

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 rates analogous to those observed on the plane. We check that our numerical results adhere to the triangle inequality and examine robustness against geometric noise such as artificial holes, vertex perturbations, and isometric deformation.

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 [2]), image processing (Blomgren et al. [3]; Chen et al. [4], Caselles et al. [5]), distance function approximation (Fayolle & Belyaev [1]), optimal transport (Fayolle & Belyaev [1]), non-Newtonian fluids (Ružička [6]), and shape morphing (Cong et al. [7]). In this paper, we focus on the application of the p -Laplacian for computing geodesic distances on surfaces.

Bhattacharya et al. [8] consider the following boundary value problem (BVP) for the p -Laplacian in a bounded domain $\Omega \subset \mathbb{R}^n$ with boundary $\partial\Omega$:

$$\Delta_p u_p = -1, \quad \text{in } \Omega, \quad (1a)$$

$$u_p = 0, \quad \text{on } \partial\Omega \quad (1b)$$

for $2 \leq p \leq \infty$. The p -Laplacian is defined by $\Delta_p u = \operatorname{div}(|\nabla u|^{p-2} \nabla u)$; note that for $p = 2$ it reduces to the usual Laplacian. Bhattacharya et al. [8] show that the solution $u_p(x)$ to the BVP (1a)-(1b) converges strongly as $p \rightarrow \infty$, 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. We note that Fayolle & Belyaev [1] only use the extrinsic p -Laplace operator for distance approximation, however they use a surface formulation for an optimal transport application which uses a slightly different BVP.

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 [9], [10], and [1]. It is also useful for medical image analysis (see Naber et al. [11]). Belyaev & Fayolle [12] applied ADMM schemes to different energy minimization problems seeking to achieve accurate distance function approximations, and showed in [1] that minimizing 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

One of the most popular PDE-based geodesic distance approximation methods is Crane et al.’s [13] heat kernel method based on Varadhan’s formula [14]. Crane et al.’s [13] method splits the problem into first finding the direction along which distance is increasing and then computing the distance itself. Crane et al.’s [13] method is computationally efficient, adaptable, robust against noisy surface data, and simple. However, it has many limitations some of which are experimental boundary conditions, producing only smooth approximations, poor approximations far from source, and often failing to be symmetric and to satisfy the triangle inequality. The boundary conditions on open surfaces, such as the hemisphere, are experimental in the sense that neither homogeneous Neumann nor Dirichlet conditions produce accurate approximations. To get around this, Neumann and Dirichlet results are both computed and then averaged. This proposed average does not have any theoretical justification.

Solomon et al. [10] propose an optimal transport inspired method for computing earth mover’s distance between probability distributions on surfaces and can be adapted for the *distance-to-feature* problem. Although computationally efficient, convergence studies are lacking for Solomon et al. [10] and the setup is complex. Recently in this group, Edelstein et al. [15] proposed a general PDE-based convex optimization problem for computing regularized geodesic distances which can handle surfaces with boundary. This framework demonstrates robustness to geometric perturbations.

Solutions to the Eikonal equation, $|\nabla u(x)| = 1$, may also be numerically approximated directly to obtain geodesic distance approximations. For example, Sethian [16] proposed the fast marching method for numerical approximation of solutions to the Eikonal equation [17]. Parallelizing the fast marching method requires significant efforts which are addressed in various ways as done by Weber et al. [18] and others. Solving the Eikonal equation directly also presents difficulties for nonsmooth problems and requires monotonicity or nonobtuse triangulations for simplicial grids [12].

Most PDE methods work for a wide range of discrete domains, including regular grids, point clouds, and unstructured triangular and tetrahedral meshes [17]. Additionally, Wang et al. [17] point out that PDE methods may be implemented much more efficiently than the computationally expensive “exact” methods. “Exact” methods work directly on the specific mesh structure to search for the shortest intrinsic distance. This is done by several authors, including Mitchell et al. [19], Surazhsky et al. [20], and Kimmel & Sethian [21]. It is pointed out in [9] and [10] that these “exact” geodesics may be unsuitable for computational applications since they are sensitive to noise and topological perturbations (such as nearly isometric deformation or artificial holes).

1.2. Contributions

There are many existing works in the literature which study the problem of distance approximation on surfaces. See, for example, [13], [12], [17], [9], [10], and [21]. However, we offer some distinct results. Our hope is that p -Poisson geodesic distance approximations will be advantageous over other methods in certain cases and help to provide a more well-rounded assessment when computed alongside other approximations. Further, p -Poisson problems arise in optimal transport for the Monge transportation problem (known as the earth mover’s distance problem) [1]. This means the computational assessment we perform may be informative for optimal transport applications. Our primary contributions are threefold.

First, we provide numerical convergence studies which demonstrate that the boundary conditions we impose on natural surface boundaries are appropriate. Additionally, we provide analytical justification for the 1D problem in Section 2.1. Since the exact geodesic distance is not known at the boundary, inexact constraints must be imposed. This comes into play when the feature to which we want the distance is not simply the surface boundary. Other methods such as Crane et al’s [13] heat kernel approach do not handle boundary conditions in a reliable manner. Works such as Solomon et al. [10] implement boundary conditions which seem suitable visually but the error introduced from the boundary conditions is not quantified.

Second, this work helps to fill the literature gap on numerical convergence studies for p -Poisson distances. Fayolle & Belyaev [1] show encouraging computational results for the p -Poisson *distance-to-surface* problem but do not provide comprehensive numerical convergence studies. Furthermore, our problem is posed on surfaces. To our knowledge, numerical convergence studies are not performed in the literature for finite element approximation of p -Poisson problems on surfaces. Numerical convergence studies for finite element approximation of p -Laplace type problems in Euclidean space are performed by Pryer [22] and Barret & Liu [23] among others.

Third, we perform robustness checks and assess stability against geometric perturbations. In particular, we check how well our numerical results satisfy metric properties, such as the triangle inequality, and examine sensitivity to geometric noise such as artificial holes, vertex perturbations, and isometric deformation. Fayolle & Belyaev [1] do not perform these robustness tests for the *distance-to-surface* problem. These are common tests in the literature, as seen, for example, in [9], [13], and [10].

2. Problem formulation

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

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

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:

$$\Delta_p^S u_p = -1, \quad \text{in } \Omega, \quad (2a)$$

$$u_p = 0, \quad \text{on } \Gamma_1 \subseteq \partial\Omega, \quad (2b)$$

$$\frac{\partial u_p}{\partial n} = 0, \quad \text{on } \Gamma_2 \subseteq \partial\Omega. \quad (2c)$$

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 natural surface boundary. In the spirit of other distance approximation works such as [15], [13], and [10], we explore this use of Neumann conditions and find that they perform well. Section 2.1 goes into more detail on the homogeneous Neumann condition for a particular example.

