On Variational and PDE-based Distance Function Approximations

Alexander G. Belyaev[1] and Pierre-Alain Fayolle[2]

[1]Institute of Sensors, Signals and Systems, School of Engineering & Physical Sciences, Heriot-Watt University, Edinburgh, UK
[2]Computer Graphics Laboratory, University of Aizu, Aizu-Wakamatsu, Japan

a.belyaev@hw.ac.uk   fayolle@u-aizu.ac.jp

Abstract
In this paper, we deal with the problem of computing the distance to a surface (a curve in 2D) and consider several distance function approximation methods which are based on solving partial differential equations (PDEs) and finding solutions to variational problems. In particular, we deal with distance function estimation methods related to the Poisson-like equations and generalized double-layer potentials. Our numerical experiments are backed by novel theoretical results and demonstrate efficiency of the considered PDE-based distance function approximations.

Keywords: distance function approximations, variational methods, iterative optimization.

Categories and Subject Descriptors (according to ACM CCS): I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Geometric algorithms, languages, and systems; G.1.8 [Numerical Analysis]: Partial Differential Equations—Iterative solution techniques.

1. Introduction

Efficient computation of distance maps and their approximations and generalizations is an important component of many recent studies including those on object and action recognition [Zuc13, WBR14], surface reconstruction [CT11], computational mechanics [FST11], computational photography [CSCRP10], medical imaging [RC13], computer graphics [CWW13, CHK13, YWH13], architectural geometry [PHD*10], and computational fluid dynamics [Tuc11, RS13]. In particular, in the computer graphics field, "shape aware" substitutes and approximations of the exact distance functions are becoming increasingly popular [RLF09, LRF10, PBDSH13, SRGB14]. See also references therein.

In this paper, we deal with the problem of approximating the distance to a surface (a curve in 2D). More precisely, let us consider a bounded domain Ω in $\mathbb{R}^m$ and assume that $\partial\Omega$ is oriented by its inner normal $\boldsymbol{n}$. Denote by

$$d(\boldsymbol{x}) = \text{dist}(\boldsymbol{x}, \partial\Omega)$$

the distance from $\boldsymbol{x} \in \Omega$ to $\partial\Omega$. The problem of fast and reliable approximation of $d(\boldsymbol{x})$ is important, for example, for level-set methods [OF01, EZL*12, SOG14], computational mechanics applications [BST04, FST11, XT11, XTC12], turbulence modelling [Tuc11, Tuc14], and pattern recognition studies [GBS*07, Zuc13].

We consider several distance function approximation methods which are based on energy minimization and solving partial differential equations (PDEs). In particular, we deal with distance function estimation methods related to the Poisson, screened Poisson and p-Poisson equations and generalized double-layer potentials. The contribution of the paper is threefold:

- we introduce a variational approach for distance function estimation and propose efficient iterative schemes for corresponding energy minimization problems;
- we investigate several normalization procedures used to enhance distance function approximations and, in particular, develop a normalization scheme for the p-Laplacian;
- we extend our previous results [BFP13] by analyzing asymptotic properties of the L_p-distance fields and demonstrating how they can be used for an accurate estimation of the distance function near the boundary.

The rest of the paper is organized as follows. In Section 2, we describe some basic properties of the distance function. Section 3 is devoted to a brief explanation of the heat propagation approach proposed in [Var67] for distance function estimation and employed in [CWW13] for a highly efficient computation of surface geodesics. In Section 4, we propose and study simple iterative schemes for approximating the distance function. We apply the alternating direction method of multipliers (ADMM) to derive a more sophisticated iterative scheme in Section 5. In Section 6, we discuss various normalization schemes used to estimate the distance function. In Section 7, we develop a novel normalization scheme to be used with the solution of a p-Poisson equation. In Section 8, we study asymptotic properties of the so-called L_p-distance fields introduced in [BFP13]. It allows us to achieve a highly accurate estimation of the distance function near the boundary by normalized L_p-distance fields. Section 9 is devoted to numerical experiments with the proposed variational PDE-based methods for distance function estimation. We conclude and indicate directions for future work in Section 10. Finally, the appendix contains technical derivations of our normalization schemes for p-Laplacian distances (subsection A) and our theoretical results on asymptotic properties of the L_p-distance fields (subsection B).

2. Basic properties of the distance function

Let us assume that $\partial\Omega$ is oriented by its inner normal $\boldsymbol{n}$. It is well-known that the distance function $d(\boldsymbol{x})$ satisfies the eikonal equation

$$|\nabla d| = 1 \quad \text{in} \quad \Omega. \tag{1}$$

and boundary conditions

$$d = 0 \quad \text{and} \quad \partial d/\partial\boldsymbol{n} = 1 \quad \text{on} \quad \partial\Omega, \tag{2}$$
$$\partial^k d/\partial\boldsymbol{n}^k = 0 \quad \text{on} \quad \partial\Omega, \quad k = 2,3,\dots.$$

Typically (1) is used with the first (Dirichlet) boundary condition of (2).

It is well known (see, for example, [Giu84, Appendix B] that the Laplacian of the distance function $d(\boldsymbol{x})$ is proportional to the mean curvature $H(\boldsymbol{x})$ of the distance function level set $d = \text{const}$ passing through $\boldsymbol{x}$

$$\Delta d = (1-m)H, \tag{3}$$

where m is the number of dimensions and where we assume that the level set of $H(\boldsymbol{x})$ is smooth at $\boldsymbol{x}$. Here $H = (k_1 + \cdots + k_{m-1})/(m-1)$, where $k_1,\dots,k_{m-1}$ are the principal curvatures of $d = \text{const}$. Since $(1-m)H(\boldsymbol{x}) = \text{div}(\boldsymbol{n})$, where $\boldsymbol{n}(\boldsymbol{x}) = \nabla d(\boldsymbol{x})$ is the unit inner normal of the level set of $d(\boldsymbol{x})$ at $x \in \Omega$, we arrive at a 2nd-order nonlinear PDE

$$\Delta d = \text{div}(\nabla d/|\nabla d|) \tag{4}$$

which is equivalent to (3). It is interesting that (4) serves as

the Euler-Lagrange equation for the energy

$$E(u) = \int_\Omega (|\nabla u| - 1)^2 \, d\boldsymbol{x}, \tag{5}$$

which was considered, for example, in [LXGF05] and [XQYH12].

