Geodesic distance approximation using a surface finite element method for the p-Laplacian

Hannah Potgieter[a], Razvan C. Fetecau[a], Steven J. Ruuth[a]

[a] *Department of Mathematics, Simon Fraser University, Burnaby, British Columbia Canada*

Abstract

We use the p-Laplacian with large p-values in order to approximate geodesic distances to features on surfaces. This differs from Fayolle and Belyaev's (2018) [1] computational results using the p-Laplacian for the *distance-to-surface* problem. Our approach appears to offer some distinct advantages over other popular PDE-based distance function approximation methods. We employ a surface finite element scheme and demonstrate numerical convergence to the true geodesic distance functions. We check that our numerical results adhere to the triangle inequality and examine robustness against geometric noise such as vertex perturbations. We also present comparisons of our method with the heat method from Crane et al. [2] and the classical polyhedral method from Mitchell et al. [3].

Keywords: p-Laplacian, distance estimation, geodesic distance, finite element method, surface PDE

1. Introduction

Problems involving the p-Laplace operator appear in a variety of applications including data clustering (Bühler and Hein [4]), image processing (Blomgren et al. [5]; Chen et al. [6], Caselles et al. [7]), distance function approximation (Fayolle & Belyaev [1]), optimal transport (Fayolle & Belyaev [1]), non-Newtonian fluids (Ružička [8]), and shape morphing (Cong et al. [9]). In this paper, we focus on the application of the p-Laplacian for computing geodesic distances on surfaces.

Bhattacharya et al. [10] consider the following boundary value problem (BVP) for the p-Laplacian in a bounded domain $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega$:

$$-\Delta_p u_p = 1, \qquad \text{in } \Omega, \tag{1a}$$

$$u_p = 0, \qquad \text{on } \partial\Omega \tag{1b}$$

for $2 \leq p \leq \infty$. The p-Laplacian is defined by $\Delta_p u = \operatorname{div}\left(|\nabla u|^{p-2}\nabla u\right)$; note that for $p = 2$ it reduces to the usual Laplacian. Bhattacharya et al. [10] show that the solution $u_p(x)$ to the BVP (1a)-(1b) converges strongly as $p \to \infty$, in a certain functional sense, to $\operatorname{dist}(x, \partial\Omega)$, where $\operatorname{dist}(\cdot, \partial\Omega)$ denotes the distance function to the boundary $\partial\Omega$. The problem (1a)-(1b) is fully nonlinear and degenerate, unless $p = 2$ in which case it is a linear Poisson problem. Taking $p = 2$ yields what is referred to as the 'Poisson distance' and $p > 2$ gives the 'p-Poisson distance'.

Fayolle & Belyaev [1] use (1a)-(1b) to compute the *distance-to-surface* problem, where Ω is a subset of $\mathbb{R}^3$ such that $\partial\Omega$ represents the surface of interest. In this paper, we are interested in the more general *distance-to-feature* problem on *surfaces* both with and without boundaries. A 'feature' can be any collection of points or curves on a given surface, to which we want to compute the distance. Such points or curves may lie either in the interior or on the boundary of the surface. We point out that our interest is to compute the surface geodesic (intrinsic) distance function to such features. This is in contrast with Fayolle & Belyaev [1] who compute the extrinsic distance (in $\mathbb{R}^3$) to a given surface.

Computer graphics problems frequently call for computation of distances to features on a discrete surface. The distance-to-feature problem is practically useful for a variety of tasks such as segmentation, deformation, and path planning as mentioned in [1], [11], and [12]. It is also useful for medical image analysis (see Naber et al. [13]). Belyaev & Fayolle [14] applied alternating direction method of multipliers (ADMM) schemes to different energy minimization problems seeking to achieve accurate distance function approximations, and showed in [1] that minimizing the p-Poisson energy yields the best results. Accordingly, we extend their work with the p-Laplacian to the distance-to-feature problem and perform more comprehensive numerical tests.

1.1. Related works

The problem of approximating geodesic distances on surfaces has been extensively studied, with a variety of approaches proposed. See, for example, [2], [11], [12],[14], [15], and [16]. These range from PDE-based methods to computational geometry techniques, each offering distinct advantages and limitations. Among these, the *heat method* of Crane, Weischedel, and Wardetzky [2] and the *polyhedral method* of Mitchell, Mount, and Papadimitriou [3] are two prominent approaches that we utilize for comparison.

The heat method [2], which computes geodesic distances using Varadhan's formula [17], is one of the most well-known PDE-based approaches. The heat method splits the problem into first finding the direction along which distance is increasing and then computing the distance itself. The method is computationally efficient, adaptable, robust against noisy surface data, and simple. However, the heat method has notable limitations. It struggles with boundary conditions on open surfaces, such as hemispheres, an issue observed in our experiments and also seen in Crane et al.'s experiments [2]. Additionally, the method often fails to satisfy the triangle inequality and produces smooth approximations [12]. Concurrent with our work, Feng and Crane [18] introduced the Signed Heat Method (SHM) for computing generalized signed distance functions. By formulating vector heat diffusion with homogeneous Neumann conditions directly on the vector field, rather than on scalar potentials, SHM yields suitable distance approximations at surface boundaries without requiring heuristics or post-processing corrections. Our comparisons are performed against the (scalar) heat method [2].

The polyhedral method [3], in contrast, operates on polyhedral surfaces and employs a "continuous Dijkstra technique," which adapts Dijkstra's algorithm [19] from graph structures to polyhedral surfaces. By carefully incorporating geometric considerations, the method computes exact geodesics along polyhedral surfaces. It is typically both computationally efficient and highly accurate when applied to tessellated surfaces. We include this method in our comparisons to better understand the properties of our solutions, and refer to a recent survey paper [20] for further details on the method, its attributes and limitations.

Other notable approaches include Sethian's fast marching method [21] for numerical approximation of solutions to the Eikonal equation $|\nabla u(x)| = 1$, Solomon et al.'s [12] optimal transport inspired framework, Edelstein et al.'s [22] PDE-based convex optimization approach, as well as graph-based methods, such as those proposed by Adikusuma et al. [23].

In this paper, we present a new algorithm for geodesic distance computation and evaluate its key properties. As part of this analysis, we compare our approach with the heat method and the polyhedral method, focusing on illustrative examples to highlight the key properties of our approach. We do not aim to provide an exhaustive comparison in these tests, nor do we compare with other well-established methods. For a broader overview of geodesic distance approximation techniques and their respective merits, see [20].

1.2. Contributions

This paper introduces a new method for approximating intrinsic geodesic distances to features on surfaces. This *p-Poisson method* is formulated using a p-Laplace equation intrinsic to the surface, employing appropriate mixed boundary conditions to define the feature sets. In the large p limit, the solution to the equations provides the solution to the intrinsic geodesic problem.

In our implementation, the Alternating Direction Method of Multipliers (ADMM) is employed alongside a surface finite element method with piecewise linear elements to solve the mixed boundary value problem. Convergence of the method is assessed through numerical studies on the hemisphere and torus, as well as on more complex surface meshes, with comparisons to the heat method and the polyhedral method. We find convergence to the desired intrinsic geodesic distance. Additionally, our proposed method provides an improved treatment of boundaries relative to the heat method in certain examples. Robustness tests further demonstrate the preservation of the triangle inequality and stability against geometric perturbations.

The aim of this work is not to compete with state-of-the-art geodesic distance approximation methods on surface meshes, which are generally faster, especially in high-accuracy scenarios. However, our PDE formulation does not require a mesh and offers a path to implementation for other surface representations, such as level sets or point clouds, through suitable numerical PDE techniques, including mesh-free methods in the case of point clouds. As well, this work provides insights into the behavior of the surface p-Laplacian for large p.

The method has certain limitations. Most notably, it is slower than the polyhedral method for surfaces defined on meshes, particularly when highly accurate solutions are required. Additionally, suitable solvers must be chosen to accommodate the surface representation and its regularity. Our implementation focuses on triangle and quadrilateral surface meshes using the well-known FEM package deal.II [24] coupled with ADMM; other surface representations and solvers are not explored in this work. Finally, we note that correctness is primarily assessed through numerical experiments.

2. Problem formulation

Consider a generic surface S and the p-Laplacian operator on S given by

$$\Delta_p^S u = \operatorname{div}^S\Big(|\nabla^S u|^{p-2}\nabla^S u\Big),$$

where div^S and ∇^S denote the surface divergence and gradient, respectively. Let $\Gamma_1 \subset S \cup \partial S$ denote the 'feature' set, i.e., the set to which we want to compute the geodesic distance. The set Γ_1 may consist of any collection of points or curves from the interior or boundary of S. Denote the open set Ω by $\Omega = S \setminus \Gamma_1$ and $\Gamma_2 = \partial S \setminus \Gamma_1$; note that $\partial\Omega = \Gamma_1 \cup \Gamma_2$, where the union is disjoint. Figure 1 illustrates examples of Γ_1 and Γ_2 on the open hemisphere S. Note that Γ_1 may consist of points, closed curves, and open curves, and may also overlap with the natural surface boundary ∂S.

For $2 \leq p \leq \infty$ fixed, we consider the following BVP for the surface p-Laplacian:

$$-\Delta_p^S u_p = 1, \qquad \text{in } \Omega, \tag{2a}$$

$$u_p = 0, \qquad \text{on } \Gamma_1 \subseteq \partial\Omega, \tag{2b}$$

$$\frac{\partial u_p}{\partial n} = 0, \qquad \text{on } \Gamma_2 \subseteq \partial\Omega. \tag{2c}$$

Note that Dirichlet conditions are imposed on Γ_1 (the feature set) and Neumann conditions are imposed on Γ_2, which represents the unconstrained portion of the surface boundary.