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

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

subject to proper boundary conditions. Setting $0 < p < 1$ would lead to a non-convex energy, however, we are interested in $p \in [2, \infty)$. The Euler-Lagrange equations give rise to equation (2a).

We will demonstrate by numerical simulations that

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

Hence, large p values are of interest. Unfortunately, large p values lead to ill-conditioned problems and are therefore numerically challenging to solve.

We point out that in Bhattacharya et al. [8] and in the computational work on distance approximation of Fayolle & Belyaev [1], the problem considered is (1a)-(1b), which is different and also, very restrictive. Problem (1a)-(1b) computes (in the limit $p \rightarrow \infty$) the distance in the Euclidean space to the boundary of a set $\Omega \subset \mathbb{R}^n$. In (2a)-(2c) we set up a BVP on a subset Ω of a surface S using the surface p -Laplacian, and 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 . If $\Gamma_1 = \partial S$ (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. [8], except that here we are interested in geodesic (rather than Euclidean) distances.

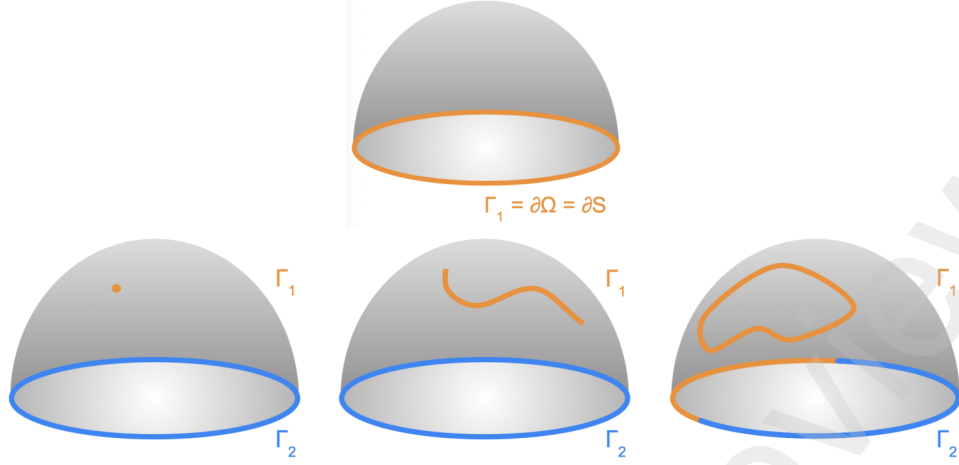


Figure 1: S is the open hemisphere and Γ_1 (indicated in orange) represents the feature set. Top row: Distance to boundary problem ($\Gamma_1 = \partial S, \Gamma_2 = \emptyset$). Bottom row: *Distance-to-feature* problem. Γ_2 is indicated in blue.

2.1. 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

$$(|u'|^{p-2} u')' = -1, \quad 0 < |x| < 1, \quad (4a)$$

$$u_p(0) = 0, \quad (4b)$$

$$u_p'(-1) = u_p'(1) = 0. \quad (4c)$$

The exact solution of (4a)-(4c) is

$$u_p(x) = -\frac{p-1}{p} (1 - |x|)^{\frac{p}{p-1}} + \frac{p-1}{p}. \quad (5)$$

Using (5) one can then show immediately that for any $x \in \Omega$ fixed,

$$\lim_{p \rightarrow \infty} u_p(x) = |x|, \quad (6)$$

which is the distance to the origin. We note that the convergence (6) is in fact uniform on Ω . Indeed, use

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

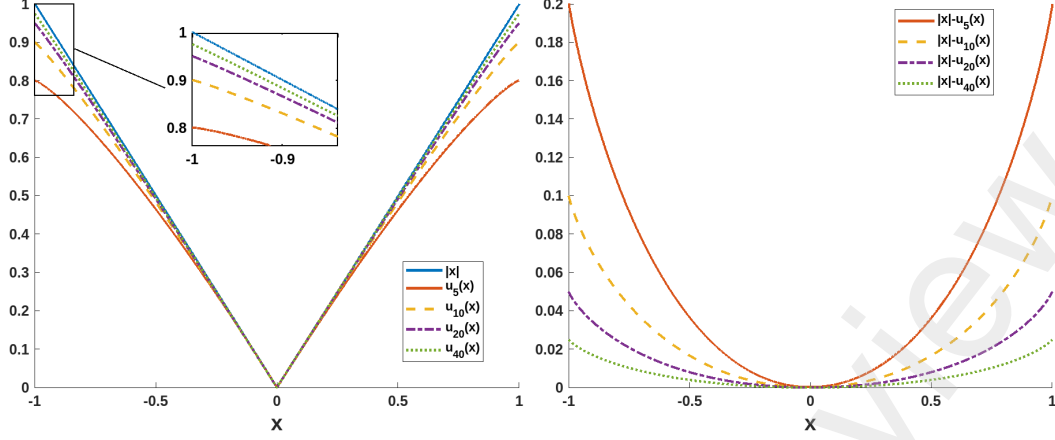


Figure 2: One-dimensional example, with $S = (-1, 1)$ and $\Gamma_1 = \{0\}$. Left: the exact solution $u_p(x)$ (see (4a)-(4c) and (5)) converges to $|x| = \text{dist}(x, \Gamma_1)$ as p increases. Right: the corresponding pointwise errors $|x| - u_p(x)$, for various p values.

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,

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

The uniform convergence in (6) can then be concluded.

Figure 2 shows the exact solution (5) 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 $p \rightarrow \infty$ limit, $u_p(x)$ violates the Neumann condition (4c), which is the desired behaviour for the distance problem. Accordingly, in computations with large p the error from the homogeneous Neumann condition (4c) 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

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

where \mathcal{T}_h is 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 .

For the surface Poisson problem, $\Delta^S u = f$, Dziuk & Elliot [24] 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 a surface finite element method due to the nonsmooth nature of distance functions. Using a finite difference discretization paired with the closest point embedding method [25] suffices for smooth problems. However, finite difference discretizations fail to converge for the examples we consider. Working with the p -Laplacian allows us to employ a variational finite element method which is more robust against nonsmooth data.

3.1. Newton scheme

We may use a Newton iteration in order to solve the nonlinear p -Poisson problem. Our implementation is very similar to those shown in [26], [27], [28] and [29] but with different boundary conditions and posed on a surface.