3. Varadhan's distance functions

In this section, inspired by a recent work of Crane et al. [CWW13], we exploit S. R. S. Varadhan's approach [Var67] to approximating smooth distance functions using solutions to the so-called screened Poisson equation

$$w - t\Delta w = 0 \quad \text{in} \quad \Omega, \qquad w = 1 \quad \text{on} \quad \partial\Omega, \tag{6}$$

where t is a small, positive parameter. Then, according to [Var67, Theorem 2.3],

$$\lim_{t\to 0} \left\{ -\sqrt{t}\ln[w(\boldsymbol{x})] \right\} = d(\boldsymbol{x}), \tag{7}$$

Thus $d(\boldsymbol{x})$ is approximated by

$$u(\boldsymbol{x}) = -\sqrt{t}\ln w(\boldsymbol{x}) \tag{8}$$

and parameter t controls the smoothing properties of $u(\boldsymbol{x})$.

An intuitive explanation of (7) is given in [GR09] and uses a variant of the so-called Hopf-Cole transformation [Eva98]. Substituting

$$v(\boldsymbol{x}) = \exp\left\{ -u(\boldsymbol{x})/\sqrt{t} \right\} \tag{9}$$

in (6) leads to

$$0 = v - t\Delta v = v\left[\left(1 - |\nabla u|^2\right) + \sqrt{t}\Delta u \right]. \tag{10}$$

and we arrive at a regularized eikonal equation for $u(\boldsymbol{x})$

$$\left(1 - |\nabla u|^2\right) + \sqrt{t}\,\Delta u = 0 \quad \text{in} \quad \Omega, \quad u = 0 \quad \text{on} \quad \partial\Omega \tag{11}$$

and it is natural to expect that $u(\boldsymbol{x})$ provides a better and better approximation of $d(\boldsymbol{x})$, as $t \to 0$.

A distance function approximation closely related to (6) and (8) was proposed in [CWW13]. Note that $w(x)$ and $u(\boldsymbol{x})$ have the same level sets. Crane et al. suggested to consider the normalized gradient $\nabla w/|\nabla w|$ and approximate (1) by the solution to a simple least square minimization problem

$$\int_\Omega (\nabla\phi - \nabla w/|\nabla w|)^2 d\boldsymbol{x} \to \min.$$

In other words, $d(\boldsymbol{x})$ is approximated by $\phi(\boldsymbol{x})$ which satisfies

$$\Delta\phi(\boldsymbol{x}) = \text{div}(\nabla w/|\nabla w|) \text{ in } \Omega, \quad \phi(\boldsymbol{x}) = 0 \text{ on } \partial\Omega. \tag{12}$$

4. Method of Laplacian iterations and its relaxation and splitting

The gradient normalization scheme (12) of [CWW13] inspires us to consider a simple iterative process

$$\Delta d_{i+1} = \text{div}(\nabla d_i/|\nabla d_i|) \quad \text{in} \quad \Omega,$$
$$d_{i+1} = 0 \quad \text{on} \quad \partial\Omega \tag{13}$$

which delivers fixed-point iterations for (4).

One can interpret (13) as follows. Given $d_i(\boldsymbol{x})$, its normalization $\boldsymbol{n}_i(\boldsymbol{x}) = \nabla d_i(\boldsymbol{x})/|\nabla d_i(\boldsymbol{x})|$ coincides with $\boldsymbol{n}(\boldsymbol{x}) = \nabla d(\boldsymbol{x})$ on $\partial\Omega$. On the other hand, $v = d_{i+1}(\boldsymbol{x})$ is the solution to

$$\int_\Omega |\nabla v - \boldsymbol{n}_i|^2 \, d\boldsymbol{x} \to \min. \qquad (14)$$

Thus, normalization $d_i \to \boldsymbol{n}_i = \nabla d_i/|\nabla d_i|$ reduces the approximation error near $\partial\Omega$, while (14) redistributes the remaining error over Ω.

We face two problems when dealing with iterative procedure (13): our inability to establish its convergence properties rigorously and a relatively slow convergence to the true distance function $d(\boldsymbol{x})$ in practice. One possible way to accelerate (13) consists of rewriting (5) as
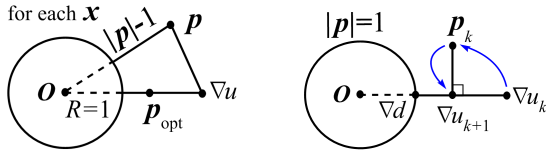
$$\int_\Omega (|\boldsymbol{p}| - 1)^2 \, d\boldsymbol{x}, \quad \text{where} \quad \boldsymbol{p} = \nabla u$$

and then relaxing the constraint $\boldsymbol{p} = \nabla u$ (the so-called penalty method)

$$E(\boldsymbol{p}, u) = \int_\Omega \left\{ (|\boldsymbol{p}| - 1)^2 + \frac{r}{2}(\boldsymbol{p} - \nabla u)^2 \right\} d\boldsymbol{x}, \qquad (15)$$

where r is positive. Note that (15) attains its minimal value, 0, if and only if $u = d(\boldsymbol{x})$ and $\boldsymbol{p} = \nabla d$.

Let us fix ∇u in (15) and optimize it w.r.t. $\boldsymbol{p}$. For given $\boldsymbol{x} \in \Omega$, the optimal $\boldsymbol{p}$ in (15) is proportional to ∇u, that is $\boldsymbol{p}(\boldsymbol{x}) = c(\boldsymbol{x})\nabla u(\boldsymbol{x})$ for some $c(\boldsymbol{x})$, as illustrated by the left image of Fig. 1,

Figure 1: *Left: at given $\boldsymbol{x} \in \Omega$, for fixed ∇u, optimal $\boldsymbol{p}$ in (15) is proportional to ∇u. Right: a schematic explanation of iterative process (18), see the main text for details.*

Substituting $\boldsymbol{p} = c\nabla u$ to (15) and optimizing w.r.t. c yields

$$c = \frac{2 + r|\nabla u|}{(2+r)|\nabla u|}, \quad \boldsymbol{p} = c\nabla u = \frac{2 + r|\nabla u|}{(2+r)|\nabla u|}\nabla u. \qquad (16)$$

Optimizing (15) w.r.t. $u(\boldsymbol{x})$ leads to

$$\Delta u = \operatorname{div}\boldsymbol{p} \quad \text{in} \quad \Omega$$

or, equivalently,

$$\int_\Omega |\nabla u - \boldsymbol{p}|^2 \, d\boldsymbol{x} \to \min, \qquad (17)$$

where $u = 0$ on $\partial\Omega$. Thus we are arriving at an iterative procedure

$$\boldsymbol{p}_k = \frac{2 + r|\nabla u_k|}{(2+r)|\nabla u_k|}\nabla u_k \quad \text{and} \quad \Delta u_{k+1} = \operatorname{div}\boldsymbol{p}_k \qquad (18)$$

which is reduced to (13) if $r = 0$ and, therefore, can be considered as a generalization of (13).