The p-Poisson problem can be derived from the minimization of the variational integral

$$E_p(u) \equiv \frac{1}{p}\int_\Omega |\nabla^S u|^p dA - \int_\Omega u\, dA. \tag{3}$$

Setting $0 < p < 1$ would lead to a non-convex energy, however, we are interested in the range $p \in [2, \infty)$. The Euler-Lagrange equations for the functional (3) give rise to equation (2a). We enforce the boundary condition (2b), as the distance from the feature set to itself must be zero. This leaves (2c) to be justified. To this purpose we use the notion of *natural* boundary conditions in calculus of variations [25] as is done by Edelstein et al. [22]. Note that there are no specific boundary conditions on Γ_2 which need to be imposed beforehand, by the distance-to-feature problem that we consider. Hence, on the unconstrained portion Γ_2 of $\partial\Omega$, we impose the natural boundary condition, which achieves the lowest energy among all possible boundary conditions [25].

By a standard calculation, the natural boundary condition corresponding to the energy (3) (with (2b) already imposed), is given by

$$|\nabla^S u|^{p-2}(\nabla^S u \cdot \mathbf{n}) = 0 \quad \text{on } \Gamma_2, \tag{4}$$

where $\mathbf{n}$ is the outward unit normal to $\partial\Omega$. In order to satisfy (4), we need to have either $|\nabla^S u|^{p-2} = 0$ or $\nabla^S u \cdot \mathbf{n} = 0$ for all points on $\Gamma_2 \subset \partial\Omega$. We are interested in a distance function approximation, so as $p \to \infty$ it is desired that $|\nabla^S u_p|$ will be close to 1. Hence, the homogeneous Neumann boundary condition (2c) follows. Below, we present further mathematical justification for the mixed BVP (2a)-(2c), and at the end of this section, we go into more detail for a particular example in 1D.

The particular case when $\Gamma_1 = \partial\Omega$, $\Gamma_2 = \emptyset$, and Ω is a bounded domain in $\mathbb{R}^n$, was considered in the mathematical literature by Bhattacharya et al. [10], and in the computational work on distance approximation of Fayolle and Belyaev [1]. In this restrictive setting, problem (1a)-(1b) computes (in the limit $p \to \infty$) the distance in the Euclidean space to the boundary of a set $\Omega \subset \mathbb{R}^n$. In an extension of the work from [10], [26] considered the more

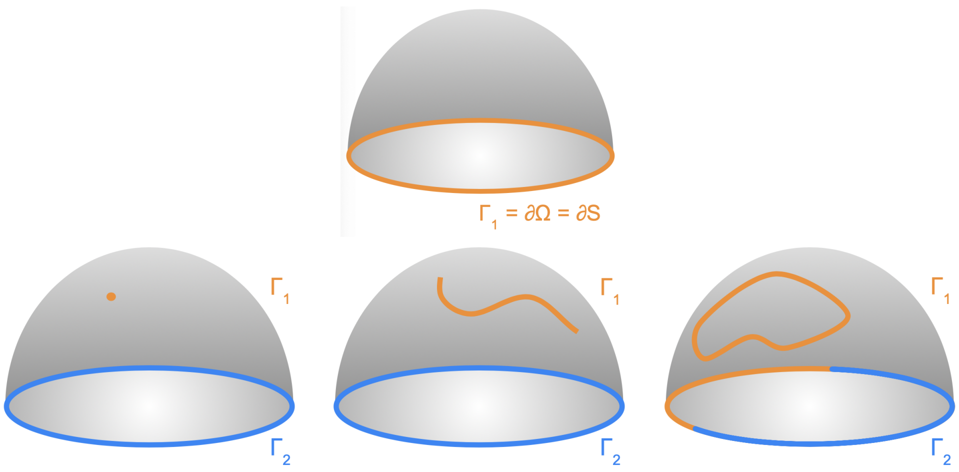


Figure 1: S is the open hemisphere and Γ_1 (indicated in orange) represents the feature set. Top row: Distance-to-boundary problem ($\Gamma_1 = \partial S, \Gamma_2 = \emptyset$). Bottom row: Distance-to-feature problem. Γ_2 is indicated in blue.

general case when Γ_1 is a strict subset of $\partial\Omega$ (hence, $\Gamma_2 \neq \emptyset$). Specifically, the authors of [26] considered the mixed BVP (2a)-(2c) on smooth convex subsets of $\mathbb{R}^n$, and investigated the limit $p \to \infty$ of its weak solutions u_p. They proved (see [26, Remark 2.2]) that

$$\lim_{p\to\infty} u_p(x) = \operatorname{dist}(x, \Gamma_1). \tag{5}$$

The convergence is, in fact, uniform in $\overline{\Omega}$. From an analysis point of view, the approach taken in [26] to study the $p \to \infty$ limit, is very similar to that from [10]. The reason the methods extend with very few modifications from [10] (where $\Gamma_2 = \emptyset$) to the mixed BVP with $\Gamma_2 \neq \emptyset$, lies exactly in the fact that the homogeneous Neumann BC (2c) is natural [25].

In the present work, we are interested in geodesic distance approximation on surfaces. For this reason, the BVP (2a)-(2c) is set up on a subset Ω of a surface S using the surface p-Laplacian, and we compute the geodesic distance function to the feature set $\Gamma_1 \subset \partial\Omega$. The feature set can be very diverse, as illustrated in Figure 1; Γ_1 may or may not have any points on the surface's boundary ∂S. When $\Gamma_1 = \partial S$ (see top row in Figure 1), then $\Omega = S$, and the distance-to-feature problem simplifies to the distance-to-boundary problem as in Bhattacharya et al. [10], except that here we are interested in the geodesic, rather than Euclidean, distance. For numerical purposes, large p-values are of interest. Unfortunately, large p-values lead to ill-conditioned problems and are therefore numerically challenging to solve [27].

We further note that the analysis in [10, 26] can be extended to compact Riemannian manifolds using the Sobolev and Kondrachov embedding theorems on manifolds from [28]. Hence, one can prove rigorously the limit (5) in general surface settings, as considered in our work. Also, from a PDE analysis point of view, the case when Γ_1 contains isolated points (see left plot in bottom row of Figure 1), is degenerate. In this case, one can consider Γ_1 to be the boundary of an arbitrarily small geodesic ball around the single point.

Exact 1D solution. For concreteness, we show a simple 1D example with an exact solution. A similar calculation may be performed for a radially symmetric problem in 2D, however solutions are not as simple and involve hypergeometric functions. Consider $S = (-1, 1)$ with feature set $\Gamma_1 = \{0\}$. Then, $\Omega = (-1, 0) \cup (0, 1)$ and $\Gamma_2 = \{-1, 1\}$. The exact solution for this problem is

$$\text{dist}(x, \Gamma_1) = |x|.$$

The one-dimensional p-Laplacian is given by $\Delta_p u_p(x) = (|u'|^{p-2}u')'$, and the BVP problem (2a)-(2c) becomes

$$-\left(|u'|^{p-2}u'\right)' = 1, \quad 0 < |x| < 1, \tag{6a}$$

$$u_p(0) = 0, \tag{6b}$$

$$u_p'(-1) = u_p'(1) = 0. \tag{6c}$$

The exact solution of (6a)-(6c) is

$$u_p(x) = -\frac{p-1}{p}\left(1 - |x|\right)^{\frac{p}{p-1}} + \frac{p-1}{p}. \tag{7}$$

Using (7) one can then show immediately that for any $x \in \Omega$ fixed,

$$\lim_{p\to\infty} u_p(x) = |x| = \text{dist}(x, \Gamma_1). \tag{8}$$

We note that the convergence (8) is in fact uniform on Ω. Indeed, use

$$\frac{\partial}{\partial x}\left[|x| - u_p(x)\right] = \text{sign}(x)\left(1 - (1 - |x|)^{\frac{1}{p-1}}\right),$$

for $x \in \Omega$. Since $(1 - (1 - |x|)^{\frac{1}{p-1}}) \geq 0$ for all $x \in \Omega$, we infer that $|x| - u_p(x)$ is a monotone function and hence,

$$\Big||x| - u_p(x)\Big| \leq \big|| \pm 1| - u_p(\pm 1)\big| \leq 1 - \frac{p-1}{p} = \frac{1}{p}, \qquad \text{for all } x \in \Omega.$$

The uniform convergence in (8) can then be concluded.

Figure 2 shows the exact solution (7) for various values of p, and their corresponding errors, $|x| - u_p(x)$. Here we use (2c), ensuring a well-posed problem for our computations with finite p. In the $p \to \infty$ limit, $u_p(x)$ violates the Neumann condition (6c) (see the zoom-in of the left plot in Figure 2), which is the desired behaviour for the distance problem. Accordingly, in computations with large p the error from the homogeneous Neumann condition (6c) does not interfere with the linear convergence rate of $1/p$. This supports our choice of homogeneous Neumann boundary conditions on Γ_2.