Newton's method seeks solutions of $F(u(x)) = 0$. We solve for the update via

$$F'(u^n)[\delta u^n] = -F(u^n), \quad (7a)$$

$$u^{n+1} = u^n + \alpha^n \delta u^n, \quad (7b)$$

where $0 < \alpha_n \leq 1$ and $F'(u)[\delta u]$ is the derivative of F in direction of δu . The damping parameter, $0 < \alpha_n \leq 1$, is chosen at each iteration via a quadratic interpolation line search with a sufficient decrease condition as done by Alvarez & Flores [26]. That is, we find $0 < \alpha_n \leq 1$ such that

$$E_p(u^n + \alpha_n \delta u^n) \leq E_p(u^n) + c_1 \alpha_n E_p'(u^n)[\delta u^n],$$

where c_1 is a prescribed tolerance, E_p is given by (3), and $E_p'(u)[\delta u]$ is the derivative of E_p in the direction of δu .

For our p -Poisson problem, (7) is used with

$$F(u) = \nabla^S \cdot (\gamma(u) \nabla^S u) + 1, \quad \gamma(u) = (\eta^2 + |\nabla^S u|^2)^{\frac{p-2}{2}}.$$

Here, η is a small regularization constant which ensures positive-definiteness of the matrix in the resulting linearized problem. This ensures a CG solver will be suitable. This gives the Newton iteration equations as

$$\nabla^S \cdot \left(\gamma(u^n) \nabla^S \delta u^n \right) + (p-2) \nabla^S \cdot \left(\gamma(u^n) \frac{\nabla^S u^n \cdot \nabla^S \delta u^n}{\eta^2 + |\nabla^S u^n|^2} \nabla^S u^n \right) \quad (8a)$$

$$= -\nabla^S \cdot \left(\gamma(u^n) \nabla^S u \right) - 1,$$

$$u^{n+1} = u^n + \alpha^n \delta u^n. \quad (8b)$$

The weak formulation of (8a) is obtained by multiplying with a test function ϕ and integrating by parts on both sides. Reducing to a finite dimensional space with basis $\{\phi_h^0, \dots, \phi_h^{N-1}\} \in S_h$, we can write the update $\delta u^n \in H^1(\Omega)$, as

$$\delta u^n = \sum_{j=0}^{N-1} \delta U_j^n \phi_h^j.$$

The resulting linear system for the update coefficients δU^n at each iteration is then

$$A^n \delta U^n = b^n, \quad (9)$$

with the entries of the matrix A^n given by

$$A_{ij}^n := \int_{\Omega_h} \left(\nabla^S \phi_h^i \cdot (\gamma(u^n) \nabla^S \phi_h^j) + (p-2) (\nabla^S u^n \cdot \nabla^S \phi_h^i) \left(\frac{\gamma(u^n)}{\eta^2 + |\nabla^S u^n|^2} \nabla^S u^n \cdot \nabla^S \phi_h^j \right) \right) dA_h,$$

and the right hand side b^n given by

$$b_i^n := \int_{\Omega_h} (-\nabla^S \phi_h^i \cdot (\gamma(u^n) \nabla^S u^n) + \phi_h^i) dA_h.$$

The linear system (9) quickly becomes ill-conditioned when $\nabla^S u^n$ becomes large and when p is large. In addition to applying preconditioning to a CG solver, we need an initial guess very close to the exact solution in order to see convergence. To get around this, we initially run the algorithm for a low value of p , and then increase p by δp using the previous solution as an initial guess until reaching the target p value. We initialize for the starting p value of p_0 by taking $u_{p_0}^0 = 0$. Then, we follow the steps given below

- Solve numerically for $u_{p_i}^{N_i}$ using Newton algorithm such that

$$\frac{\|u_{p_i}^{N_i} - u_{p_i}^{N_i-1}\|_2}{\|u_{p_i}^{N_i}\|_2} \leq \text{tol}.$$

- Increment the p value via $p_{i+1} = p_i + (\delta p)_{i+1}$.
- Initialize for the new p value with $u_{p_{i+1}}^0 = u_{p_i}^{N_i}$.
- Repeat previous 3 steps until reaching the target p value.

Typically, we take $2 \leq p_0 \leq 5$, $1 \leq (\delta p)_i \leq 10$, and $10^{-4} \leq \text{tol} \leq 10^{-10}$. There are many possible ways to select a suitable target p value. Most often, we will continue incrementing in p until either

$$\frac{\|u_{p_{i+1}}^{N_{i+1}} - u_{p_i}^{N_i}\|_2}{\|u_{p_{i+1}}^{N_{i+1}}\|_2} \leq \text{tol}_p.$$

for a prescribed tol_p or we are unable to solve the system due to the ill-conditioning. Typically we can take $p \sim 300$ before the latter occurs.

3.2. ADMM scheme

An alternative to a Newton scheme is to use an ADMM scheme. ADMM is a competitive tool in solving problems arising in data science due to its efficiency and computational advantages over other similar methods [30]. 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. Boyd et al. [30] state that ADMM is best for problems where efficiency is more important than very high accuracy. Our problem of distance function approximation falls broadly into geometry processing where we care mostly about the visual results and do not require very high accuracy. Therefore, it makes sense to employ ADMM for the variational optimization problem. Additionally, we observe that the p -Poisson ADMM algorithm can achieve the same level of accuracy as the Newton algorithm.

We now present an ADMM algorithm for a p -Poisson problem. As discussed in the prior subsection, 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 $\frac{\partial u}{\partial n}|_{\Gamma_2} = 0$ incorporated into the finite element discretization. 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.

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

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

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 ξ_{\parallel} and ξ_{\perp} so that $\xi = \xi_{\parallel} + \xi_{\perp}$. Plugging this into (10) gives

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

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

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

or equivalently solving

$$\left| \nabla^S u - \frac{y}{\beta} \right|^{p-2} c^{p-1} + \beta(c-1) = 0 \quad (12)$$

for $c \in [1]$. The latter is obtained by differentiating (11) 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 (12) gives $-\beta < 0$ and $c = 1$ in (12) 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

$$-\Delta^S u = -\operatorname{div}^S(\xi) - \operatorname{div}^S \left(\frac{y}{\beta} \right) + \frac{1}{\beta} \quad \text{in } \Omega, \quad (13a)$$

$$u = 0 \quad \text{on } \Gamma_1, \quad (13b)$$

$$\frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_2 \quad (13c)$$

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 (12) 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 (13) 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, we can make convergence statements. Details of these assumptions are given by Gabay and Mercier [31]. Glowinski and Marrocco [32] verify that the p -Poisson problem satisfies necessary assumptions on the plane. Provided these assumptions hold, the ADMM iterates satisfy the following: (i) $r^k = \xi^k - \nabla^S u^k \rightarrow 0$ as $k \rightarrow \infty$, i.e., the iterates approach feasibility, (ii) the objective function of the iterates approaches the optimal value as $k \rightarrow \infty$, and (iii) $y^k \rightarrow y^*$ as $k \rightarrow \infty$, where y^* is a dual optimal point [30]. It is important to note that although we have convergence, the rates may be slow.