Intuitively, convergence of (18) can be justified as follows. The first step of (18), $\nabla u_k \to \boldsymbol{p}_k$, makes $\boldsymbol{p}_k$ closer to the unit sphere $|\boldsymbol{p}| = 1$ than ∇u_k is. Indeed, according to (15) we have

$$\int_\Omega (|\boldsymbol{p}_k| - 1)^2 \, d\boldsymbol{x} \le E(\boldsymbol{p}_k, u_k) \qquad (19)$$
$$\le E(\nabla u_k, u_k) = \int_\Omega (|\nabla u_k| - 1)^2 \, d\boldsymbol{x}$$

In view of (17), the second step of (18), $\boldsymbol{p}_k \to \nabla u_{k+1}$, can be considered as the orthogonal projection of $\boldsymbol{p}_k$ onto the linear subspace of gradients ∇u. Therefore, ∇u_{k+1} is closer to $|\boldsymbol{p}| = 1$ than $\boldsymbol{p}_k$ is. See the right image of Fig. 1 for an illustration.

Assume now that u_k and $\boldsymbol{p}_k$ converge in a proper sense to u^* and $\boldsymbol{p}^*$, respectively. Then, in view of the second inequality in (19), we have $\nabla u^* = \boldsymbol{p}^*$. Passing to the limit in the second equation of (16) yields $|\nabla u^*| = 1$ and, therefore, $u^*(\boldsymbol{x})$ is the distance function $d(\boldsymbol{x})$.

According to our experiments (see Section 9 for details), (18) demonstrates a faster convergence than simple Laplacian iterations (13). Better numerical algorithms for minimizing (5) can be derived by exploiting a similarity between (5) and minimization problems associated with sparse signal recovery methods (see, for example, [HLY13, Chapter 4] for a friendly introduction to sparse optimization algorithms). In particular, in the next section, we demonstrate how the alternating direction method of multiplies (ADMM) can be used for minimizing (5).

5. Alternating direction method of multiplies (ADMM) for minimizing (5)

For better handling constraint $\boldsymbol{p} = \nabla u$ in (15), we can use ADMM (see, for example, [BPE*11] for an excellent overview of ADMM and its extensions and modifications) and add a Lagrange multiplier term to (15):

$$\int_\Omega \left\{ (|\boldsymbol{p}| - 1)^2 + \boldsymbol{\lambda} \cdot (\boldsymbol{p} - \nabla u) + \frac{r}{2}(\boldsymbol{p} - \nabla u)^2 \right\} d\boldsymbol{x}, \qquad (20)$$

where $\boldsymbol{\lambda}(\boldsymbol{x})$ is the vector of the Lagrange multipliers.

It is easy to rewrite (20) as

$$\int_\Omega \left\{ (|\boldsymbol{p}| - 1)^2 + \frac{r}{2}\left[\boldsymbol{p} - \left(\nabla u - \frac{\boldsymbol{\lambda}}{r}\right)\right]^2 - \frac{\boldsymbol{\lambda}^2}{2r} \right\} d\boldsymbol{x}. \qquad (21)$$

Thus, for fixed ∇u and $\boldsymbol{\lambda}$, the optimal $\boldsymbol{p}$ is proportional to $\boldsymbol{q} = \nabla u - \boldsymbol{\lambda}/r$. In other words,

$$\boldsymbol{p} = c\boldsymbol{q} = c\left(\nabla u - \boldsymbol{\lambda}/r\right) \qquad (22)$$

for some scalar c. Now substituting (22) into (21) and optimizing w.r.t. c yields

$$c = \frac{2 + rq}{(2+r)q}, \qquad \boldsymbol{p} = c\boldsymbol{q} = \frac{2 + rq}{(2+r)q}\boldsymbol{q} \quad \text{with} \quad q = |\boldsymbol{q}|.$$

Optimizing (20) w.r.t. $u(\boldsymbol{x})$ leads to

$$r\Delta u = r\operatorname{div}\boldsymbol{p} + \operatorname{div}\boldsymbol{\lambda}.$$

Thus, following ADMM, we arrive at the following iterative procedure

$$\boldsymbol{p}_k = \frac{2 + rq_k}{(2+r)q_k}\boldsymbol{q}_k \quad \text{with} \quad \boldsymbol{q}_k = \nabla u_k - \boldsymbol{\lambda}_k/r, \ q_k = |\boldsymbol{q}_k|, \ (23)$$

$$\Delta u_{k+1} = \operatorname{div}\boldsymbol{p}_k + \operatorname{div}\boldsymbol{\lambda}_k/r \ \text{ in } \ \Omega, \ u_{k+1} = 0 \ \text{ on } \ \partial\Omega, \ (24)$$

$$\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + r\left(\boldsymbol{p}_k - \nabla u_k\right). \tag{25}$$

Notice that instead of (24) one can use

$$\nabla u_{k+1} = \boldsymbol{p}_k + \boldsymbol{\lambda}_k/r \tag{26}$$

and, therefore, solving systems of linear equations for each iteration is not required. However (24) imposes zero boundary conditions on $u_{k+1}(\boldsymbol{x})$ and that helps to reduce the approximation error at least near the boundary. In contrast, (26) does not care about boundary conditions. One possible way to use (26) while imposing boundary conditions consists of using (26) for a fixed number of iterations (e.g., for 10 iterations), then use (24), and then again use (26) for the same number of iterations, etc.