3. Surface finite element method

The continuous surface Ω is replaced by a piecewise polynomial surface Ω_h; this introduces a geometric error between Ω and Ω_h. We will use a finite element space:

$$S_h = \{\phi_h \in C(\Omega_h) : \phi_h|_T \text{ is in } P^1 \text{ for each } T \in \mathcal{T}_h\},$$

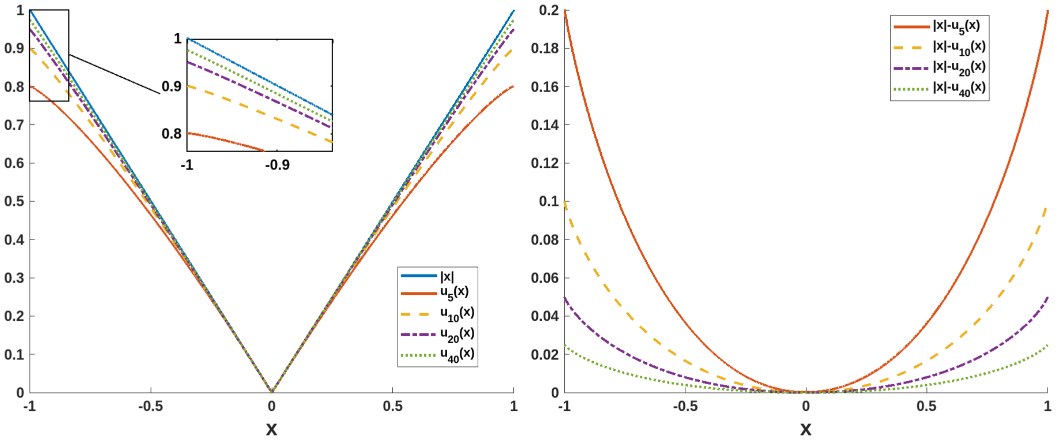


Figure 2: One-dimensional example, with $S = (-1, 1)$ and $\Gamma_1 = \{0\}$. Left: the exact solution $u_p(x)$ (see (6a)-(6c) and (7)) converges to $|x| = \text{dist}(x, \Gamma_1)$ as p increases. Right: the corresponding pointwise errors $|x| - u_p(x)$, for various p-values.

where P^1 is the space of linear elements on $\mathcal{T}_h$, the collection of faces on the surface mesh. The corresponding lifted finite element space is

$$S_h^l = \{\varphi_h = \phi_h^l : \phi_h \in S_h\},$$

where the superscript denotes the constant extension from Ω in the normal direction or lift onto Ω.

Convergence has been established for the surface finite element method (SFEM) with linear elements. In particular, for the surface Poisson problem, $\Delta^S u = f$, Dziuk & Elliot [29] show that if $||f - f_h||_{L_2(\Omega)} \leq c_f h^2$, then

$$||u - u_h||_{L_2(\Omega)} \leq c_1 h^2, \quad ||\nabla^S (u - u_h)||_{L_2(\Omega)} \leq c_2 h,$$

which is analogous to the Euclidean result.

Distance functions are typically nonsmooth which means we do not see improved convergence rates with higher order elements. Therefore, we consider only first order elements. We choose to use SFEM due to the nonsmooth nature of distance functions. Using a finite difference discretization paired with the closest point embedding method [30] suffices for smooth problems. However, finite difference discretizations fail to converge for the examples we consider. The divergence structure of the p-Laplacian allows us to employ a variational finite element method which is more robust against nonsmooth data.

Alternating direction method of multipliers (ADMM). To solve the nonlinear p-Poisson problem, we use two methods: a standard Newton scheme and ADMM. Note that throughout the paper, we will present only the results obtained with ADMM. Using the Newton scheme we were able to obtain very similar results; however, Newton is less robust for large p and requires more careful numerical treatment [27]. ADMM is a competitive tool in solving problems arising in data science due to its efficiency and computational advantages over other similar methods [31]. In particular, using an ADMM algorithm with a finite element discretized PDE problem can greatly improve practical employability over a more traditional

Newton iteration when working on a discrete computational domain represented by a large mesh.

We now present an ADMM algorithm for the p-Poisson problem. We note that we are minimizing functionals as in [32] which is a more general setting than the ADMM setup in [31]. As discussed above, this problem easily becomes ill-conditioned for large p-values so it is ideally suited to this approach. For the application of geodesic distance approximation, large p-values may be used to yield more accurate results. Thus, it is of interest to use a numerical solver which is robust against ill-conditioned systems.

Fayolle & Belyaev [1] consider p-Poisson problems and implement an ADMM algorithm for computing numerical solutions. Our implementation is very similar to that shown by Fayolle & Belyaev [1] but with different boundary conditions. We will see that using ADMM amounts to iterations which require solving a Poisson equation and a 1D polynomial equation.

In order to split the problem, a slack variable $\xi(x)$ is introduced in [1], which gives rise to the constrained problem

$$\begin{aligned} &\text{minimize} \quad \frac{1}{p}\int_\Omega |\xi|^p dA - \int_\Omega u\, dA, \\ &\text{subject to} \quad \xi = \nabla^S u, \end{aligned}$$

with boundary conditions of $u|_{\Gamma_1} = 0$ and $\frac{\partial u}{\partial n}|_{\Gamma_2} = 0$. Section 2 elaborates on the relation between minimizing the energy functional and these boundary conditions. In the SFEM discretization, the Dirichlet conditions are imposed by adding constraints to the nodes lying on Γ_1 and the homogeneous Neumann condition is imposed by disregarding the boundary term obtained from applying the divergence theorem. An ADMM algorithm can now be used on the convex optimization problem. Relaxing the constraint and adding a Lagrange multiplier term gives

$$\begin{aligned} L_\beta(\xi, u, y) &= \frac{1}{p}\int_\Omega |\xi|^p dA - \int_\Omega u dA + \int_\Omega y \cdot (\nabla^S u - \xi) dA + \frac{\beta}{2}\int_\Omega |\nabla^S u - \xi|^2 dA \\ &= \int_\Omega \left\{ \frac{1}{p}|\xi|^p - u + \frac{\beta}{2}\left|\xi - \left(\nabla^S u - \frac{y}{\beta}\right)\right|^2 - \frac{1}{2\beta}|y|^2 \right\} dA, \end{aligned}$$

where $\beta > 0$ and $y(x)$ is the vector of the Lagrange multipliers.

First, if u and y are fixed, the optimization with respect to ξ results in a 1D polynomial equation. The optimal ξ will minimize

$$\frac{1}{p}|\xi|^p + \frac{\beta}{2}\left|\xi - \left(\nabla^S u - \frac{y}{\beta}\right)\right|^2, \tag{9}$$

and therefore will be proportional to $\nabla^S u - \frac{y}{\beta}$, taking the form $c(x)\left(\nabla^S u(x) - \frac{y}{\beta}\right)$.

Heuristically, minimizing only $\frac{1}{p}|\xi|^p$ results in $\xi = 0$ and minimizing only $\frac{\beta}{2}\left|\xi - \left(\nabla^S u - \frac{y}{\beta}\right)\right|^2$ results in $\xi = \nabla^S u - \frac{y}{\beta}$ so it makes sense that the minimizer will be situated on the straight segment connecting the origin of coordinates with $\nabla^S u - \frac{y}{\beta}$. To illustrate this, we can separate ξ into parallel and orthogonal components relative to $\nabla^S u - \frac{y}{\beta}$ denoted, respectively, by $\xi_\parallel$ and $\xi_\perp$ so that $\xi = \xi_\parallel + \xi_\perp$. Plugging this into (9) gives

$$\begin{aligned}\arg\min_{\xi} &\left\{ \frac{1}{p}|\xi|^p + \frac{\beta}{2}\left|\xi - \left(\nabla^S u - \frac{y}{\beta}\right)\right|^2 \right\} \\ &= \arg\min_{\xi} \left\{ \frac{1}{p}|\xi|^p + \frac{\beta}{2}\left|\xi\right|^2 - \beta\left(\nabla^S u - \frac{y}{\beta}\right)^\top \xi_\parallel \right\},\end{aligned}$$

from which we can see that the optimal ξ is parallel to $\nabla^S u - \frac{y}{\beta}$.

For each $x \in \Omega_h$, using $\xi = c(x)\left(\nabla^S u(x) - \frac{y}{\beta}\right)$ leads to minimizing

$$\frac{1}{p}c^p\left|\nabla^S u - \frac{y}{\beta}\right|^p + \frac{\beta}{2}\left|\nabla^S u - \frac{y}{\beta}\right|^2 (c-1)^2, \tag{10}$$

or equivalently solving

$$\left|\nabla^S u - \frac{y}{\beta}\right|^{p-2} c^{p-1} + \beta(c-1) = 0 \tag{11}$$

for c [1]. The latter is obtained by differentiating (10) with respect to c and then dividing by $\left|\nabla^S u - \frac{y}{\beta}\right|^2$. This problem can be dealt with numerically using a Newton iteration. Note that for a given $x \in \Omega_h$, if $|\nabla^S u(x) - \frac{y}{\beta}| \neq 0$, we will have a root $0 < c < 1$. This is because $c = 0$ in (11) gives $-\beta < 0$ and $c = 1$ in (11) gives $|\nabla^S u - \frac{y}{\beta}|^{p-2} > 0$.

If ξ and y are fixed, optimizing with respect to u leads to solving the Poisson boundary value problem given by

$$-\Delta^S u = -\mathrm{div}^S(\xi) - \mathrm{div}^S\left(\frac{y}{\beta}\right) + \frac{1}{\beta} \quad \text{in } \Omega, \tag{12a}$$

$$u = 0 \quad \text{on } \Gamma_1, \tag{12b}$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_2, \tag{12c}$$

which can be done in a standard way using a surface finite element method [1].