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, setting both between 10^{-4} and 10^{-6} typically gives results similar to the Newton algorithm. 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 Newton and ADMM algorithms for the p -Poisson *distance-to-feature* problem. Both the Newton and ADMM algorithms are implemented in C++ using deal.ii [33]. This means surface meshes must be quadrangulations. Although computing on surface quadrangulations is required for deal.ii [33], the methods are applicable to surface triangulations if using a compatible finite element software. 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. 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.

# cells	p	L^2 error	Rate	H^1 error	Rate	L^∞ error	Rate
80	25	1.658×10^{-1}	<i>NA</i>	5.020×10^{-1}	<i>NA</i>	3.572×10^{-1}	<i>NA</i>
320	30	1.086×10^{-1}	0.61	3.449×10^{-1}	0.54	1.977×10^{-1}	0.85
1280	70	6.470×10^{-2}	0.75	1.944×10^{-1}	0.85	9.770×10^{-2}	1.02
5120	110	3.827×10^{-2}	0.76	1.153×10^{-1}	0.75	5.011×10^{-2}	0.96
20480	195	2.232×10^{-2}	0.78	6.714×10^{-2}	0.78	2.648×10^{-2}	0.92

Table 1: Numerical convergence study comparing computed solutions with the exact geodesic distance to a point on the hemisphere. For each of the successively refined meshes, errors are shown using the optimal p value.

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.

4.1.1. Distance to point on 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 (x, y, z) on the hemisphere and Γ_1 is given by

$$d_{\text{exact}} = \arccos \left(\frac{\sqrt{2}}{2}x + \frac{1}{2}y + \frac{1}{2}z \right). \quad (14)$$

The solutions u_p may be approximated using either the Newton algorithm or ADMM algorithm. Results are shown for the ADMM algorithm. The Newton algorithm gives almost identical results. The first row of Figure 3 shows the computed p -Poisson distance approximations using a low p value ($p = 5$) and a high p value ($p = 100$). For $p = 5$, 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.

Table 1 shows L^2 , H^1 , and L^∞ error norms comparing to the exact distance (14). Approximations are performed on successively refined surface meshes and convergence rates are measured relative to the maximum cell edge length, h . For each mesh, we take the p value at which the errors level off. Since convergence is approximately linear in $1/p$, the optimal p value increases with the grid refinement. Table 1 shows convergence across all three norms. However, the rates are slightly below linear as we are approximating a nonsmooth distance function with the Newton or ADMM algorithm.

Figure 4 shows how the primal and dual residuals evolve with the number of ADMM iterations for various p values. Additionally, Figure 4 tracks how well the computed p -Poisson distance satisfies $|\nabla^S u_p| = 1$ versus ADMM iterations for various p values using a normalized L^1 error norm. Computations in Figure 4 are performed on the most refined mesh from Table 1 with 20480 cells. 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 for

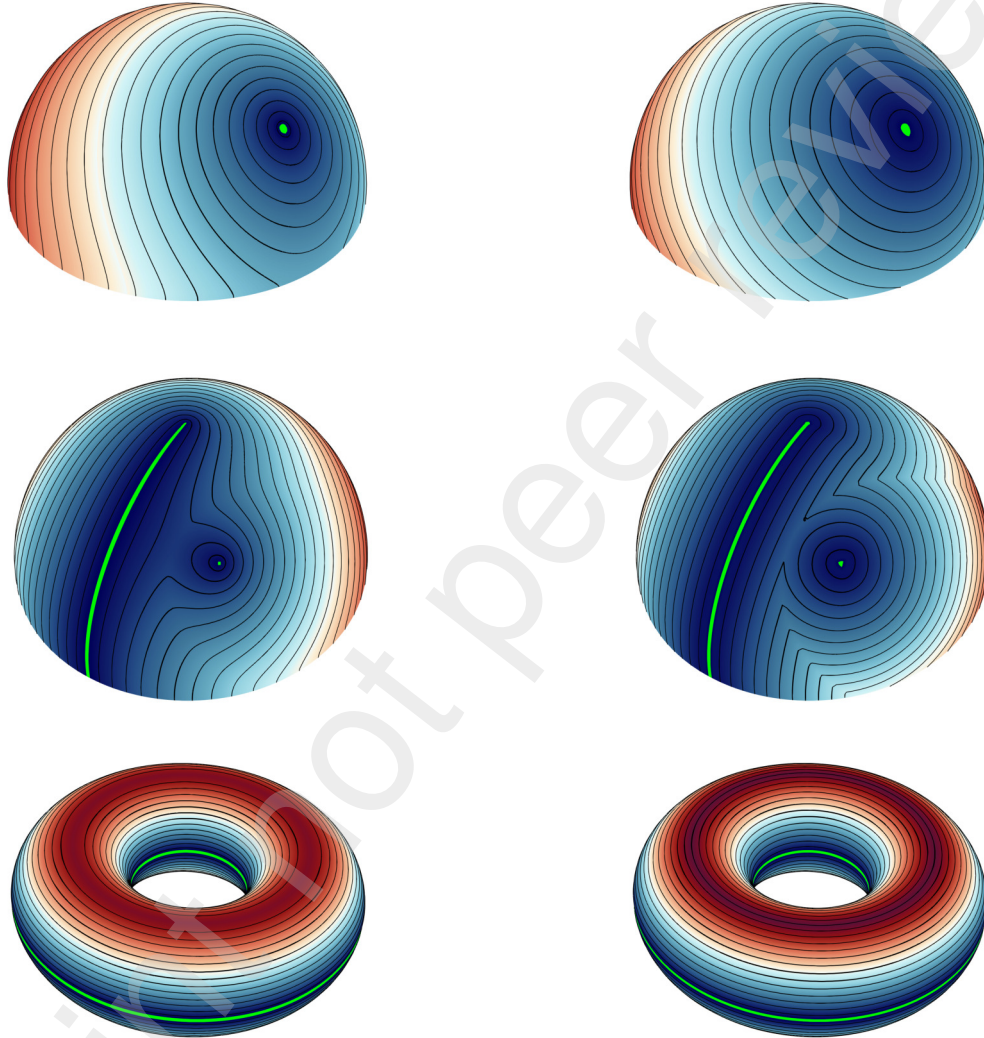


Figure 3: Computed distance approximations using $p = 5$ (left) and $p = 100$ (right). Top row: distance to a point on hemisphere. Second row: distance to an open curve and a point on hemisphere. Bottom row: distance to two closed curves on torus. For each plot, Γ_1 is shown in green.