6. Rvachev, Taubin, and Spalding-Tucker normalization schemes

In their studies, Rvachev [Rva74, RSST01] and Taubin [Tau94] dealt with the problem of computing an approximate distance from a point to implicitly defined surfaces (curves in the two-dimensional case). Assume that $\partial\Omega$ is the zero-level set of function $u(\boldsymbol{x})$. Then according to [Rva74] $d(\boldsymbol{x})$ can be estimated by

$$\omega_1[u](\boldsymbol{x}) = \frac{u(\boldsymbol{x})}{\sqrt{u(\boldsymbol{x})^2 + |\nabla u(\boldsymbol{x})|^2}}. \tag{27}$$

Alternatively, Taubin considered approximations of $d(\boldsymbol{x})$ near $\partial\Omega$ by

$$\delta_1[u](\boldsymbol{x}) = u(\boldsymbol{x})/|\nabla u(x)| \tag{28}$$

$$\delta_2[u](\boldsymbol{x}) = \frac{1}{F}\left(\sqrt{|\nabla u|^2 + 2Fu} - |\nabla u|\right)$$
$$= \frac{2u}{\sqrt{|\nabla u|^2 + 2Fu} - |\nabla u|}, \tag{29}$$

where F is the Frobenious norm of the Hessian of $u(\boldsymbol{x})$.

Interestingly, setting $F = 1$ in (29) leads to

$$\nu[u](\boldsymbol{x}) = \sqrt{|\nabla u|^2 + 2u} - |\nabla u| \tag{30}$$
$$= \frac{2u}{|\nabla u| + \sqrt{|\nabla u|^2 + 2u}},$$

a simple normalization scheme proposed by Spalding [Spa94] and further developed by Tucker [Tuc98] in a relation to their studies of turbulence phenomena.

It is easy to see that $\omega_1[u]$, $\delta_1[u]$, $\delta_2[u]$, and $\nu[u]$ satisfy (2) and, therefore, deliver accurate approximations of $d(\boldsymbol{x})$ near $\partial\Omega$. Further refinements are possible. For example, following [Rva74] (see also [Sha07]) one can set

$$\omega_2(\boldsymbol{x}) = \omega_1(\boldsymbol{x}) - \frac{1}{2}\omega_1^2\frac{\partial^2\omega_1}{\partial\boldsymbol{n}^2} \tag{31}$$

and arrive at

$$\omega_2 = 0, \quad \partial\omega_2/\partial\boldsymbol{n} = 1, \quad \partial^2\omega_2/\partial\boldsymbol{n}^2 = 0 \quad \text{on} \quad \partial\Omega. \tag{32}$$

Unfortunately, in a general case, normalizations $\omega_1[u](\boldsymbol{x})$, $\omega_2[u](\boldsymbol{x})$, $\delta_1[u](\boldsymbol{x})$, $\delta_2[u](\boldsymbol{x})$, $\nu[u](\boldsymbol{x})$, and similar schemes may behave unpredictably far from $\partial\Omega$.

7. Poisson and p-Laplacian (p-Poisson) distances

A common approach for a quick and simple approximation of a smooth distance function consists of solving the homogeneous Dirichlet problem for Poisson's equation

$$\Delta u = -1 \quad \text{in} \quad \Omega, \qquad u = 0 \quad \text{on} \quad \partial\Omega. \tag{33}$$

Applications of the Poisson distance function $u(\boldsymbol{x})$ include action recognition [GBS*07, GGS*06], shape skeletonisation [AA12], estimating the so-called wall distance in turbulence modeling [Tuc98], and geometric de-featuring [XTC12].

While the approximation accuracy of the true distance function by the solution to (33) is rather poor, a simple normalization procedure (30) proposed in [Spa94, Tuc98] significantly improves the approximation accuracy. See also [XTC12] for recent applications of (33) and (30) for medial axis detection and geometric de-featuring.

Let us now consider a natural generalization of (33)

$$\operatorname{div}\left(|\nabla u_p|^{p-2}\nabla u_p\right) = -1 \quad \text{in} \ \Omega, \quad u_p = 0 \ \text{on} \ \partial\Omega \tag{34}$$

with $2 \le p < \infty$. It can be shown [BDM89, Kaw90] that

$$u_p(\boldsymbol{x}) \to d(\boldsymbol{x}) \quad \text{as} \quad p \to \infty. \tag{35}$$

Moreover, according to [BDM89], convergence (35) is strong in the Sobolev space $W^{1,k}(\Omega)$ for arbitrary $k > 1$.

Similar to (30) one can normalize $u_p(\boldsymbol{x})$ and arrive at

$$v_p(\boldsymbol{x}) = -|\nabla u_p|^{p-1} + \left[\frac{p}{p-1}u_p + |\nabla u_p|^p\right]^{\frac{p-1}{p}}. \tag{36}$$

See Appendix A of this paper for a derivation. It is easy to verify that

$$v_p = 0 \quad \text{and} \quad \partial v_p/\partial\boldsymbol{n} = 1 \quad \text{on} \quad \partial\Omega. \tag{37}$$

Therefore, one can expect that $v_p(\boldsymbol{x})$ delivers a more accurate approximation of the distance function than $u_p(\boldsymbol{x})$.

8. L_p-distance fields

Following [BFP13] let us consider a singular potential

$$\int_{\partial\Omega} \frac{\boldsymbol{n}_y \cdot (\boldsymbol{y} - \boldsymbol{x})}{|\boldsymbol{x} - \boldsymbol{y}|^{m+p}} \sigma(\boldsymbol{y}) \, dS_y, \quad p \geq 0, \tag{38}$$

where $\boldsymbol{x} \in \Omega \subset \mathbb{R}^m$, $\boldsymbol{y} \in \partial\Omega$, $\boldsymbol{n}_y$ is the orientation normal at $\boldsymbol{y}$, dS_y is the surface element at $\boldsymbol{y}$, and $\sigma(\boldsymbol{y})$ is a density function defined on $\partial\Omega$. The classical double-layer potential corresponds to (38) with $p = 0$. As discussed in [BFP13], setting $p = 1$ in (38) yields integrals used to define the mean value coordinates [Flo03, JSW05].

Generalized double-layer potentials (38) and their single-layer counterparts were also studied by R. Rustamov [Rus07] in relation to generalized barycentric coordinates and transfinite interpolations and their applications to free-form deformations.