ADMM optimizes the augmented Lagrangian with respect to each variable separately. The ADMM iterative procedure for numerically solving the p-Poisson problem is then given by

- ξ-update step: For each $x \in \Omega_h$, solve (11) with $u = u^k$ and $y = y^k$ to find c, then obtain the update $\xi^{k+1} = c(x)\left(\nabla^S u^k - \frac{y^k}{\beta}\right)$.

- u-update step: Solve (12) with $y = y^k$ and $\xi = \xi^{k+1}$ to obtain the update u^{k+1}.

- Dual update: We update the dual variable by $y^{k+1} = y^k + \beta(\xi^{k+1} - \nabla^S u^{k+1})$. This is because the gradient of y may be computed as $\nabla^S y = \xi^* - \nabla^S u^*$ where star superscripts denote optimal values. Here, we can think of β as a step size.

To start the iterative procedure we set $\xi^0 = y^0 = 0$ so u is initialized by solving $-\Delta^S u^0 = \frac{1}{\beta}$ subject to boundary conditions of $u^0|_{\Gamma_1} = 0$ and $\frac{\partial u^0}{\partial n}|_{\Gamma_2} = 0$.

Under certain convexity and existence assumptions given by Gabay and Mercier [32], convergence may be established. Glowinski and Marrocco [33] verify that the p-Poisson problem satisfies necessary assumptions on the plane. Provided these assumptions hold, the ADMM iterates satisfy the following: (i) $r^k = \xi^k - \nabla^S u^k \to 0$ as $k \to \infty$, i.e., the iterates approach feasibility, (ii) the objective function of the iterates approaches the optimal value as $k \to \infty$, and (iii) $y^k \to y^*$ as $k \to \infty$, where y^* is a dual optimal point [31]. Our numerical simulations exhibit convergence of the algorithm.

In addition to tracking the change in the computed solution between iterates, the primal and dual residuals can help inform stopping criteria. The 'primal residual' r^k is given by

$$r^k = \xi^k - \nabla^S u^k = (y^k - y^{k-1})/\beta,$$

and the 'dual residual' s^k is given by

$$s^k = -\beta(\xi^k - \xi^{k-1}).$$

It is often reasonable to use a stopping condition of $||r^k|| < tol_{primal}$ and $||s^k|| < tol_{dual}$ where tol_{primal} and tol_{dual} are chosen by the user. For most test problems, we set $tol_{dual} = 10^{-3}$ and $tol_{primal} = 10^{-6}$. Note that a smaller penalty parameter, β, often yields a better dual residual and a larger β often yields a better primal residual. We wish to strike a healthy balance; typically, we use $\beta = 10$.

4. Computational assessment

This section numerically assesses the ADMM algorithm for the p-Poisson distance-to-feature problem. The algorithm is implemented in C++ using deal.II [24]. Visualizations shown are performed with ParaView [34]. For smooth examples, we can anticipate numerical convergence of order $\mathcal{O}\left(h^2 + \frac{1}{p}\right)$ where h is the maximum cell edge length. We use ℓ to denote the average cell edge length on the mesh. However, distance functions are generally nonsmooth leading us to observe some deterioration in the convergence rates. In particular, distance functions will be nonsmooth near Γ_1, e.g., distance to the origin in 1D gives the absolute value function.

We numerically compute the distance-to-feature for several examples, using three methods: our p-Poisson method (using a low and a large value of p, respectively), the heat method from Crane at al. [2], and the classical polyhedral method from Mitchell et al. [3]. We use the C++ implementations of these methods from geometry-central [35] and export the results to ParaView [34] for consistent visualization. We use the same surface triangulation for all methods.

We emphasize that the p-value can control the smoothness of our p-Poisson distance approximations. This allows us to obtain smoothed distance function representations which may be desired in some practical applications [2]. Additionally, by taking a larger p-value, we recover the nonsmooth nature of the true distance function. For a visualization of small and large p-values shown side-by-side, see Figures 3 and 6.

4.1. Numerical convergence studies

In this subsection we present numerical convergence studies for some examples on simple surfaces where we know the exact distance functions, $\mathrm{dist}(\cdot, \Gamma_1)$, to the feature sets.

The symmetric mean absolute percentage error (SMAPE) [36] is given by

$$\mathrm{SMAPE}(\mathrm{dist}, u_p) = \frac{100}{n} \sum_{i=1}^{n} \frac{|\mathrm{dist}^i - u_p^i|}{\left(|\mathrm{dist}^i| + |u_p^i|\right)/2},$$

where u_p^i is the numerically computed p-Poisson distance at vertex i, and dist^i denotes the exact value of the geodesic distance from the vertex i to the feature set. Here, n denotes the number of vertices.

For a well-rounded error assessment, we also present L^2 integral norm relative errors between the computed p-Poisson distance, u_p, and the exact geodesic distance function, $\mathrm{dist}(\cdot, \Gamma_1)$.

4.1.1. Distance to a point on the hemisphere

The first example we consider is the geodesic distance to a point on the unit hemisphere. In this example, $S = \{(x, y, z) : x^2 + y^2 + z^2 = 1, x > 0\}$ is the unit hemisphere, $\Gamma_1 = \left\{\left(\frac{\sqrt{2}}{2}, \frac{1}{2}, \frac{1}{2}\right)\right\}$, and $\Gamma_2 = \partial S$. The geodesic distance between two points on a sphere is given by the spherical law of cosines. In particular, the geodesic distance between a point $\mathbf{x} = (x, y, z)$ on the hemisphere and Γ_1 is given by

$$\mathrm{dist}\left(\mathbf{x}, \Gamma_1\right) = \arccos\left(\frac{\sqrt{2}}{2}x + \frac{1}{2}y + \frac{1}{2}z\right). \tag{13}$$

The first row of Figure 3 shows the computed distance-to-feature in this example, by the three methods. The order (from left to right) is as follows: the first and second plots correspond to the p-Poisson distance approximation using $p = 5$ and $p = 100$, respectively; the third plot is for the heat method, and the fourth shows the polyhedral distance [3]. For the p-Poisson method with $p = 5$, the contours are less evenly spaced and the Neumann condition on Γ_2 is visually prominent. For $p = 100$, the Neumann condition is visually imperceptible and the contours are evenly spaced. For the heat method we note a severe effect of the boundary on the contours. The polyhedral method seems to give comparable results to the p-Poisson method with $p = 100$.

Figure 4 shows how the primal and dual residuals evolve with the number of ADMM iterations for various p-values. For each $p > 5$, we initialize with the computed solution from the previous (smaller) p-value shown in Figure 4. Since we have already computed the solution for the previous (smaller) p-value, this initialization process reduces the number of iterations needed for the current (larger) p-value. However, this is not necessary to see convergence using the ADMM scheme. Unlike a Newton algorithm, ADMM will converge even with a poor initial guess.

4.1.2. Distance to an open curve and a point on the hemisphere

We now consider an example that seeks the distance to Γ_1 consisting of an open curve and a point on the unit hemisphere. Specifically, $S = \{(x, y, z) : x^2 + y^2 + z^2 = 1, x > 0\}$ is