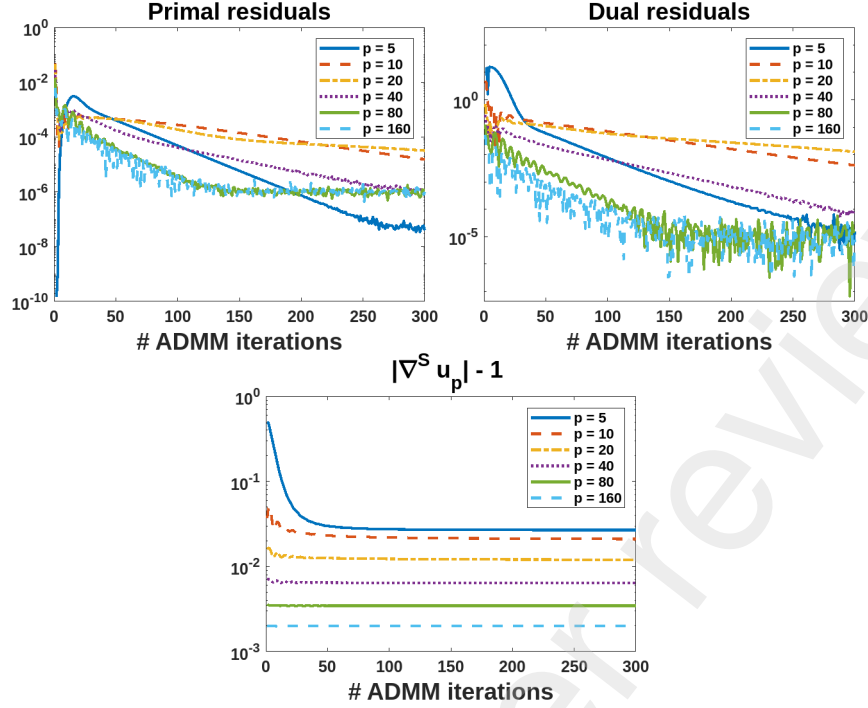


Figure 4: Tracking of residuals and gradient norm for the example of distance to a point on the hemisphere using various p values. Top row: primal (left) and dual (right) residuals versus ADMM iterations. Bottom row: normalized L^1 norm of $|\nabla^S u_p| - 1$ versus ADMM iterations.

the previous (smaller) p value, this initialization process reduces the amount of iterations needed for the current (larger) p value. However, this is not necessary to see convergence using the ADMM scheme. Unlike Newton, ADMM will converge for a poor initial guess.

4.1.2. Distance to open curve and point on hemisphere

We now consider an example which 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 the hemisphere,

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

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

We observe on the second row of Figure 3 that the contours sharpen for larger p values and the influence of the Neumann condition diminishes. Particularly, for $p = 5$ we see smoothing around points which are equidistant to the point and open curve which comprise Γ_1 . For $p = 100$, there is no perceptible smoothing of the distance.

Table 2 shows L^2 , H^1 , and L^∞ error norms for both Newton and ADMM algorithms. We may obtain an analytical expression for the geodesic distance using the Spherical Law of Cosines and compare our computed solutions against this exact distance. As with the previous example, computations are performed on successively refined surface meshes and we take the p value at which the errors level off. Table 2 shows that convergence across all three norms still holds for a more complicated feature set. We also see that the errors are

Newton algorithm							
# cells	p	L^2 error	Rate	H^1 error	Rate	L^∞ error	Rate
80	20	4.626×10^{-2}	NA	3.010×10^{-1}	NA	1.271×10^{-1}	NA
320	35	2.627×10^{-2}	0.82	1.883×10^{-1}	0.68	8.079×10^{-2}	0.65
1280	60	1.599×10^{-2}	0.72	1.154×10^{-1}	0.71	5.072×10^{-2}	0.67
5120	105	9.539×10^{-3}	0.75	6.979×10^{-2}	0.73	3.090×10^{-2}	0.71
20480	180	5.583×10^{-3}	0.77	4.257×10^{-2}	0.71	1.861×10^{-2}	0.73

ADMM algorithm							
# cells	p	L^2 error	Rate	H^1 error	Rate	L^∞ error	Rate
80	20	4.626×10^{-2}	NA	3.010×10^{-1}	NA	1.271×10^{-1}	NA
320	35	2.627×10^{-2}	0.82	1.883×10^{-1}	0.68	8.079×10^{-2}	0.65
1280	60	1.599×10^{-2}	0.72	1.154×10^{-1}	0.71	5.072×10^{-2}	0.67
5120	105	9.539×10^{-3}	0.75	6.978×10^{-2}	0.73	3.089×10^{-2}	0.71
20480	180	5.534×10^{-3}	0.79	4.260×10^{-2}	0.71	1.860×10^{-2}	0.73

Table 2: Numerical convergence studies comparing computed solutions with exact geodesic distance to an open curve and a point on the hemisphere. For each of the successively refined meshes, errors are shown using the optimal p value. Results are shown using both the Newton algorithm (top) and ADMM algorithm (bottom).

almost identical for Newton and ADMM algorithms. This shows that the ADMM algorithm is able to obtain the same level of accuracy as the Newton algorithm.

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 shows the computed distance using a low p value ($p = 5$) and a high p value ($p = 100$). Since the torus is a closed surface, there is no error from the Neumann boundary condition. Lower p values give smoother approximations and higher p values give sharper, more accurate approximations.

Table 3 shows L^2 , H^1 , and L^∞ error norms comparing the computed solutions with the exact geodesic distance to the two closed curves where $y = 0$ on the torus. Results are shown for successively refined surface meshes using the ADMM algorithm; the Newton algorithm gives almost indistinguishable results. As with the examples on the hemisphere, Table 3 shows convergence across all three norms.

We may also monitor how the p -Poisson energy (3) evolves with the ADMM (or Newton) iterations. Figure 5 plots the energy versus number of iterations for a variety of p values. As with the residuals, computations in Figure 5 are performed on the mesh from Table 3 with 24576 cells and for each $p > 5$, we initialize with the computed solution from the previous

# cells	p	L^2 error	Rate	H^1 error	Rate	L^∞ error	Rate
384	10	1.690×10^{-1}	<i>NA</i>	1.936×10^0	<i>NA</i>	6.538×10^{-2}	<i>NA</i>
1536	25	6.753×10^{-2}	1.32	1.448×10^0	0.42	3.575×10^{-2}	0.87
6144	75	2.797×10^{-2}	1.27	1.055×10^0	0.46	1.328×10^{-2}	1.43
24576	200	1.116×10^{-2}	1.33	7.591×10^{-1}	0.47	7.788×10^{-3}	0.77
98304	460	4.997×10^{-3}	1.16	5.413×10^{-1}	0.49	4.118×10^{-3}	0.92

Table 3: Numerical convergence studies comparing computed solutions with exact geodesic distance to closed curves on a torus. For each of the successively refined meshes, errors are shown using the optimal p value.

