vertices to get an estimate with a lower median MSE than the IHK method. This confirms the importance of including the data information in the mesh simplification procedure. Now, to quantitatively verify these results, we use pairwise Wilcoxon tests [30]. The pairwise Wilcoxon test is a non-parametric statistical hypothesis test that is used here to assess whether the MSEs of SR-NP estimates are significantly lower than the MSEs of the IHK estimates. In particular, the lower the p-value for this test the stronger the statistical evidence that the distribution of MSEs for the SR-NP estimators is stochastically lower than the corresponding distribution for the IHK estimators. A p-value smaller than 0.05 is considered significant; values smaller than 0.001 are considered strongly significant. In the last column of Table 1, we provide the p-values for pairwise Wilcoxon tests. These values verify that the estimates obtained via the SR-NP method on the original mesh ($m = 2527$) and on all the `Data+Geo` simplified meshes have significantly lower MSEs than the estimates obtained via the IHK method. While the `OnlyGeo` simplified meshes require that at least $m \geq 1600$ vertices to produce significantly lower MSEs with respect to the IHK method.

Finally, we remark that we are able to produce quality statistical estimates (in terms of MSEs) similar to the ones associated with the original mesh by working, for instance, on a simplified mesh with 1400 vertices. The mesh simplification procedure detailed in Section 2 takes 739 seconds and leads to a reduction of the computational time for the statistical analysis from 156 seconds on the original mesh to 66 seconds on the `Data+Geo` simplified mesh. Notice that the mesh simplification code has not yet been optimized. Several suggestions for future improvements in this direction are provided in Section 6. As an alternative, by adopting an offline/online paradigm, we may assume to perform the mesh simplification in an offline phase, while developing an online statistical analysis. As expected, the IHK method is computationally much cheaper taking only 4 seconds, at a price of a less accurate statistical analysis as highlighted by Table 1.

As described in [29], a numerical simulation may be corrupted by just a few "bad elements", i.e., extremely stretched triangles. To evaluate the level of distortion of the resulting mesh, we consider the so-called aspect ratio, q [29]. If $q(T) \approx 0$, the triangle T is stretched, while, for triangles close to the equilateral one, we have $q(T) \approx 1$. Using the pawn geometry, we analyzed the mean and minimum values of the aspect ratio of the meshes obtained via the `Data+Geo` approach. For all the levels of simplification, the mean of q is around 0.7, while a more significant variation is exhibited by the minimum value of q ranging from 2.32×10^{-4} to 2.35×10^{-2}. Despite the presence of very stretched triangles in the simplified mesh, the accuracy of the associated statistical analysis is really satisfying. A more through investigation of this issue may be of interest in the future.

5 Application to cortical surface data

In this section, we apply the proposed approach to the cortical surface geometry in Figure 1 (left). In this case, we are dealing with an original mesh with 40962 nodes. First, we apply the proposed method to a simulation study. Then, we apply it to the real cortical surface thickness data shown in Figure 1 (right) and studied in [3] and [4].

As in the previous section, we simulate noisy data on the cortical surface mesh by generating fifty test functions via (32) and by adding independent normally distributed errors with mean zero and a standard deviation 0.5 to the function values at each of the data locations. We compare the SR-NP method using the two mesh simplification strategies proposed in Section 2.3 with several levels of simplification to the IHK results on the full mesh. For the IHK method, we set the bandwidth $B = 1$, as suggested in [4] for data over this cortical surface mesh, and use the F-

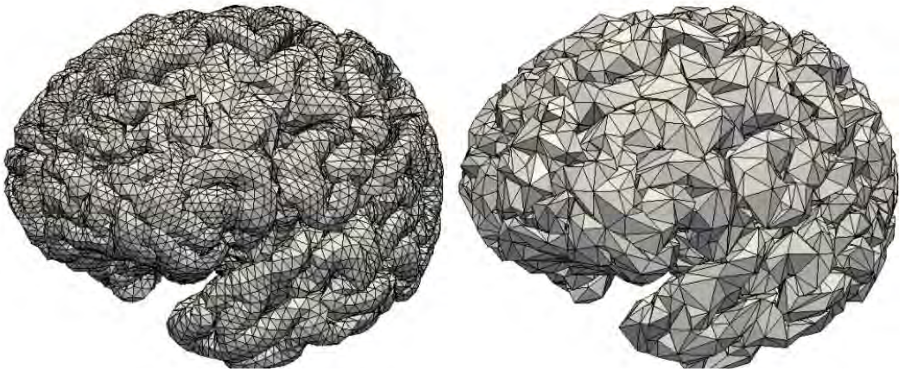

Figure 16: Example of cortical surface mesh simplification: original mesh with 40962 nodes (left); simplified mesh via the `Data+Geo` approach with 10000 nodes (right).