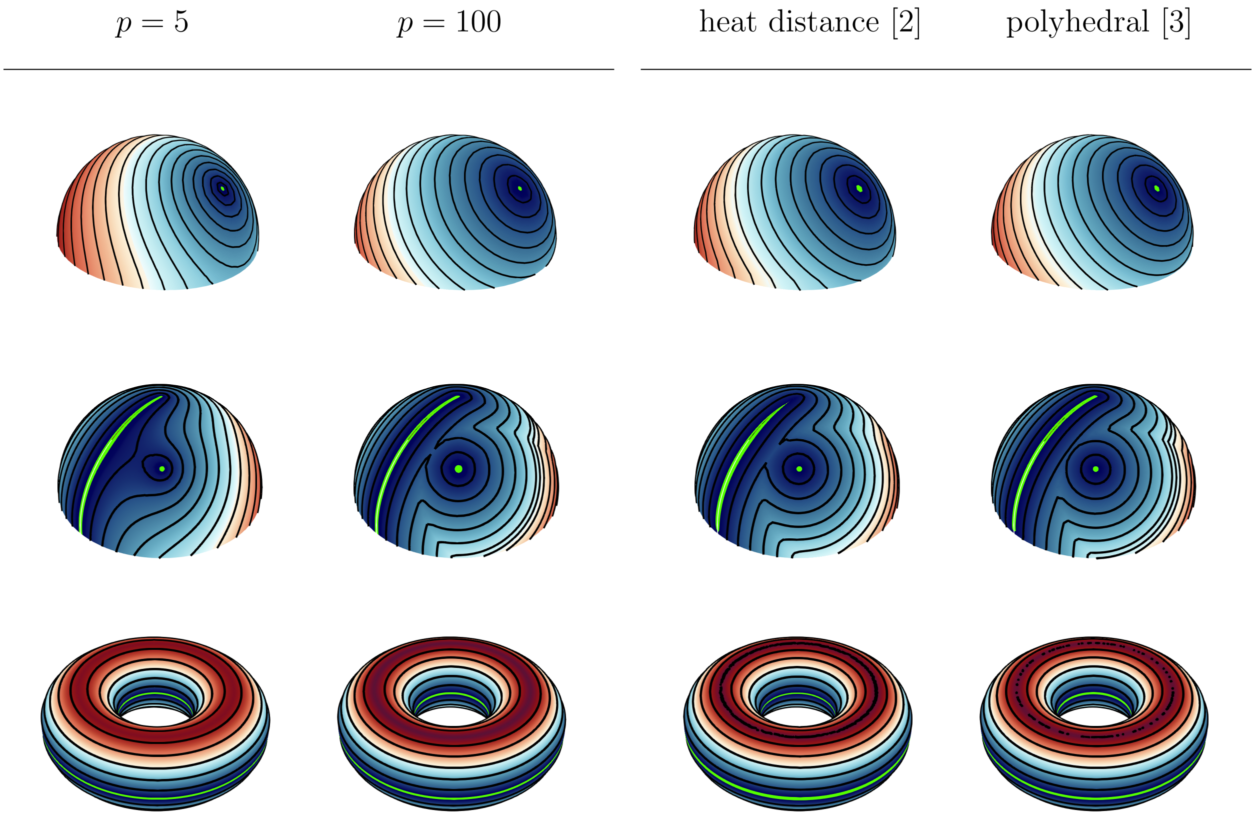


Figure 3: Computed distance approximations using three methods: our p-Poisson method (first and second columns, using $p = 5$ and $p = 100$, respectively), the heat method (third column), and the polyhedral method (fourth column). Top row: distance to a point on hemisphere. Second row: distance to an open curve and a point on hemisphere. Bottom row: distance to two closed curves on torus. For visualization near the feature set Γ_1, we highlight in green values falling below a chosen tolerance. The triangulations consists of 2307 vertices and 791 faces, 9484 vertices and 3228 faces, and 35019 vertices and 105057 faces respectively.

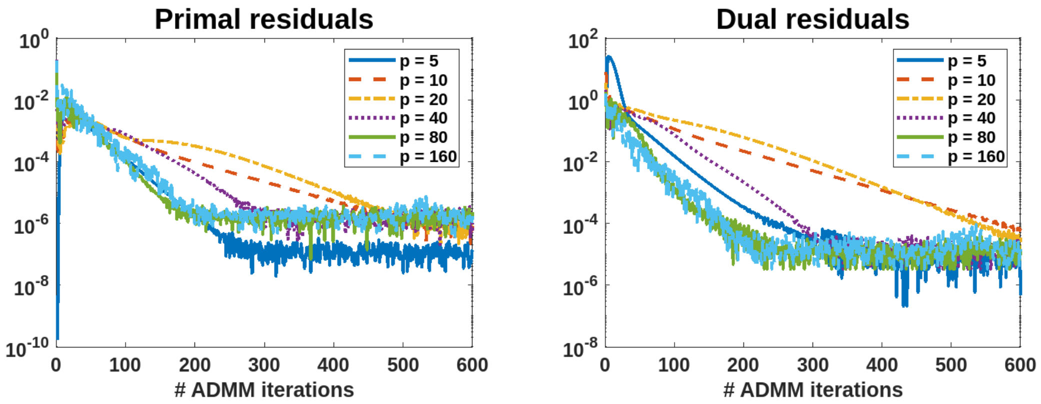


Figure 4: Tracking of residuals and gradient norm for the example of distance to a point on the hemisphere using various p-values. Primal (left) and dual (right) residuals versus ADMM iterations. The algorithm is run on a hemisphere mesh with 10240 tri cells and $\ell \approx 3.95 \times 10^{-2}$, $h \approx 6.29 \times 10^{-2}$.

ADMM algorithm for p-Poisson distance on hemisphere				
p	L^2 (relative error)	Rate	SMAPE (% error)	Rate
5	1.2928×10^{-1}	NA	16.301	NA
10	6.2872×10^{-2}	1.04	8.1910	0.99
20	3.0476×10^{-2}	1.04	3.9934	1.04
40	1.4554×10^{-2}	1.07	1.9136	1.06
80	6.9663×10^{-3}	1.06	9.6856×10^{-1}	0.98
160	3.4894×10^{-3}	1.00	4.8681×10^{-1}	0.99

Table 1: Numerical convergence study in p, comparing computed solutions with exact geodesic distance to an open curve and a point on the hemisphere. The algorithm is run on a hemisphere mesh with 655360 tri cells and $\ell \approx 9.89 \times 10^{-3}$, $h \approx 1.57 \times 10^{-2}$. We note that the SMAPE values are % errors and are thus about 100 times larger than the L^2 relative error values.

the hemisphere,

$$\Gamma_1 = \left\{ \left(\sqrt{2}/2, 1/2, 1/2 \right) \right\} \cup \{(x, y, z) \in S : z = 0, y \geq 0\},$$

and $\Gamma_2 = \partial S \setminus \{(0, 1, 0)\}$ – see the second row in Figure 3.

The second row of Figure 3 shows the numerical computations for this example, using the three methods (the order of the plots is similar to the first row). For the p-Poisson method, we observe that the contours sharpen for larger p-values and the influence of the Neumann condition diminishes. In particular, for $p = 5$, we see smoothing around points that are equidistant to the point and the open curve which comprise Γ_1. The smoothing of the contours is also clearly visible in the heat method. However, for $p = 100$, there is no perceptible smoothing of the distance, and a similar comment applies to the polyhedral method as well.

Table 1 shows L^2 relative errors and SMAPE for our method applied to the problem of computing geodesic distance to an open curve and a point on the hemisphere. Here we use an analytical expression for the geodesic distance, derived from the spherical law of cosines, to compare against our computed solutions. Our primary focus is on the convergence of our method with respect to p. Thus, computations are performed on a large, fixed number of cells, iterating until convergence of the ADMM algorithm. Computations are performed by successively doubling p at each iteration. The results show linear convergence in $1/p$ for both error metrics. It is worth noting that the SMAPE table entries are roughly 100 times larger than the L^2 entries because SMAPE represents a percentage error.

4.1.3. Distance to closed curves on torus

This example seeks the distance to Γ_1 consisting of two closed curves on a torus. Here, $S = \{(x, y, z) : (R - \sqrt{x^2 + z^2})^2 + y^2 = r^2\}$ is the torus with outer radius $R = 2$ and inner radius $r = 1$, Γ_1 consists of the inner and outer circles on the torus lying in the plane $y = 0$, and $\Gamma_2 = \emptyset$ – see the bottom row in Figure 3.

The bottom row of Figure 3 showcases the numerical computations for this example (same three methods and same layout of plots as for the first two rows). Note that since the torus is a closed surface, there is no error from the Neumann boundary condition for

ADMM algorithm for p-Poisson distance on torus				
p	L^2 (relative error)	Rate	SMAPE (% error)	Rate
5	5.8144×10^{-2}	NA	7.0674	NA
10	2.9089×10^{-2}	1.00	3.3470	1.08
20	1.4132×10^{-2}	1.04	1.6215	1.05
40	6.4439×10^{-3}	1.13	7.7920×10^{-1}	1.06
80	2.9304×10^{-3}	1.14	3.7879×10^{-1}	1.04
160	1.5206×10^{-3}	0.95	1.9328×10^{-1}	0.97

Table 2: Numerical convergence study in p, comparing computed solutions with exact geodesic distance to closed curves on a torus. The algorithm is run on a torus mesh with 196608 tri cells and $\ell \approx 1.65 \times 10^{-2}$, $h \approx 2.74 \times 10^{-2}$. We note that the SMAPE values are % errors and are thus about 100 times larger than the L^2 relative error values.

the p-Poisson method. The lower p-value gives a smoother approximation, and the higher value gives a sharper, more accurate approximation. The heat and polyhedral methods give similar results to the p-Poisson method with $p = 100$.

Table 2 shows L^2 relative errors and SMAPE for our method applied to computing the geodesic distance to closed curves defined by $y = 0$ on a torus. An analytical expression for the geodesic distance is used to compute the errors. As in the previous example, computations are performed on a large, fixed number of cells, iterating until convergence of the ADMM algorithm. For each experiment, p is successively doubled. Once again, we find linear convergence in $1/p$ for both error metrics.

We may also monitor how the p-Poisson energy (3) evolves with the ADMM iterations. Figure 5 plots the energy versus the number of iterations for a variety of p-values. As with the residuals, computations in Figure 5 are initialized for each $p > 5$ with the computed solution from the previous (smaller) p-value. We see that the energy decays rapidly within the first few iterations and then remains level at the (supposed) minimizer. Other examples exhibit similar results. If the energy begins to increase, it is often a sign that the p-value is too large for the mesh resolution or that the choice of parameters is unsuitable; for instance, taking the penalty parameter β too small.

Our numerical simulations were performed on a single thread of an AMD Ryzen 7 8845HS (16-thread) Lenovo Yoga Pro 7 14AHP9. For this torus example, an average ADMM iteration took about 0.049 seconds on a mesh with 1536 vertices and 4608 faces, and about 0.185 seconds on a mesh with 6144 vertices and 18432 faces. We observed that the computational timing remains stable as the p-value increases. For successive refinements, the computation time increased by approximately a factor of 4.