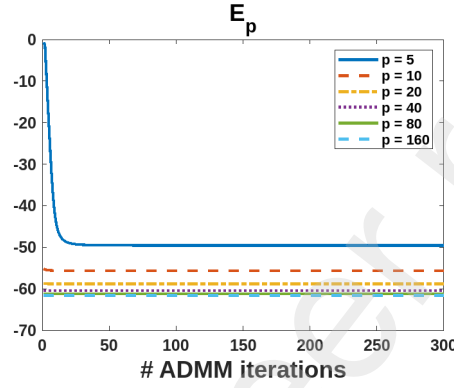


Figure 5: p -Poisson energy (3) versus ADMM iterations for distance to closed curves on torus using various p values.

(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 an unsuitable choice of parameters, i.e., taking the penalty parameter, β , too small.

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. Surface meshes which are only available to download as triangulations are converted to quadrangulations using <https://github.com/martemyev/tethex> in order to be compatible with deal.ii [33]. Results are shown using the ADMM algorithm since this allows us to compute for a large p value without iterating in p . However, the Newton algorithm yields very similar results.

Figure 6 shows the computed p -Poisson distance to Γ_1 (shown in green) using both $p = 5$ and $p = 100$ on four different surfaces. 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 and the $p = 100$ solution possesses more uniform spacing between contours. These differences are most noticeable on the fingers and legs. The third row shows distance to Γ_1 consisting of two points on the octopus. Here, there is a significant increase in sharpness of the contours going from $p = 5$ to $p = 100$. Lastly, the bottom row shows the distance to Γ_1

consisting of two open curves on an open surface. We have sliced off one side of the cylinder so this example has nonempty Γ_2 as well as sharp edges on the box. As with the octopus, we observe a smoother result for lower p .

In Figure 7, we plot the computed distance approximations to curves which split the left and right side of the pig mesh. We only show one angle due to symmetry across Γ_1 . Results are shown for $p = 2, 5, 10, 100, 200$. We observe a large difference between smaller p values ($p = 2, p = 5, p = 10$) and almost no difference between $p = 100$ and $p = 200$. For each p value, the same number of contours is shown. We notice the contours become more sharp and evenly spaced as p increases.

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 8 shows the normalized L^1 norm of $|\nabla^S u_p| - 1$ versus the number ADMM iterations for various p values. The example considered in Figure 8 is the distance to ear example on the pig mesh seen in row 2 of Figure 6. It appears that the computed solution gradient is approaching something with norm 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 $|\nabla^S u_p| - 1$ occurs only during the first few iterations for $p = 5$. This behaviour is also observed for other examples.

Figure 9 tests how well our p -Poisson distance approximations adhere to the triangle inequality. The top row of Figure 9 shows computations of the distance to a point on the ear (q_1) and a point on the lower body (q_2) for $p = 5$. The bottom row of Figure 9 shows that $d_p(q_1, x) + d_p(x, q_2) \geq \min\{d_p(q_1, q_2), d_p(q_2, q_1)\}$ throughout the entire mesh. We note that there is a small difference between our computations of $d_p(q_1, q_2)$ and $d_p(q_2, q_1)$ which is why we take $\min\{d_p(q_1, q_2), d_p(q_2, q_1)\}$. In contrast to our p -Poisson approximations, distances using Crane et al.'s [13] method often fail to satisfy the triangle inequality on significant patches across varied parameter choices. For this example, Crane et al.'s [13] method has large regions for which the triangle inequality is violated. See Figure 9 in [10] for 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.

In Figure 10, we show different angles of the computed p -Poisson distance to a point on the Stanford bunny mesh. Results are shown for $p = 40$ obtained from the Newton algorithm. We observe that the unnatural holes in this mesh do not significantly obstruct the contour lines. This supports that the homogeneous Neumann condition on Γ_2 is appropriate.

In Figure 11, we plot the the computed distance approximations to the tip on the right middle toe of the armadillo. Results are shown for $p = 100$. We see some minor differences in the contour lines but the coloring and contour structure are similar. The distance to toe problem for the armadillo and nearly isometrically deformed armadillo is also considered by Lipman et al. [9] and Solomon et al. [10].