test criterium to determine the level of smoothness as in Section 4. For the SR-NP method, we consider three levels of mesh simplification, i.e., we generate simplified meshes with $m = 10000,\ 15000,\ 20000$ nodes (see Figure 16) by applying both the `Data+Geo` and the `OnlyGeo` approaches. For this test case, we do not give the SR-NP estimate over the original cortical surface mesh because it is computationally expensive.

Table 2 reports the median MSEs and the corresponding IQRs over the fifty simulation replicates. The SR-NP method consistently produces better results than the IHK method, while using less than half the original nodes. The employment of a more complex surface does not seem to compromise the performances of the proposed procedure. As in the pawn test case, the SR-NP estimates computed over the `Data+Geo` simplified meshes are better than the ones computed over the `OnlyGeo` meshes. The low p-values of pairwise Wilcoxon tests verify that the distribution of MSEs for the SR-NP estimators are stochastically lower than the corresponding distribution for the IHK estimators. Figure 18 displays the box plots of the MSE values in Table 2. We recognize the same trend as in Figure 15 where the `Data+Geo` simplification produces excellent results (with lower errors and more robust estimates) using fewer nodes. Figure 17 shows a simulation replicate: an example of a test function generated by (32) in (a), the corresponding level of noise in (b), the IHK estimate obtained on the original mesh in (c), the SR-NP estimate on the 10000 node mesh yielded by the `Data+Geo` and the `OnlyGeo` simplifications in (d) and (e), respectively. The SR-NP method is better at detecting variation in the data. This is most evident in the right hemisphere of the cortical surface.

Concerning the improvement in terms of computational effort, we remark that also in this case the employment of a simplified mesh greatly reduces the CPU times. We are able to produce quality statistical estimates (in terms of MSEs) by