Table 3 shows side-by-side statistics for our p-Poisson method, the heat method, and the polyhedral method. For comparison, we fix a large value of $p = 160$. Both L^2 relative errors and SMAPEs are measured against the exact geodesic distance function. For Example 4.1.1, the geodesic distance function is given by the spherical law of cosines (see (13)). Similarly, we have explicit geodesic distance functions for Example 4.1.2 and Example 4.1.3. Table 3 shows that in all three examples, our p-Poisson method yields errors that lie between those of the heat method and the polyhedral method. For Example 4.1.2 and Example 4.1.3, the

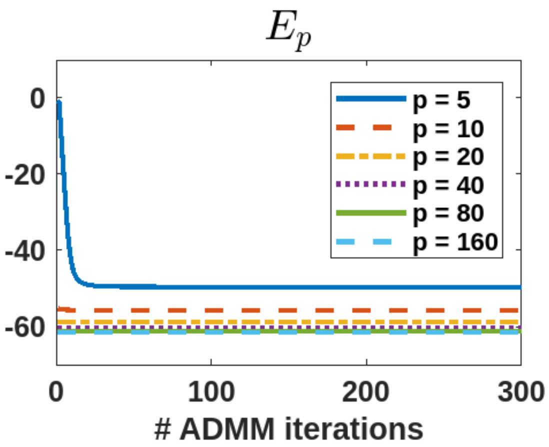


Figure 5: p-Poisson energy (3) versus ADMM iterations for distance to closed curves on torus using various p-values. The algorithm is run on a torus mesh with 196608 tri cells and $\ell \approx 3.30 \times 10^{-2}$, $h \approx 5.49 \times 10^{-2}$.

SMAPE produced by the heat method is comparable to that of the p-Poisson method with $p \in (10, 20)$ and $p \in (40, 80)$ respectively, as shown in Table 1 and Table 2.

4.2. Examples on various surfaces

We now show examples on a wider variety of surfaces. Additionally, we computationally assess robustness to poor mesh quality and check that our distance approximations satisfy the triangle inequality.

Figure 6 shows the computed distance-to-feature Γ_1 on four different surfaces, using the p-Poisson distance approximation with $p = 5$ and $p = 100$ (first and second columns, respectively), the heat method (third column) and the polyhedral method (fourth column). The top two rows show distances to a point on a closed surface (hand and pig). For both of these examples, the $p = 5$ solution overestimates the distance, while the $p = 100$ solution shows a more uniform spacing between contours. These differences are most noticeable on the fingers and legs. The heat method and the polyhedral method for these examples give comparable results to our method with $p = 100$.

The third row in Figure 6 shows the distance to a feature set consisting of two points on an octopus. Here, there is a significant increase in sharpness of the contours going from $p = 5$ to $p = 100$. The heat method gives slightly more smoothing, most prominently around the curve which is equidistant to the two points of the feature set – a similar issue was pointed out for the example in Section 4.1.2 (see the second row of Figure 3). The polyhedral method gives qualitatively similar contours to the p-Poisson method with $p = 100$. Lastly, the bottom row shows the distance to two open curves on a surface with boundaries. We have sliced off one side of the cylinder so this example has a nonempty set Γ_2, as well as sharp edges on the box. Similar to the octopus, we observe a smoother result for $p = 5$. The other three methods ($p = 100$, heat distance, and polyhedral) give similar contours.

4.2.1. Properties of the distance function

We perform some numerical tests to see how well the p-Poisson distance approximations satisfy some fundamental properties of the distance function. Namely, that its gradient has norm 1 and that distances obey the triangle inequality.

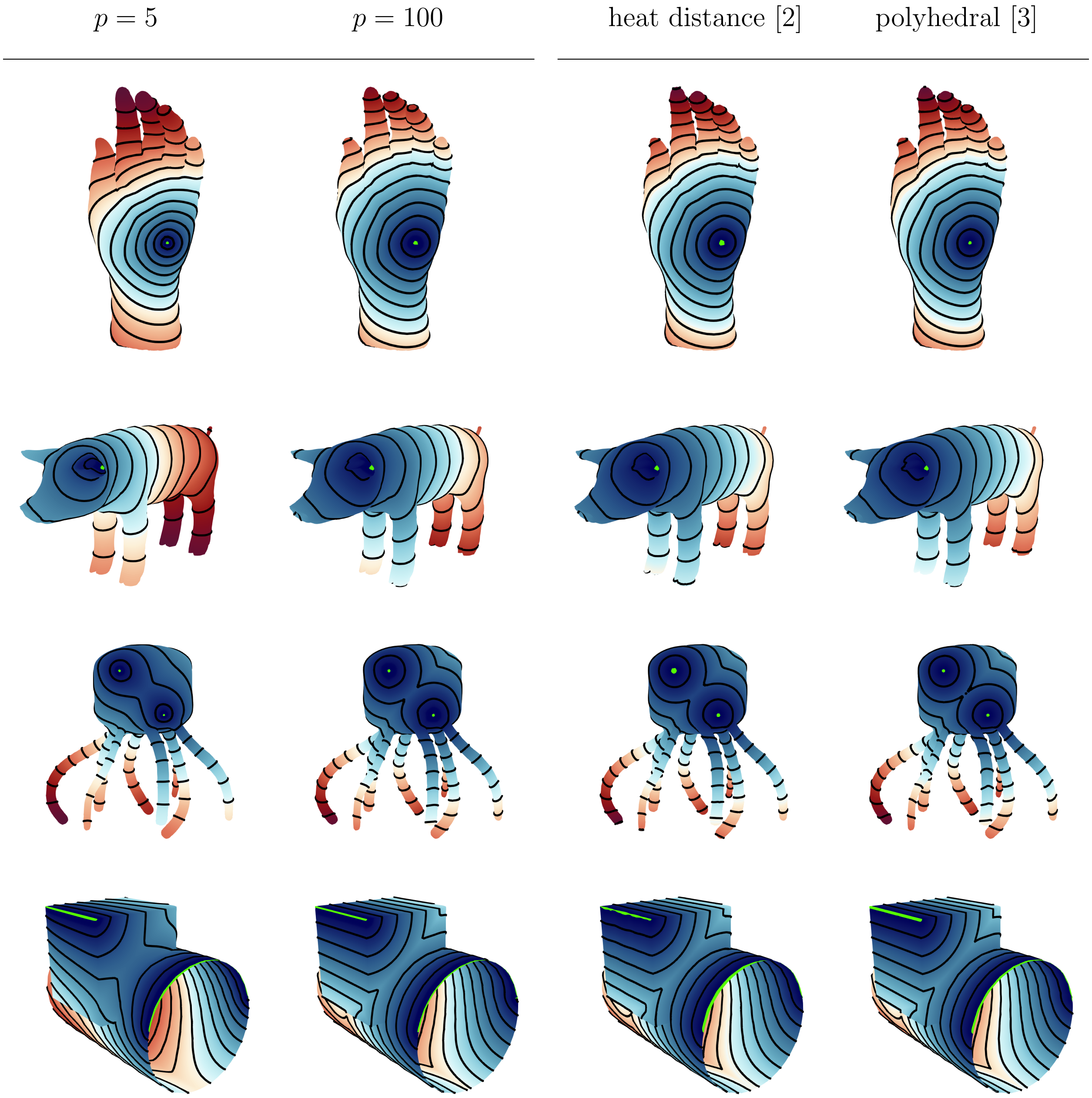


Figure 6: Computed distance approximations using three methods: the p-Poisson method (first and second columns, with $p = 5$ and $p = 100$, respectively), the heat method (third column), and the polyhedral method (fourth column). For visualization near the feature set Γ_1, we highlight in green values falling below a chosen tolerance. Top row: distance to a point on the palm of a hand. Second row: distance to the tip of an ear on the pig. Third row: distance to two distinct points on an octopus. Bottom row: distance to two open curves on a surface with boundaries. Each row uses identical surface triangulations. The hand consists of 8647 vertices and 17290 faces, the pig consists of 8411 vertices and 16818 faces, the octopus consists of 11051 vertices and 22098 faces, and the cylinder with box consists of 81199 vertices and 161648 faces.

<table>
<tr><th colspan="3">Example 4.1.1: Distance to point on the hemisphere</th></tr>
<tr><th>Method</th><th>L^2 (relative error)</th><th>SMAPE (% error)</th></tr>
<tr><td>p-Poisson ($p = 160$)</td><td>4.7582×10^{-3}</td><td>7.0265×10^{-1}</td></tr>
<tr><td>heat distance [2]</td><td>1.0743×10^{-2}</td><td>1.0852</td></tr>
<tr><td>polyhedral [3]</td><td>8.7917×10^{-5}</td><td>2.3249×10^{-2}</td></tr>
<tr><th colspan="3">Example 4.1.2: Distance to an open curve and a point on the hemisphere</th></tr>
<tr><th>Method</th><th>L^2 (relative error)</th><th>SMAPE (% error)</th></tr>
<tr><td>p-Poisson ($p = 160$)</td><td>3.4894×10^{-3}</td><td>4.8681×10^{-1}</td></tr>
<tr><td>heat distance [2]</td><td>2.7041×10^{-2}</td><td>4.8715</td></tr>
<tr><td>polyhedral [3]</td><td>2.0248×10^{-4}</td><td>3.3905×10^{-2}</td></tr>
<tr><th colspan="3">Example 4.1.3: Distance to closed curves on torus</th></tr>
<tr><th>Method</th><th>L^2 (relative error)</th><th>SMAPE (% error)</th></tr>
<tr><td>p-Poisson ($p = 160$)</td><td>1.5206×10^{-3}</td><td>1.9328×10^{-1}</td></tr>
<tr><td>heat distance [2]</td><td>2.4888×10^{-3}</td><td>6.6466×10^{-1}</td></tr>
<tr><td>polyhedral [3]</td><td>6.6534×10^{-4}</td><td>1.2683×10^{-2}</td></tr>
</table>