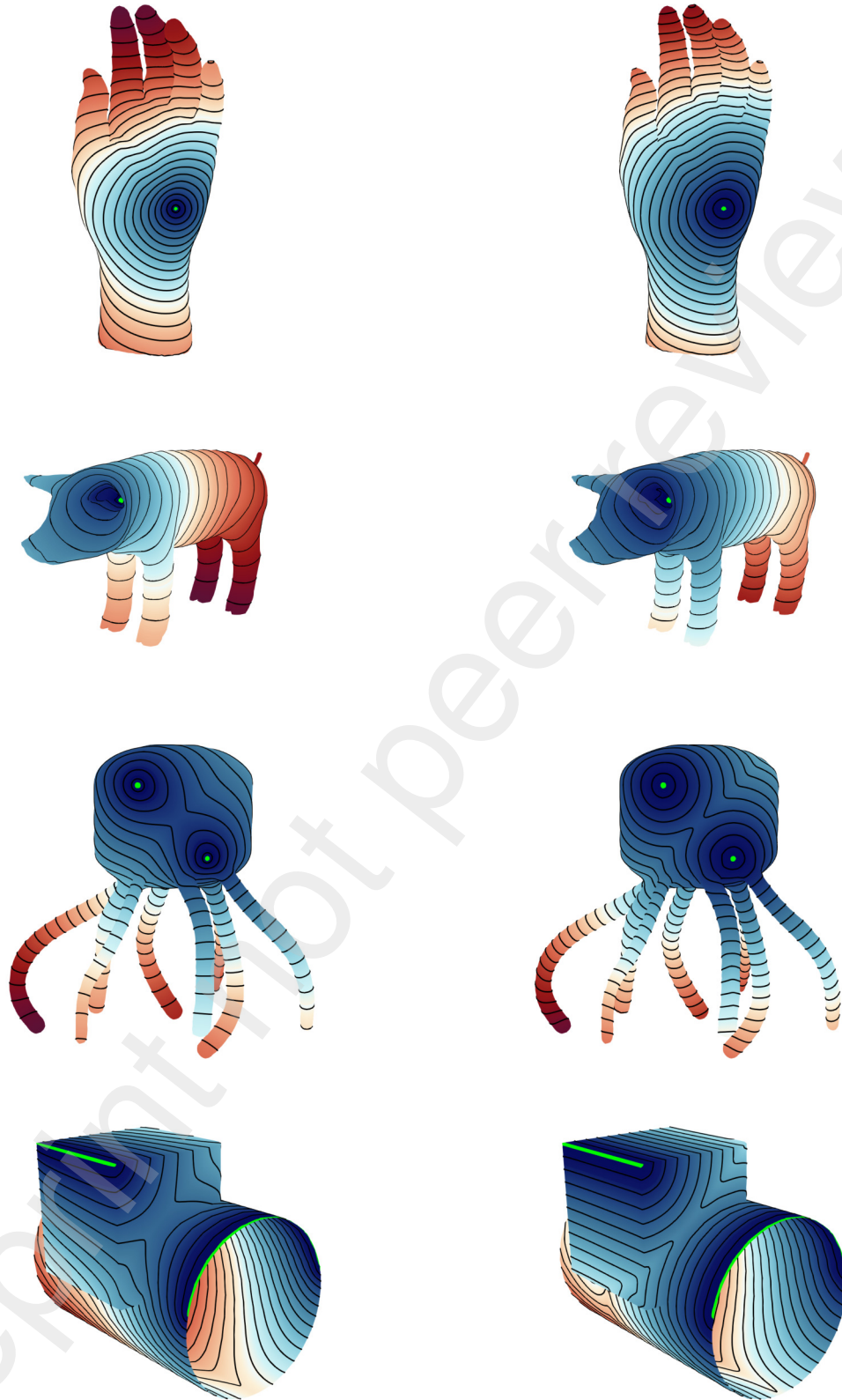


Figure 6: Computed distance approximations using $p = 5$ (left) and $p = 100$ (right) on various surfaces. For each plot Γ_1 is shown in green. Top row: distance to a point on the palm of the hand. Second row: distance to the tip of an ear on the pig. Third row: closest distance to 2 distinct points on the octopus. Bottom row: closest distance to two open curves.

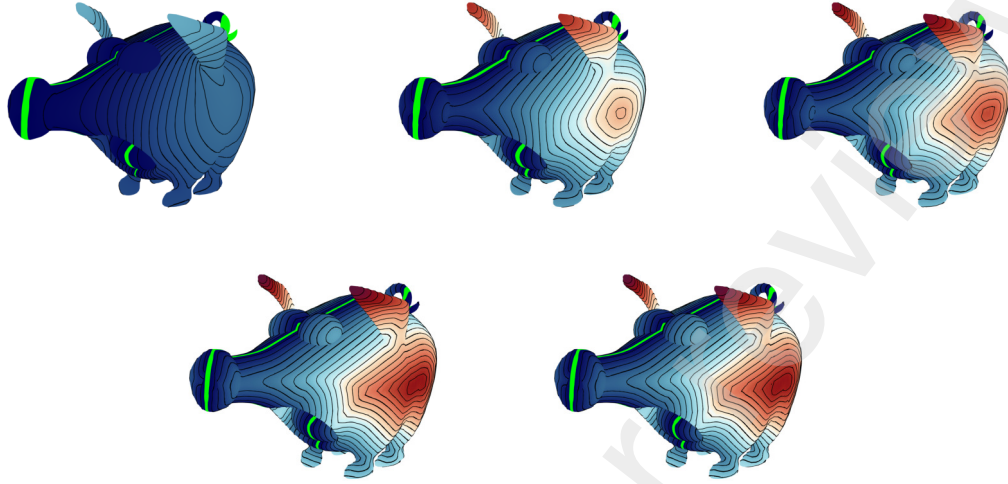


Figure 7: Computed p -Poisson distances to curves which separate the left and right sides of the pig. Γ_1 is shown in green. Top row: $p = 2$ (left), $p = 5$ (center), $p = 10$ (right). Bottom row: $p = 100$ (left) and $p = 200$ (right). Note that the contours become more sharp and evenly spaced as p increases.

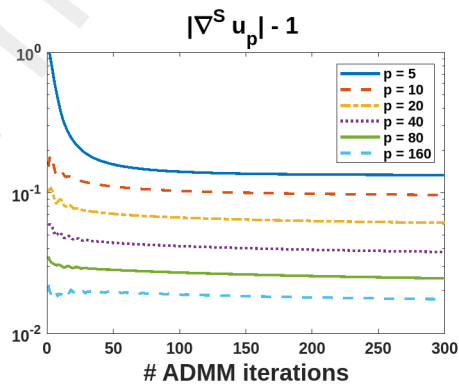


Figure 8: Normalized L^1 norm of $|\nabla^S u_p| - 1$ versus ADMM iterations for various p values. Results are shown for the distance to ear example on the pig.

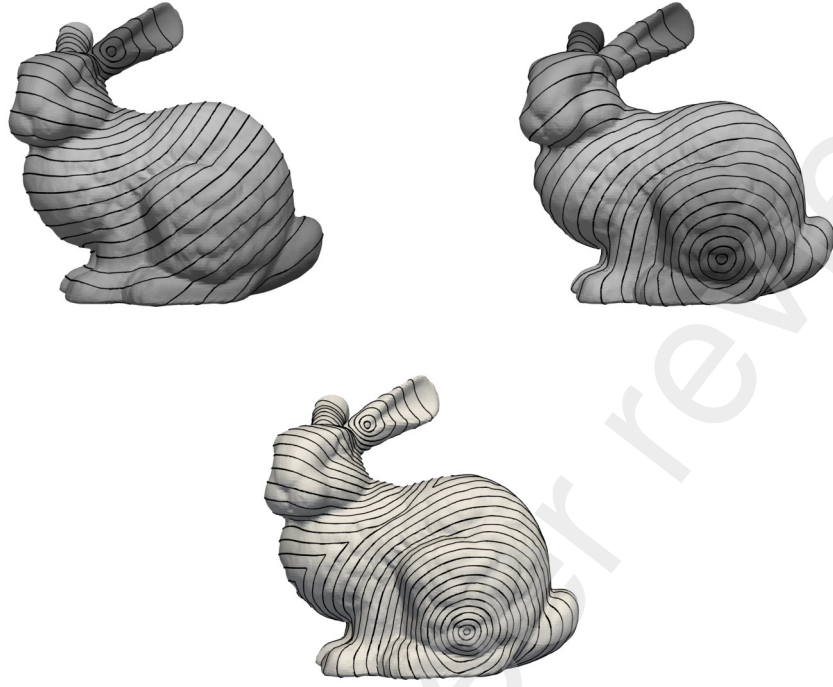


Figure 9: Triangle inequality check. Top row: computed distance approximations for $p = 5$ with level sets of $d_p(q_1, x)$ (left) and $d_p(q_2, x)$ (right). Here, q_1 and q_2 correspond to a point on the ear and to a point on the lower body of the bunny, respectively. Bottom row: Regions which satisfy the triangle inequality are shown in white with level sets of $\min\{d_p(q_1, x), d_p(q_2, x)\}$. Notice the entire bunny is white, implying satisfaction of the triangle inequality.



Figure 10: Two different angles of computed p -Poisson distance to a point on the Stanford bunny mesh which contains artificial holes. Results are shown using $p = 40$.