Let us assume that $\sigma \equiv 1$ which means that the generalized dipoles are uniformly distributed over $\partial\Omega$ and consider

$$\Phi_p(\boldsymbol{x}) = \int_{\partial\Omega} \frac{\boldsymbol{n}_y \cdot (\boldsymbol{y} - \boldsymbol{x})}{|\boldsymbol{x} - \boldsymbol{y}|^{m+p}} \, dS_y = \int_{\Sigma} \frac{d\Sigma_y}{|\boldsymbol{x} - \boldsymbol{y}|^p}, \tag{39}$$

where $p > 0$, Σ is the unit sphere centered at $\boldsymbol{x}$, and $d\Sigma_y$ is the solid angle at which surface element dS_y is seen from $\boldsymbol{x}$. We have used a simple relation

$$dS_y = \rho^m d\Sigma_y / h, \text{ where } \rho = |\boldsymbol{x} - \boldsymbol{y}| \text{ and } h = \boldsymbol{n}_y (\boldsymbol{y} - \boldsymbol{x}).$$

Here h is the distance from $\boldsymbol{x}$ to the plane tangent to $\partial\Omega$ at $\boldsymbol{y}$. The two-dimensional version of (39) can be written in polar coordinates as

$$\Phi_p(\boldsymbol{x}) = \int_0^{2\pi} \frac{d\theta}{\rho(\theta)^p}, \tag{40}$$

where θ is the angle between vector $\boldsymbol{y} - \boldsymbol{x}$ and a fixed direction.

The integral over $\partial\Omega$ in (39) is correctly defined for an arbitrary bounded domain Ω. However the integral over unit sphere Σ is properly defined if Ω is star-shaped w.r.t. $\boldsymbol{x}$. To overcome this problem, we follow [BFP13] which, in its turn, follows [DF09, BF10]. Consider a ray originated at $\boldsymbol{x}$ and intersecting $\partial\Omega$ at l points $\boldsymbol{y}_1, \ldots, \boldsymbol{y}_l$. We set $\varepsilon_i = 1$ if the ray $[\boldsymbol{x}, \boldsymbol{y}_i)$ arrives at $\boldsymbol{y}_i$ from the positive side of $\partial\Omega$, $\varepsilon_i = -1$ if the ray approaches $\boldsymbol{y}_i$ from the negative side of $\partial\Omega$, and $\varepsilon_i = 0$ if the ray is tangent to $\partial\Omega$ at $\boldsymbol{y}_i$. Now let us assume that $|\boldsymbol{x} - \boldsymbol{y}|^p$ in the denominator of the integral over the unit sphere Σ in (39) means $\sum_i \varepsilon_i |\boldsymbol{x} - \boldsymbol{y}_i|^p$. See the left image of Fig. 2 below for a visual explanation.

As demonstrated in [BFP13],

$$\Psi_p(\boldsymbol{x}) = \left[c_p / \Phi_p(\boldsymbol{x}) \right]^{1/p}, \quad p \geq 1, \tag{41}$$

where c_p is a certain normalization constant, approximates the distance function $d(\boldsymbol{x})$. (The particular case $p = 1$ was studied in [DF09, BF10] where $\Phi_1(\boldsymbol{x})$ was used for interpolation purposes.)
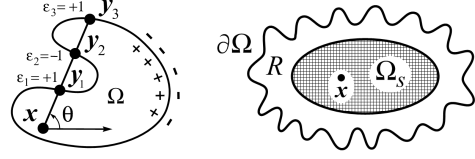
Figure 2: *Left: generalized double layer potential (39) can be defined using polar (spherical in 3D) coordinates. Right: generalized double layer potential is smooth far from $\partial\Omega$, see the main text for an explanation.*

In contrast to the distance function approximations considered in the previous sections of this paper, (41) approximates the signed distance function. Another attractive property of (41) is that it can be calculated analytically for the odd values of p [BFP13].

As we will see below, (41) nicely approximates the distance function near $\partial\Omega$. However, for relatively small p and far from the boundary, (41) is too smooth to approximate the distance function accurately. Indeed, note that

$$\frac{1}{|\boldsymbol{x} - \boldsymbol{y}|^N} = \frac{1}{N - m} \operatorname{div}_y \left(\frac{\boldsymbol{x} - \boldsymbol{y}}{|\boldsymbol{x} - \boldsymbol{y}|^N} \right). \tag{42}$$

Assume that $\partial\Omega$ has bumps, as shown in the right image of Fig. 2, and consider $\Omega_s \subset \Omega$ such that $\partial\Omega_s$ is much smoother than $\partial\Omega$ and $R = \Omega \setminus \Omega_s$ is small. In view of (42) and the divergence theorem, the difference between the potentials (39) for $\partial\Omega$ and $\partial\Omega_s$ is given by

$$p \int_R \frac{d\boldsymbol{y}}{|\boldsymbol{x} - \boldsymbol{y}|^{m+p}}$$

which is small if R is small and x is inside Ω_s and far from $\partial\Omega_s$. Therefore, the approximations of the distance functions from $\partial\Omega$ and $\partial\Omega_s$ by (41) are close to each other at $\boldsymbol{x}$.

Nevertheless one can easily improve the distance function estimation by (41) far from the boundary. Generalized mean value potential $\Phi_p(\boldsymbol{x})$ satisfies some interesting properties including the following one (see Proposition 3 in [BFP13])

$$\Delta\Phi_p(\boldsymbol{x}) = p(p + m)\Phi_{p+2}(\boldsymbol{x}) \tag{43}$$

which delivers a simple procedure to improve the approximation accuracy at x.

It turns out that in the two-dimensional case

$$\Phi_p(\boldsymbol{x}) = \int_0^{2\pi} \frac{d\theta}{\rho(\theta)^p} = \frac{c_p}{h^p} + k \frac{d_p}{h^{p-1}} + O\left(\frac{1}{h^{p-2}} \right) \tag{44}$$

with

$$c_p = \int_{-\pi/2}^{\pi/2} \cos^p \theta \, d\theta \quad \text{and} \quad d_p = c_{p-2}/2,$$

where k is the curvature of $\partial\Omega$. See Appendix B for a derivation.