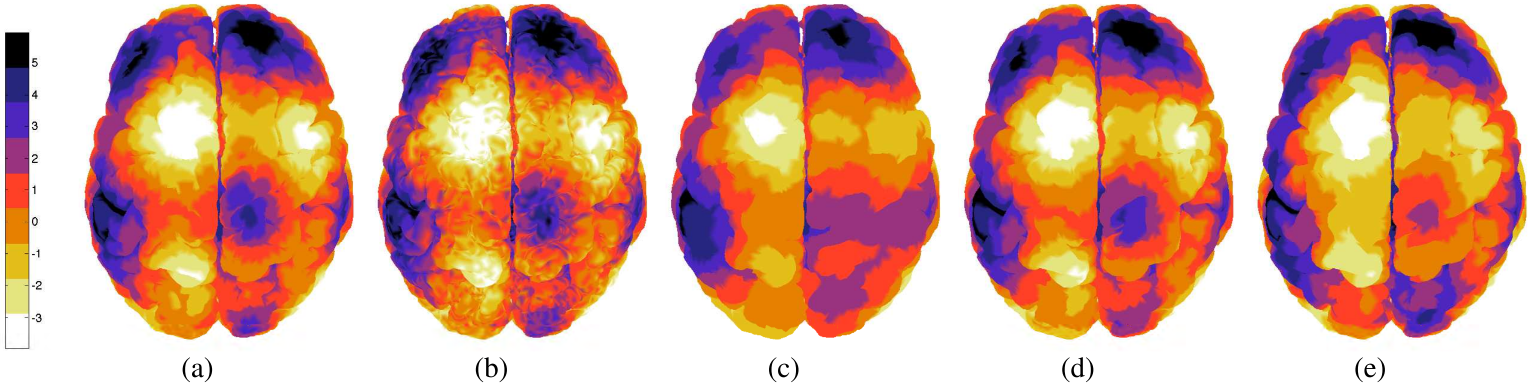

Figure 17: Example of a simulation on the cortical surface mesh with 40962 nodes: (a) the test function, (b) the data with noise, (c) the IHK estimate using the original mesh, (d) the SR-NP estimate on the 10000 node mesh produced by the `Data+Geo` simplification, (e) the SR-NP estimate on the 10000 node mesh generated by the `OnlyGeo` simplification. The color maps are obtained by linear interpolation of the data at $\mathbf{x}_j$ for j=1,...,40962.

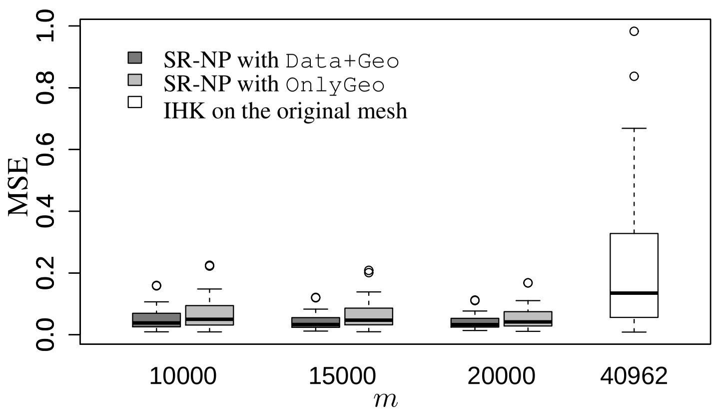

Figure 18: Box-plots of the Mean Square Errors (MSEs) of the estimates over the 50 simulation repetitions over the cortical surface.

Table 2: Median (IQR) of the Mean Square Errors of the estimates obtained in the 50 simulation repetitions over the cortical surface and p-values of pairwise Wilcoxon tests verifying that the MSE of the SR-NP estimates are significantly lower than those for IHK estimates.

Method	m	Simplification	MSE	p-value
SR-NP	10000	`Data+Geo`	0.0383 (0.0427)	4.399×10^{-10}
		`OnlyGeo`	0.0501 (0.0614)	4.399×10^{-10}
SR-NP	15000	`Data+Geo`	0.0332 (0.0310)	5.275×10^{-10}
		`OnlyGeo`	0.0473 (0.0540)	4.674×10^{-10}
SR-NP	20000	`Data+Geo`	0.0328 (0.0281)	5.951×10^{-10}
		`OnlyGeo`	0.0432 (0.0476)	5.275×10^{-10}
IHK	40962		0.1349 (0.2662)	

working on a mesh with less than quarter of the original vertices, i.e., 10000 vertices. This level of simplification reduces the computational time for the analysis from 3668 seconds on the original mesh to 544 seconds on the simplified mesh. As for the pawn test case, the time demand for the mesh simplification method is considerable. It takes exactly 20936 seconds. In the absence of a code optimization, an offline/online paradigm may justify the non-optimal performance of the mesh simplification algorithm. We note that the IHK method is computationally cheaper per iteration. Nevertheless, since for each simulation the iterative residual fitting algorithm needs more iterations to satisfy the F-test criterion, we have that, for the cortical surface simulations, the IHK method takes a considerably longer time with respect to the SR-NP approach. In general, we can state that the extra computational time possibly required by the SR-NP approach allows us to get substantially better estimates which may be enriched via the inclusion of covariates.

Now, let us consider real cortical surface thickness data. For the IHK method, we set the bandwidth $B = 1$ and the number of iterations selected for smoothing to 200, as suggested for this data set in [4]. For the SR-NP method, we adjust the smoothing parameter λ to have about the same amount of smoothing as the IHK method. Essentially, this level of smoothing is set only to highlight areas of interest, i.e., to identify regions with high or low areas of thickness. The results are shown in Figure 19. Notice that the SR-NP method with the `Data+Geo` simplification is able to identify an additional area of the low thickness (circled in Figure 19 (bottom-left)) with respect to what is detected by the IHK approach and by the same SR-NP method combined with the `OnlyGeo` simplification. This low thickness area is recognizable in the original thickness data (see Figure 19 (top-left)).