Figure 11: Computed p -Poisson distances to a toe using $p = 100$. Γ_1 is shown in green. Results are shown for the armadillo (left) and a nearly isometrically deformed armadillo (right).

Crane et al. [13] and Solomon et al. [10] both demonstrate that their methods for distance approximation can withstand some per-vertex noise. Figure 12 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 ℓ 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, which is required with Crane et al.’s [13] method. Both the Newton scheme and ADMM scheme for solving the boundary value problem demonstrate numerical convergence to the exact geodesic distance as $p \rightarrow \infty$. Although the ADMM scheme may require a large amount of iterations to reach the same level of accuracy as the Newton scheme, it 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. The numerical methods presented are also adaptable to different problems involving Δ_p^S so this work provides a starting point for the study of other applications, such as optimal transport.

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 Pouët pig mesh is provided courtesy of Crane’s 3D Model Repository. The remaining meshes are obtained from the Chen et al. [35] benchmark set for segmentation.

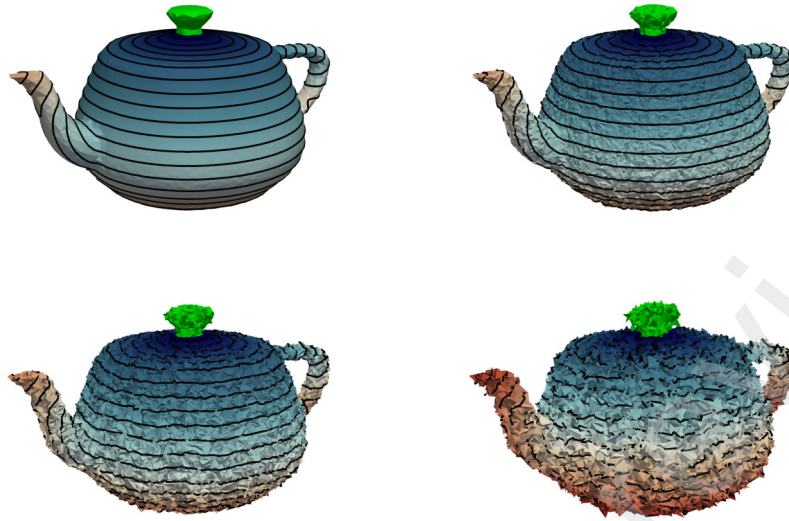


Figure 12: Computed p -Poisson distances to the lid of a teapot with Gaussian noise added to the mesh vertices. The standard deviation is in terms of the average mesh cell edge length ℓ . Top left: no noise. Top right: minimal amount of noise ($\sigma = \frac{\ell}{8}$). Bottom left: medium amount of noise ($\sigma = \frac{\ell}{4}$). Bottom right: large amount of noise ($\sigma = \frac{\ell}{2}$).

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] 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.
- [3] 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.
- [4] 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.
- [5] V. Caselles, L. Igual, O. Sander, An axiomatic approach to scalar data interpolation on surfaces, *Numerische Mathematik* 102 (2006) 383–411.
- [6] M. Ruzicka, *Electrorheological fluids: Modeling and mathematical theory*, Lecture Notes in Mathematics, Springer Berlin Heidelberg 1748 (2000).
- [7] 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.

- [8] T. Bhattacharya, E. DiBenedetto, J. Manfredi, Limits as $p \rightarrow \infty$ of $\Delta_p u = f$ and related extremal problems, *Rend. Sem. Mat. Univ. Politec. Torino* 47 (1989) 15–68.
- [9] Y. Lipman, R. M. Rustamov, T. A. Funkhouser, Biharmonic distance, *Association for Computing Machinery Trans. Graph.* 29 (3) (2010).
- [10] J. Solomon, R. Rustamov, L. Guibas, A. Butscher, Earth mover’s distances on discrete surfaces, *Association for Computing Machinery Trans. Graph.* 33 (4) (2014).
- [11] 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).
- [12] A. G. Belyaev, P.-A. Fayolle, On variational and PDE-based distance function approximations, *Computer Graphics Forum* 34 (8) (2015) 104–118.
- [13] K. Crane, C. Weischedel, M. Wardetzky, The heat method for distance computation, *Commun. Association for Computing Machinery* 60 (11) (2017) 90–99.
- [14] S. R. S. Varadhan, Diffusion processes in a small time interval, *Communications on Pure and Applied Mathematics* 20 (4) (1967) 659–685.
- [15] 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.
- [16] 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.
- [17] 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.
- [18] O. Weber, Y. S. Devir, A. M. Bronstein, M. M. Bronstein, R. Kimmel, Parallel algorithms for approximation of distance maps on parametric surfaces, *Association for Computing Machinery Trans. Graph.* 27 (4) (2008).
- [19] 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.
- [20] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, H. Hoppe, Fast exact and approximate geodesics on meshes, *Association for Computing Machinery Trans. Graph.* 24 (3) (2005) 553–560.
- [21] R. Kimmel, J. A. Sethian, Computing geodesic paths on manifolds, *Proceedings of the National Academy of Sciences* 95 (15) (1998) 8431–8435.
- [22] T. Pryer, On the finite-element approximation of ∞ -harmonic functions, *Proceedings of the Royal Society of Edinburgh: Section A Mathematics* 148 (2018) 819 – 834.

- [23] F. LIU, Solutions to a class of parabolic inhomogeneous normalized p-Laplace equations, *Acta Mathematica Scientia* 35 (2) (2015) 477–494.
- [24] G. Dziuk, C. M. Elliott, Finite element methods for surface PDEs, *Acta Numerica* 22 (2013) 289–396.
- [25] 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.
- [26] F. Alvarez, S. Flores, Existence and approximation for variational problems under uniform constraints on the gradient by power penalty, *Society for Industrial and Applied Mathematics Journal on Mathematical Analysis* 47 (5) (2015) 3466–3487.
- [27] J. W. Barrett, W. B. Liu, Finite element approximation of the parabolic p-Laplacian, *Society for Industrial and Applied Mathematics Journal on Numerical Analysis* 31 (2) (1994) 413–428.
- [28] Y. Q. Huang, R. Li, W. Liu, Preconditioned descent algorithms for p-Laplacian, *Journal of Scientific Computing* 32 (2007) 343–371.
- [29] Z. Luo, F. Teng, An effective finite element Newton method for 2D p-Laplace equation with particular initial iterative function, *Journal of Inequalities and Applications* 2016 (2016) 1–24.
- [30] 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.
- [31] 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.
- [32] 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.
- [33] 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.
- [34] U. Ayachit, *The ParaView Guide: A Parallel Visualization Application*, Kitware, Inc., 2015.
- [35] X. Chen, A. Golovinskiy, T. Funkhouser, A benchmark for 3D mesh segmentation, *Association for Computing Machinery Transactions on Graphics (Proc. SIGGRAPH)* 28 (3) (2009).