It is interesting that for $p = 1$ corresponding to mean-value coordinates we have

$$\Phi_1(\boldsymbol{x}) \sim \frac{2}{h} + 2k\ln\frac{1}{h} + O(1), \quad \text{as } h \to 0.$$

Similar to the 2D case, in 3D we have

$$\Phi_p(\boldsymbol{x}) \sim \int_\Sigma \frac{d\Sigma}{\rho(\theta,\varphi)^p} = \frac{c_p}{h^p} + H\frac{d_p}{h^{p-1}} + O\left(\frac{1}{h^{p-2}}\right) \quad (45)$$

where the integration is taken over the unit sphere Σ, H is the surface mean curvature of $\partial\Omega$, and

$$c_p = \frac{1}{2}\int_\Sigma \cos^p\theta\, d\Sigma \quad \text{with} \quad d\Sigma = \sin\theta\, d\theta\, d\varphi.$$

In particular, $c_1 = \pi$.

Notice that (44) and (45) imply that

$$\Psi_p = 0 \quad \text{and} \quad \partial\Psi_p/\partial\boldsymbol{n} = 1 \quad \text{on} \quad \partial\Omega. \quad (46)$$

Obviously one can get more from (44) and (45). Namely we have

$$\Psi_p(\boldsymbol{x}) = h - \frac{d_p k}{p c_p}h^2 + O\left(h^3\right),$$

where k is the curvature in the 2D (curve) case and the mean curvature in the 3D (surface) case. So one possible way to get a higher-order normalization of $\Psi_p(\boldsymbol{x})$ is similar to (31) and consists of

$$\tilde\Psi_p(\boldsymbol{x}) = \Psi_p(\boldsymbol{x}) + \frac{d_p k}{p c_p}[\Psi_1(\boldsymbol{x})]^2 = h + O\left(h^3\right) \quad (47)$$

as $h \to 0$. Thus

$$\tilde\Psi_p = 0, \quad \partial\tilde\Psi_p/\partial\boldsymbol{n} = 1, \quad \partial^2\tilde\Psi_p/\partial\boldsymbol{n}^2 = 0 \quad \text{on} \quad \partial\Omega \quad (48)$$

and $\tilde\Psi_p(\boldsymbol{x})$ is supposed to deliver a more accurate approximation of the distance function $d(\boldsymbol{x})$ than $\Psi_p(\boldsymbol{x})$ near $\partial\Omega$.

9. Numerical experiments

9.1. Discretization

In our numerical experiments, we solve the various partial differential equations appearing via the finite element method (FEM). We use piecewise linear hat functions. In 2D, the domain Ω is discretized by triangles. We use the program Triangle [She96] for meshing. The left image in Fig. 3 shows a triangulation of a 2D polygon used in some of our experiments. The right image presents the exact distance field for the polygon.

In our 3D experiments, we deal with tetrahedral meshes generated from surface meshes by applying the volumetric mesher from the CGAL library [Cga]. In particular, we used CGAL to generate volumetric models (tetrahedral meshes) corresponding to the Stanford bunny, Fertility, and Armadillo surface meshes.

Scalar fields are defined at each node, and interpolated inside the elements (triangles or tetrahedra) by piecewise linear functions. Vector fields are defined per element. Some of
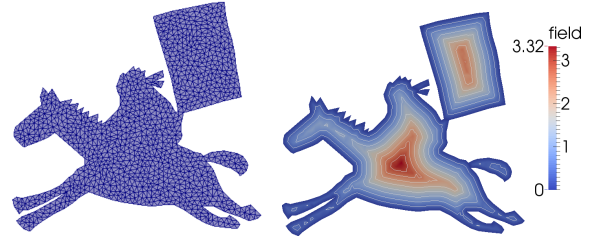
Figure 3: *Left: the triangulated domain used for our two-dimensional experiments. Right: the exact distance field.*

the proposed methods (Sections 7, 4 or 5) require the computation of the gradient of a scalar field or the divergence of a vector field on the discretization. We compute their approximations following the descriptions given in [CWW13, Section 3.2.1] for a triangulation and [TLHD03] for a tetrahedral mesh.

9.2. Initial solution with the Poisson distance

Since an initial solution is needed to most of the iterative schemes introduced in this paper, we use the solution to the Poisson equation (33). In practice, it is possible to get a better initial approximation when starting with the Spalding-Tucker normalization (30) applied to the solution of the Poisson equation (33). The left column of Fig. 4 compares the result obtained with the Spalding-Tucker normalization (30) applied to the solution of the Poisson equation (33) in the bottom-row, against the solution of the Poisson equation (33) in the top-row.

In practice, since the normalized solution gives a better approximation of the distance, we use it as an initial solution for each of the iterative methods (Sections 4, 5, and 7).

9.3. Iterative schemes

Given an initial solution computed by solving (33) and using the normalization (30), we can compute an approximation of the distance by iterations of (13). Figure 5 illustrates the results obtained by this approach after $5, 10$ and 15 iterations.

In practice, using the iterative procedure in (18) seems to converge faster. Figure 6 illustrates the results obtained by respectively $5, 10$ and 15 iterations of the steps in (18). The rightmost image in Fig. 6 indicates for comparison the solution obtained by iterating until the relative error: $\frac{\|u_{k+1}-u_k\|_1}{\|u_k\|_1}$ is below some threshold.