6 Conclusions and future developments

The mesh simplification method based on both geometry and data controls consistently produces meshes that lead to quality statistical estimates via the SR-NP method and outperforms the comparison methods. In particular, the proposed simplification method effectively builds a mesh that approximates the original geometry and properly (in terms of displacement and distribution) associates the data with the new geometry allowing for subsequent statistical estimates with good inferential properties.

In future research, it is of interest to solve the estimation problem (9) directly on the non-planar domain Γ, without resorting to a conformal map. This would undoubtedly lead to a considerable saving in terms of computation time as well as to a possible improvement in terms of accuracy since the flattening phase could be skipped. Nevertheless, mapping the estimates to a reference domain could still be of interest in the context of a multiple patients analysis, allowing for comparisons across different geometries. Mapping to a reference is indeed a standard way of proceeding in current population studies, where the data for each patient are registered to a template surface for reference coordinates (see, e.g., [3] and [4]). The

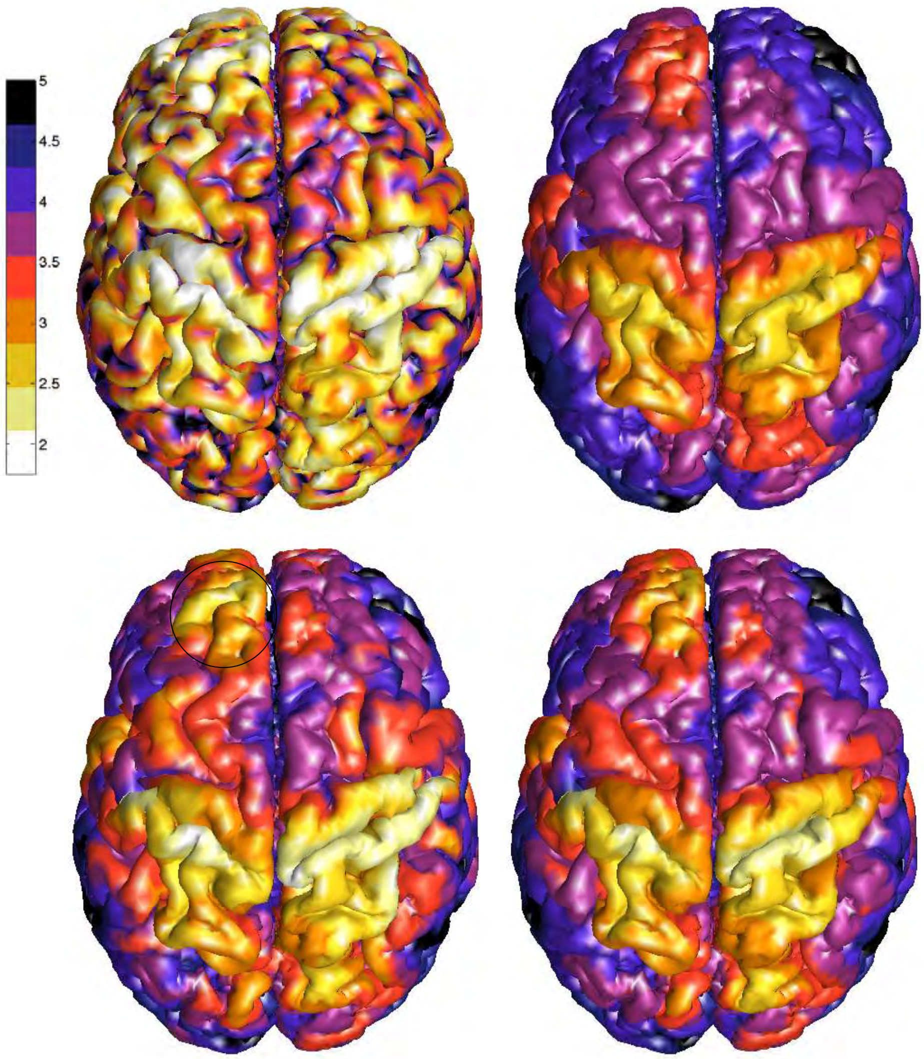

Figure 19: The SR-NP estimate on the original mesh (top-left), the IHK estimate computed on the original mesh (top-right), the SR-NP estimate on the 10000 node mesh provided by `Data+Geo` simplification (bottom-left), the SR-NP estimate on the 10000 mesh generated by the `OnlyGeo` simplification (bottom-right). The color map shows the cortical surface thickness. For reference the thickness data on the original cortical surface mesh is shown in Figure 1.