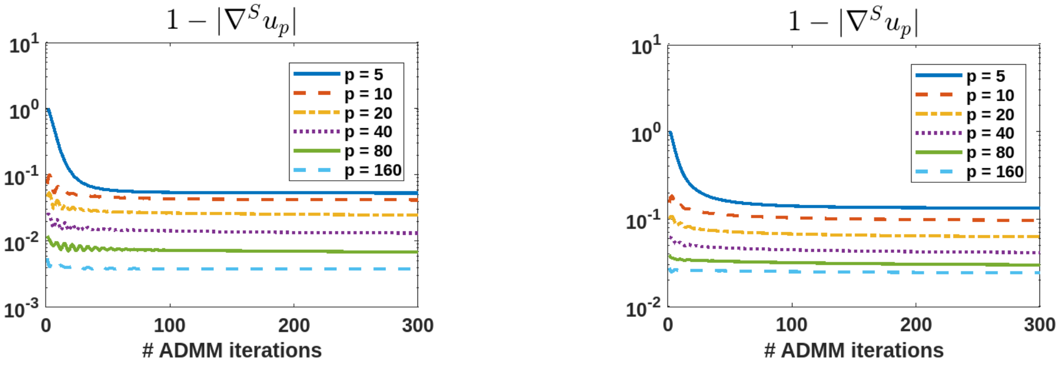
Table 3: Side-by-side method statistics (L^2 relative error and SMAPE) comparing against the exact geodesic distance function; see, for example, (13). The hemisphere mesh for Examples 4.1.1-4.1.2 consists of 655360 tri cells, with $\ell \approx 9.89 \times 10^{-3}$, $h \approx 1.57 \times 10^{-2}$. The torus mesh for Example 4.1.3 consists of 196608 tri cells, with $\ell \approx 1.65 \times 10^{-2}$, $h \approx 2.74 \times 10^{-2}$.

Figure 7 shows the normalized L^1 norm of $1 - \left|\nabla^S u_p\right|$ versus the number of ADMM iterations for various p-values. The examples considered in Figure 7 are the distance-to-point on the hemisphere, as shown in row 1 of Figure 3, and the distance-to-ear example on the pig mesh, as shown in row 2 of Figure 6. The figure shows that the norm of the computed solution gradient is approaching a value of 1. The true distance function gradient has norm 1, so this is an indicator that we are getting closer to the exact distance as p increases. We initialize each of the larger p-values ($p \geq 10$) with the solution from the previous p-value. Therefore, the most rapid decrease in $1 - \left|\nabla^S u_p\right|$ occurs only during the first few iterations for $p = 5$. This behaviour is also observed for other examples.

Figure 8 illustrates how well various numerical methods for computing geodesic distances satisfy the triangle inequality. The figure shows computations of the distance on a bunny, where the feature set consists of a point q_1 on the ear and a point q_2 on the lower body. The left plot was obtained with our method (using $p = 5$), the middle plot used the heat method, and the one on the right corresponds to the polyhedral method. We note that there is a small difference between our computations of $d_p(q_1, q_2)$ and $d_p(q_2, q_1)$, and for this reason we test the triangle inequality using the maximum of the two distances, i.e., we check

$$d_p(q_1, x) + d_p(q_2, x) \geq \max\{d_p(q_1, q_2), d_p(q_2, q_1)\}, \qquad \text{for } x \in S.$$

In Figure 8, the regions that satisfy the triangle inequality above are shown in white, along with the level sets of $\min\{d(q_1, x), d(q_2, x)\}$. Here, $d(\cdot, x)$ denotes the approximate distance to x computed using the method under consideration. The figure shows that throughout the entire mesh, both our method and the polyhedral method satisfy the inequality. We also point out that our method has achieved this result with a low value of p, and the inequality



(a) distance to a point on hemisphere

(b) distance to the ear on pig

Figure 7: Normalized L^1 norm of $1 - \left|\nabla^S u_p\right|$ versus ADMM iterations for various p-values. Results are shown for the distance to a point on hemisphere in Figure 3 (top row) and the distance to the tip of an ear on the pig in Figure 6 (second row). Note the convergence to 1 of $\left|\nabla^S u_p\right|$ as p increases, indicating that the method converges to the exact geodesic distance. The hemisphere mesh consists of 40960 tri cells with $\ell \approx \times 10^{-2}, h \approx \times 10^{-2}$ and the pig mesh consists of 16818 tri cells with $\ell \approx 2.34 \times 10^{-2}, h \approx 4.89 \times 10^{-2}$.

remains satisfied for all higher values of p that we tested. In contrast, distances computed using the heat method fail to satisfy the triangle inequality over significant regions on the bunny, shown in red. This shortcoming of the heat method was already noted in [12] (see Figure 9 in their paper for another visualization of these regions).

4.2.2. Robustness against mesh quality

It is desirable for a method to withstand some topological perturbations such as vertex noise, artificial holes, and nearly isometric deformation.

Crane et al. [2] and Solomon et al. [12] both demonstrate that their methods for distance approximation can withstand some per-vertex noise. Figure 9 shows computed p-Poisson distances to the lid of a teapot with varying degrees of Gaussian noise added to the mesh. More precisely, we perturb the vertices by adding Gaussian noise with standard deviations $\sigma = \frac{\ell}{8}, \frac{\ell}{4}, \frac{\ell}{2}$ where $\ell \approx 1.60 \times 10^{-2}$ is the average cell edge length on the mesh. We observe that the contours undergo minimal change with the addition of noise. Visualizations are shown only for $p = 100$ although other p-values have similar results.

5. Conclusions

We have extended the p-Poisson distance problem to handle the intrinsic distance-to-feature problem. Our approach has the advantage that errors are controlled near the surface boundary without the need to average multiple computations, as is required with Crane et al.'s [2] method. The ADMM scheme for solving the boundary value problem demonstrates numerical convergence to the exact geodesic distance as $p \to \infty$. Additionally, ADMM decouples the ill-conditioning effects of large p-values, which means the starting p-value may be large. Our computed p-Poisson distances exhibit many desirable properties, such as robustness against geometric noise and satisfaction of the triangle inequality. Our workflows are centered on the deal.II implementation for triangular and quadrilateral meshes, leaving room for future work exploring p-Poisson distances on various discrete surface representations, as well as numerical approximation methods. The numerical methods presented are

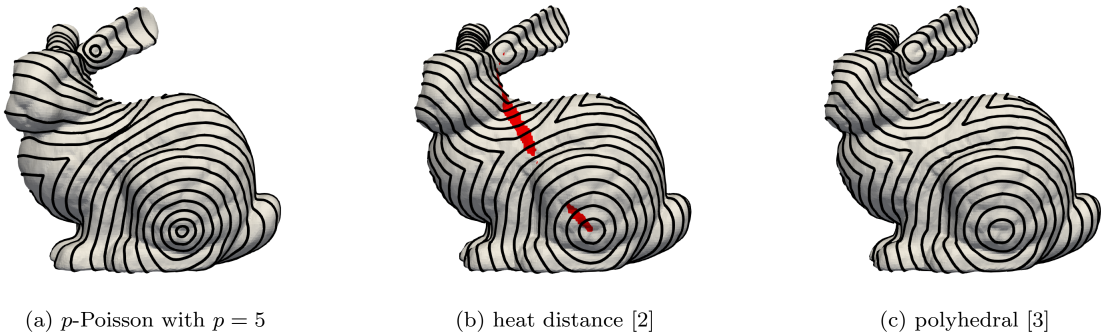
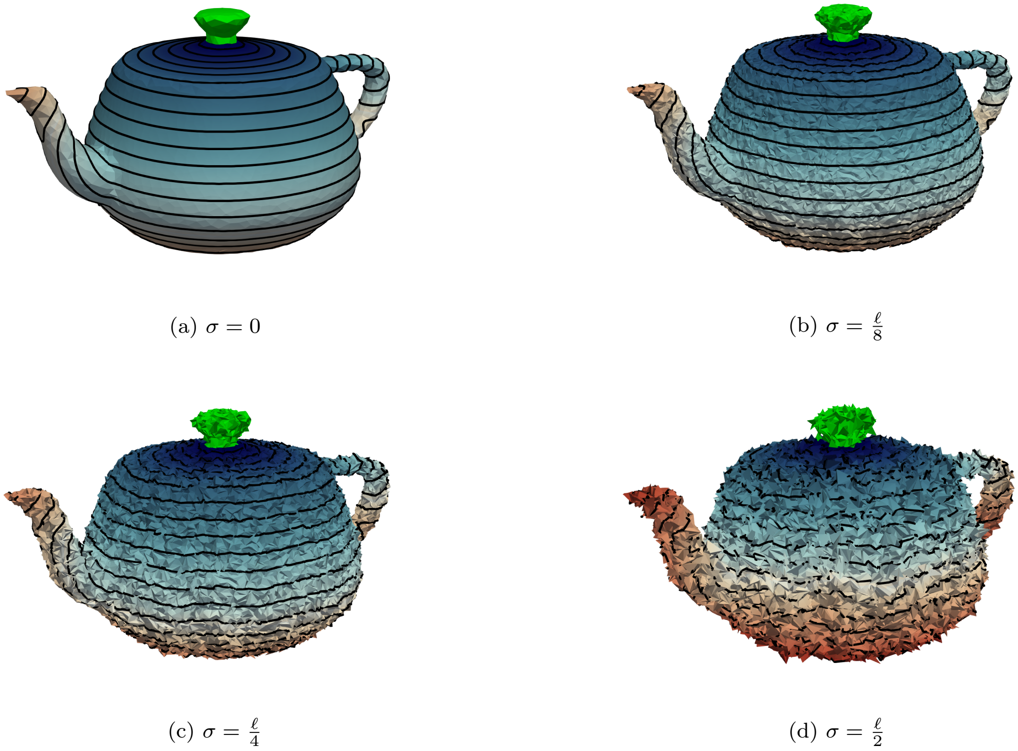

(a) p-Poisson with $p = 5$ (b) heat distance [2] (c) polyhedral [3]

Figure 8: Triangle inequality check. The feature set consists of a point q_1 on the ear on the bunny, and a point q_2 on its lower body. The computations were done with three methods: our p-Poisson distance approximation (left), the heat method (center), and the polyhedral method (right). Regions which satisfy the triangle inequality are shown in white, along with the level sets of $\min\{d(q_1, x), d(q_2, x)\}$, where $d(\cdot, x)$ denotes the distance approximation obtained by the method of consideration. Regions which violate the triangle inequality are shown in red. Note that the entire bunny is white for the p-Poisson distance approximation and the polyhedral method, while the heat method fails to satisfy the triangle inequality on fairly large regions. The bunny triangulation consists of 3900 vertices and 7800 faces.