9.4. ADMM Splitting

Alternatively, one can use ADMM described in Section 5 to solve (5). The results of this approach are shown for $5, 10$ and 15 iterations in Fig. 7. Here we use the step (24) every 5
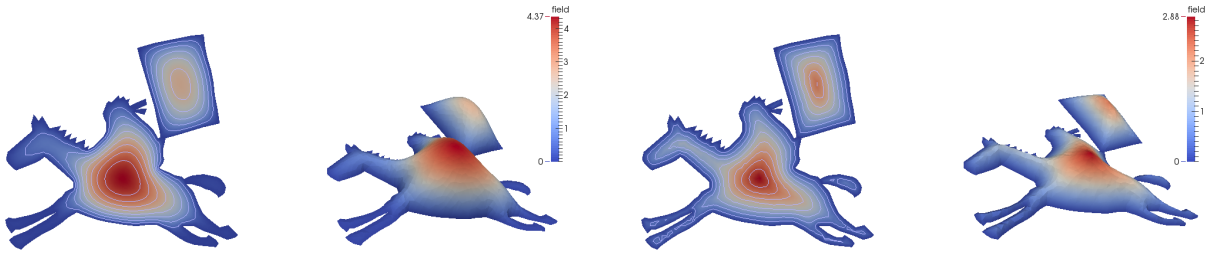
Figure 4: *Poisson (left) and normalized Poisson (right) distances. The field on the left is obtained from the solution of (33), which is then normalized by the Spalding-Tucker normalization (30) in order to obtain the result on the right.*
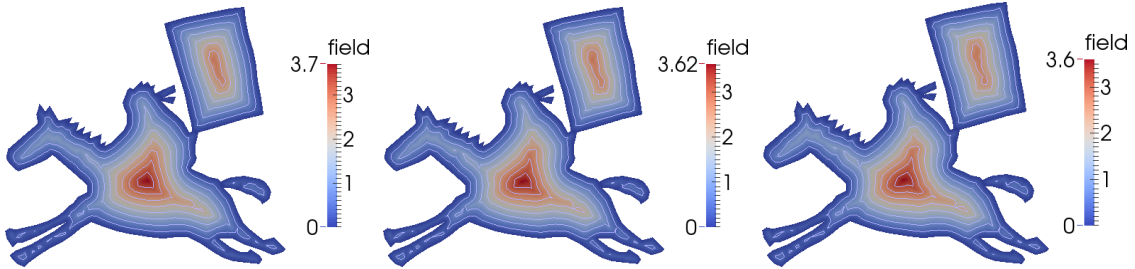
Figure 5: *Approximate distance field obtained after* 5, 10 *and* 15 *iterations of (13).*
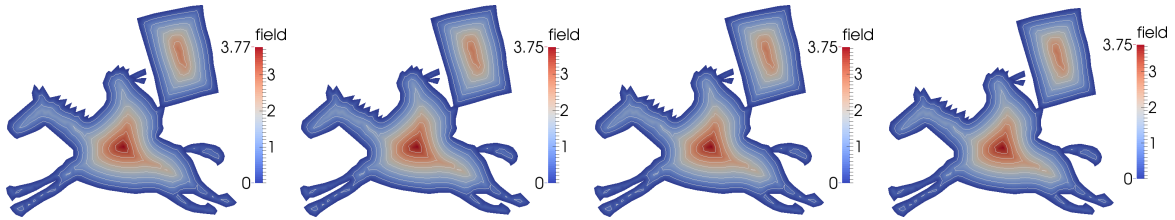
Figure 6: *Solution to the iterative scheme (18) after* 5, 10, 15 *iterations and until convergence.*

iterations to update the field u_{k+1}, otherwise we only update ∇u_{k+1} with (26). For comparison, the rightmost picture in Fig. 7 illustrates the result obtained by iterating the process until the relative error: $\frac{\|u_{k+1}-u_k\|_1}{\|u_k\|_1}$ is below some threshold.

As illustrated in Fig. 8, solving (24) at every step gives a more accurate approximation of the distance (compare the left picture in Fig. 8 to the exact distance field in Fig. 13). However, solving (24) requires solving a sparse linear system at each step. One practical alternative is to solve (24) every l iterations and the rest of the time to update only ∇u_{k+1} (instead of u_{k+1}) with (26). The middle picture in Fig. 8 illustrates this approach with $l = 5$. Compare this result (middle) with the result obtained (left) when a linear system is solved at each step. Finally, the rightmost image in Fig. 8 always uses (26). The result is less accurate, but it is fast: it solves only two sparse linear systems of equation: one to

compute the initial solution by solving (33), and one at the end to recover u from ∇u.

9.5. *p*-Laplacian

In practice, it seems that we can get a better approximation from the solution of the *p*-Poisson equation (34) and its normalization (36). The equation in (34) is non-linear. We solve it by the Newton method with the Jacobian matrix approximated by the stiffness matrix (see for example [LB13, Chapter 9]).