development of full inferential and uncertainty quantification tools for these population studies within our proposed framework will be the object of a future work

Among our future goals, we aim at providing a more rigorous approach to select the weights involved in the cost function definition, for instance, by applying some proper optimization procedure that depends on the application and/or the geometry at hand. Some preliminary studies have been carried out, so far, only on simple geometric configurations. For instance, it would be of interest to explore possible ways to relate the number of the mesh nodes, m, with the MSEs of the estimates. The goal would be to identify the minimal number of nodes that would lead to a desired level of MSE for a specific application. Furthermore, certain computational improvements of the simplification procedure are planned, such as, the employment of a greedy strategy during the edge contraction step. This would drastically reduce the computational time for the simplification procedure. Finally, we aim at extending the simplification process to different types of manifolds. For example, by properly accounting for the boundary edges, the contraction process can be adapted to manifolds with open boundaries or holes such as in the internal carotid artery application considered in [9].

Acknowledgments We are grateful to John Aston for insightful discussions. This work is supported by MIUR Ministero dell'Istruzione dell'Università e della Ricerca, *FIRB Futuro in Ricerca* research project "Advanced Statistical and Numerical Methods for the Analysis of High Dimensional Functional Data in Life Sciences and Engineering" (see `http://mox.polimi.it/users/sangalli/firbSNAPLE.html`), and by the program Dote Ricercatore Politecnico di Milano - Regione Lombardia, research project "Functional Data Analysis for Life Sciences". We would also like to thank the reviewers for their useful suggestions which have allowed us to considerably improve the quality of the paper.

7 References

References

[1] S. Angenent, S. Haker, A. Tannenbaum, and R. Kikinis, *On the Laplace-Beltrami operator and brain surface flattening*, IEEE Trans. Med. Imaging, 18 (1999), pp. 700–711.

[2] A. I. Bobenko and B. Springborn, *A discrete Laplace-Beltrami operator for simplicial surfaces*, Discrete Comput. Geom., 38 (2007), pp. 740–756.

[3] M. K. Chung, S. Robbins, and A. C. Evans, *Unified statistical approach to cortical thickness analysis*, in Information Processing in Medical Imaging (IPMI), Lect. Notes in Comput. Sci. Eng., SpringerVerlag, 2005, pp. 627–638.

[4] M. K. Chung, S. M. Robbins, K. M. Dalton, R. J. Davidson, A. L. Alexander, and A. C. Evans, *Cortical thickness analysis in autism with heat kernel smoothing*, NeuroImage, 25 (2005), pp. 1256–1265.

[5] A. M. Dale, B. Fischl, and M. I. Sereno, *Cortical surface-based analysis – I. Segmentation and surface reconstruction*, NeuroImage, 9 (1999), pp. 179–194.

[6] T. K. Dey, A. N. Hirani, B. Krishnamoorthy, and G. Smith, *Edge contractions and simplicial homology*, CoRR, (2013).

[7] U. Dierkes, S. Hildebrandt, and F. Sauvigny, *Minimal Surfaces*, vol. 1, Springer, Heidelberg, 2 ed., 2010.

[8] G. Dziuk, *Finite elements for the Beltrami operator on arbitrary surfaces*, Calc. Var. Partial Differential Equations, 1357 (1988), pp. 142–155.

[9] B. ETTINGER, S. PEROTTO, AND L. M. SANGALLI, *Spatial regression models over two-dimensional manifolds*, Tech. Rep. 54/2012, MOX - Dipartimento di Matematica-Politecnico di Milano, December 2012.

[10] M. GARLAND AND P. S. HECKBERT, *Surface simplification using quadric error metrics*, in Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '97, New York, USA, 1997, ACM Press/Addison-Wesley Publishing Co., pp. 209–216.