(a) $\sigma = 0$ (b) $\sigma = \frac{\ell}{8}$

(c) $\sigma = \frac{\ell}{4}$ (d) $\sigma = \frac{\ell}{2}$

Figure 9: Computed p-Poisson distances to the lid of a teapot with Gaussian noise added to the mesh vertices. The standard deviation is in terms of the average mesh cell edge length $\ell \approx 1.60 \times 10^{-2}$. Top left: no noise. Top right: minimal amount of noise ($\sigma = \frac{\ell}{8}$). Bottom left: medium amount of noise ($\sigma = \frac{\ell}{4}$) . Bottom right: large amount of noise ($\sigma = \frac{\ell}{2}$). The teapot quad mesh consists of 3900 vertices and 7800 faces.

also adaptable to different problems involving Δ_p^S, so this work provides a starting point for the study of other applications.

Acknowledgements

The authors gratefully acknowledge the financial support of NSERC Canada (RGPIN 341834 for RF and RGPIN 03302 for SR).

The bunny mesh is provided courtesy of the Stanford Computer Graphics Laboratory. The remaining meshes are obtained from the Chen et al. [37] benchmark set for segmentation.

References

[1] P.-A. Fayolle, A. G. Belyaev, p-Laplace diffusion for distance function estimation, optimal transport approximation, and image enhancement, Computer Aided Geometric Design 67 (2018) 1–20.

[2] K. Crane, C. Weischedel, M. Wardetzky, The heat method for distance computation, Commun. Association for Computing Machinery 60 (11) (2017) 90–99.

[3] J. S. B. Mitchell, D. M. Mount, C. H. Papadimitriou, The discrete geodesic problem, Society for Industrial and Applied Mathematics J. Comput. 16 (1987) 647–668.

[4] T. Bühler, M. Hein, Spectral clustering based on the graph p-Laplacian, in: Proceedings of the 26th Annual International Conference on Machine Learning, Association for Computing Machinery, 2009, p. 81–88.

[5] P. Blomgren, T. Chan, P. Mulet, C. Wong, Total variation image restoration: Numerical methods and extensions, in: Institute of Electrical and Electronics Engineers International Conference on Image Processing, Vol. 3, 1997, pp. 384 – 387.

[6] Y. Chen, S. Levine, M. Rao, Variable exponent, linear growth functionals in image restoration, Society for Industrial and Applied Mathematics Journal on Applied Mathematics 66 (4) (2006) 1383–1406.

[7] V. Caselles, L. Igual, O. Sander, An axiomatic approach to scalar data interpolation on surfaces, Numerische Mathematik 102 (2006) 383–411.

[8] M. Ruzicka, Electrorheological fluids: Modeling and mathematical theory, Lecture Notes in. Mathematics, Springer Berlin Heidelberg 1748 (2000).

[9] G. Cong, M. Esser, B. Parvin, G. Bebis, Shape metamorphism using p-Laplacian equation, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004., Vol. 4, 2004, pp. 15–18.

[10] T. Bhattacharya, E. DiBenedetto, J. Manfredi, Limits as $p \to \infty$ of $\Delta_p u = f$ and related extremal problems, Rend. Sem. Mat. Univ. Politec. Torino 47 (1989) 15–68.

[11] Y. Lipman, R. M. Rustamov, T. A. Funkhouser, Biharmonic distance, Association for Computing Machinery Trans. Graph. 29 (3) (2010).

[12] J. Solomon, R. Rustamov, L. Guibas, A. Butscher, Earth mover's distances on discrete surfaces, Association for Computing Machinery Trans. Graph. 33 (4) (2014).

[13] A. Naber, D. Berwanger, W. Nahm, Geodesic length measurement in medical images: Effect of the discretization by the camera chip and quantitative assessment of error reduction methods, Photonics 7 (3) (2020).

[14] A. G. Belyaev, P.-A. Fayolle, On variational and PDE-based distance function approximations, Computer Graphics Forum 34 (8) (2015) 104–118.

[15] X. Wang, Z. Fang, J. Wu, S.-Q. Xin, Y. He, Discrete geodesic graph (DGG) for computing geodesic distances on polyhedral surfaces, Computer Aided Geometric Design 52-53 (2017) 262–284.

[16] R. Kimmel, J. A. Sethian, Computing geodesic paths on manifolds, Proceedings of the National Academy of Sciences 95 (15) (1998) 8431–8435.

[17] S. R. S. Varadhan, Diffusion processes in a small time interval, Communications on Pure and Applied Mathematics 20 (4) (1967) 659–685.

[18] N. Feng, K. Crane, A heat method for generalized signed distance, ACM Trans. Graph. 43 (4) (jul 2024).

[19] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1) (1959) 269–271.

[20] K. Crane, M. Livesu, E. Puppo, Y. Qin, A survey of algorithms for geodesic paths and distances, arXiv preprint arXiv:2007.10430 (2020).

[21] J. A. Sethian, A fast marching level set method for monotonically advancing fronts., Proceedings of the National Academy of Sciences 93 (4) (1996) 1591–1595.

[22] M. Edelstein, N. Guillen, J. Solomon, M. Ben-Chen, A convex optimization framework for regularized geodesic distances, in: Association for Computing Machinery SIGGRAPH 2023 Conference Proceedings, SIGGRAPH '23, Association for Computing Machinery, 2023.

[23] Y. Y. Adikusuma, Z. Fang, Y. He, Fast construction of discrete geodesic graphs, ACM Trans. Graph. 39 (2) (Mar. 2020).

[24] D. Arndt, W. Bangerth, M. Feder, M. Fehling, R. Gassmöller, T. Heister, L. Heltai, M. Kronbichler, M. Maier, P. Munch, J.-P. Pelteret, S. Sticko, B. Turcksin, D. Wells, The `deal.II` library, version 9.4, Journal of Numerical Mathematics 30 (3) (2022) 231–246.

[25] O. Stein, E. Grinspun, M. Wardetzky, A. Jacobson, Natural boundary conditions for smoothing in geometry processing, ACM Transactions on Graphics 37 (2), article 23 (2018).

[26] J. Garcia-Azorero, J. J. Manfredi, I. Peral, J. D. Rossi, The limit as $p \to \infty$ for the p-Laplacian with mixed boundary conditions and the mass transport problem through a given window, Rend. Lincei Mat. Appl. 20 (2009) 111–126.

[27] Y. Q. Huang, R. Li, W. Liu, Preconditioned descent algorithms for p-Laplacian, Journal of Scientific Computing 32 (2007) 343–371.

[28] T. Aubin, Nonlinear Analysis on Manifolds. Monge-Ampère Equations, 1st Edition, Vol. 252 of Grundlehren der mathematischen Wissenschaften, Springer New York, NY, 1982.

[29] G. Dziuk, C. M. Elliott, Finite element methods for surface PDEs, Acta Numerica 22 (2013) 289–396.

[30] S. J. Ruuth, B. Merriman, A simple embedding method for solving partial differential equations on surfaces, Journal of Computational Physics 227 (3) (2008) 1943–1961.

[31] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, Foundations and Trends in Machine Learning 3 (2011) 1–122.

[32] D. Gabay, B. Mercier, A dual algorithm for the solution of nonlinear variational problems via finite element approximation, Computers & Mathematics with Applications 2 (1) (1976) 17–40.

[33] R. Glowinski, A. Marrocco, On the Solution of a Class of Non-Linear Dirichlet Problems by a Penalty-Duality Method and Finite Elements of Order One, Springer Berlin Heidelberg, 1975, pp. 327–333.

[34] U. Ayachit, The ParaView Guide: A Parallel Visualization Application, Kitware, Inc., 2015.

[35] N. Sharp, K. Crane, et al., Geometrycentral: A modern c++ library of data structures and algorithms for geometry processing (2019).

[36] T. Nguyen, T. Nguyen, B. M. Nguyen, G. Nguyen, Efficient time-series forecasting using neural network and opposition-based coral reefs optimization, International Journal of Computational Intelligence Systems 12 (2019) 1144–1161.

[37] X. Chen, A. Golovinskiy, T. Funkhouser, A benchmark for 3D mesh segmentation, Association for Computing Machinery Transactions on Graphics (Proc. SIGGRAPH) 28 (3) (2009).