[11] G. GOLUB, M. HEATH, AND G. WAHBA, *Generalized cross-validation as a method for choosing a good ridge parameter*, Technometrics, 21 (1979), pp. 215–223.

[12] D. HAGLER, JR., A. SAYGIN, AND M. SERENO, *Smoothing and cluster thresholding for cortical surface-based group analysis of fMRI data*, NeuroImage, 33 (2006), pp. 1093–1103.

[13] T. HASTIE, R. TIBSHIRANI, AND J. H. FRIEDMAN, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, New York: Springer-Verlag, 2001.

[14] H. HOPPE, T. DEROSE, T. DUCHAMP, J. MCDONALD, AND W. STUETZLE, *Mesh optimization*, in Proceedings of the 20th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '93, New York, USA, 1993, ACM, pp. 19–26.

[15] F. C. HUANG, B. Y. CHEN, AND Y. Y. CHUANG, *Progressive deforming meshes based on deformation oriented decimation and dynamic connectivity updating*, in Proceedings of the 2006 ACM SIGGRAPH/Eurographics symposium on Computer animation, SCA '06, Eurographics Association, 2006, pp. 53–62.

[16] T. HUGHES, *The Finite Element Method : Linear Static and Dynamic Finite Element Analysis*, Englewood Cliffs, N.J. Prentice-Hall International, 1987.

[17] M. K. HURDAL AND K. STEPHENSON, *Discrete conformal methods for cortical brain flattening*, NeuroImage, 45 (2009), pp. S86 – S98.

[18] T. KANAYA, Y. TESHIMA, K. KOBORI, AND K. NISHIO, *A topology-preserving polygonal simplification using vertex clustering*, in Proceedings of the 3rd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia, GRAPHITE '05, New York, USA, 2005, ACM, pp. 117–120.

[19] W. H. KIM, D. PACHAURI, C. HATT, M. K. CHUNG, S. JOHNSON, AND V. SINGH, *Wavelet based multi-scale shape features on arbitrary surfaces for cortical thickness discrimination*, Adv. Neural Inf. Process. Syst., 25 (2012), pp. 1250–1258.

[20] B. LÉVY AND J.-L. MALLET, *Constrained discrete fairing for arbitrary meshes*, tech. rep., GOCAD Consortium, 1999.

[21] J. L. LIONS AND E. MAGENES, *Non-Homogeneous Boundary Value Problems and Applications*, vol. III, Springer-Verlag, New York, 1973.

[22] T. MÖLLER, *A fast triangle-triangle intersection test*, J. Graph. Tools, 2 (1997), pp. 25–30.

[23] U. PINKALL AND K. POLTHIER, *Computing discrete minimal surfaces and their conjugates*, Exp. Math., 2 (1993), pp. 15–36.

[24] J. RAMSAY AND B. W. SILVERMAN, *Functional Data Analysis*, Springer-Verlag, New York, 2 ed., 2005.

[25] R. RONFARD AND J. ROSSIGNAC, *Full-range approximation of triangulated polyhedra*, Computer Graphics Forum, 15 (1996), pp. 67–76.

[26] J. ROSSIGNAC AND P. BORREL, *Multi-resolution 3d approximations for rendering complex scenes*, in Modeling in Computer Graphics: Methods and Applications, Springer-Verlag, 1993, pp. 455–465.

[27] L. M. SANGALLI, J. O. RAMSAY, AND T. O. RAMSAY, *Spatial spline regression models*, J. R. Stat. Soc. Ser. B Stat. Methodol., 75 (2013), pp. 1–23.

[28] W. J. SCHROEDER, J. A. ZARGE, AND W. E. LORENSEN, *Decimation of triangle meshes*, in Computer Graphics SIGGRAPH '92 Proceedings, 1992, pp. 65–70.

[29] J. R. SHEWCHUK, *What is a good linear finite element? - interpolation, conditioning, anisotropy, and quality measures*, tech. rep., In Proceedings of the 11th International Meshing Roundtable, 2002.

[30] F. WILCOXON, *Individual comparisons by ranking methods*, Biometrics Bulletin, 1 (1945), pp. 80–83.