In Fig. 9, the result of the *p*-Laplacian distance (34) for the case $p = 4, 8, 10$ is illustrated in the top row. The bottom row corresponds to its normalized solution (36). They both produce an accurate approximation of the exact distance function.
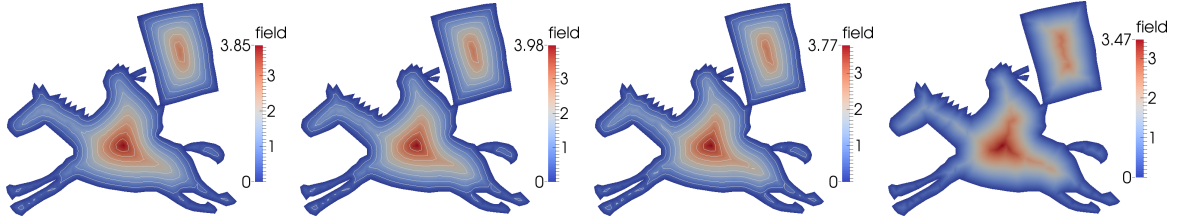
Figure 7: *Solution to (5) using ADMM. From left to right: the result after 5, 10, 15 iterations and until convergence.*
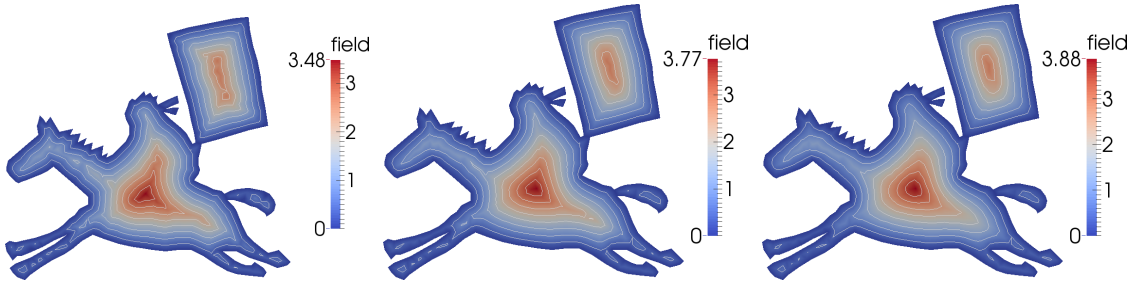
Figure 8: *Solution to the iterative scheme with ADMM after 15 iterations. The left picture corresponds to the case where a linear system is solved at each iteration. The middle picture corresponds to the case where a linear system is solved every 5 iterations. The rightmost picture corresponds to the case where a linear system is solved only at the end.*
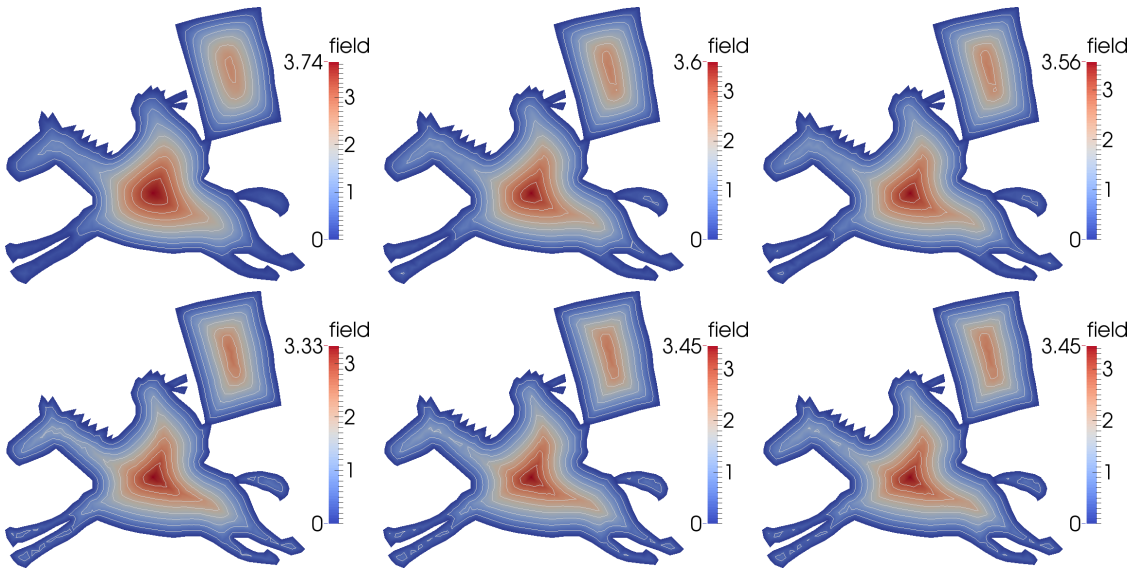
Figure 9: *Top row: p-Laplacian (p-Poisson) distances for $p = 4, 8, 10$. Bottom row: their normalizations (36).*
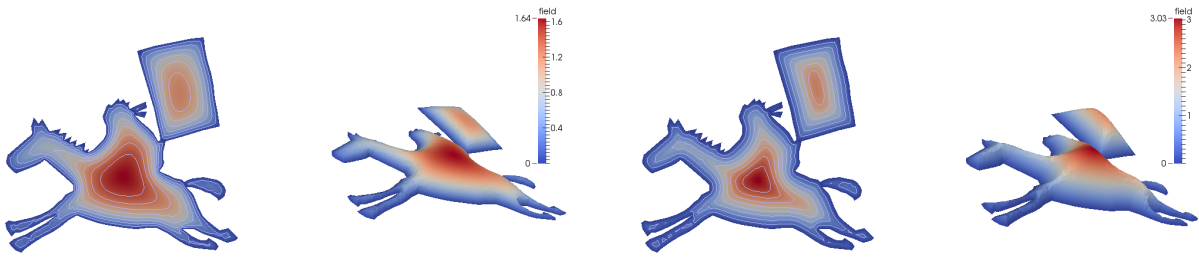
9.6. L_p-distance fields

In Fig. 10, the L_p-distance fields (41): Ψ_1 and Ψ_5 are shown for comparison. While the result obtained from Ψ_1 is only accurate close to the boundary, Ψ_5 delivers a smooth accurate approximate of the exact distance. The interesting thing with this approach is that the solutions can be computed an-

alytically in contrary to the other methods that require a numerical approach (FEM is used in this paper).

We illustrate then two additional properties of the L_p-distance fields: the formula for computing Φ_{p+2} from the Laplacian of Φ_p (43) and a normalization of Ψ_p near the boundary (47). Given Φ_1, one can use (43) to compute Φ_3, by using for example a 5-point stencil to approximate the

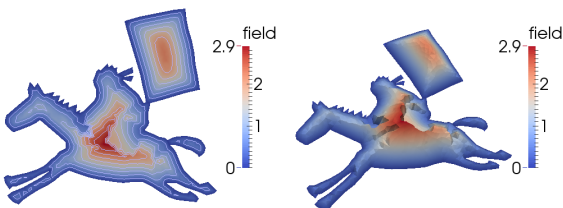**Figure 10:** *The L_p-distance field, for $p = 1$ on the left; and $p = 5$ on the right.*

Laplacian on a regular grid. Ψ_3 can then be computed by using (41). This is illustrated in Fig. 11 where Ψ_3 obtained from a direct evaluation of Φ_3 (left image) is compared against the approximation of Φ_3 computed from the Laplacian of Φ_1 using (43).
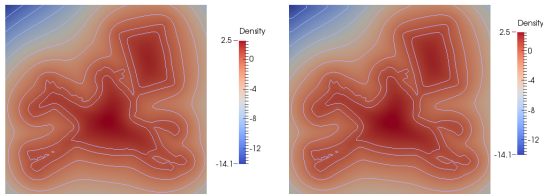
Figure 11: *L_p-distance field for $p = 3$ computed from a direct evaluation of Φ_3 (left) or from its approximation with the Laplacian formula (43).*

Near the boundary of the domain, a normalization of Ψ_p can be applied using (47). This is illustrated in Fig. 12 with the level-sets on the left and the graph of the normalized function on the right. The normalization given by the asymptotic analysis is valid only near the boundary, so it tends to produce artifacts far away.

Figure 12: *Normalization of the L_p-distance field for $p = 5$ using asymptotics near the boundary (47). Left: level-sets of the normalized field; right: the graph of the normalized function.*

9.7. Comparison

In Fig. 13, we compare the various approaches introduced in this paper. Included in the qualitative comparison is also the heat propagation method introduced in [CWW13]. As noted earlier, the heat propagation method [CWW13] corresponds to one iteration of (13) applied to the solution of the heat diffusion equation after one time-step.

Qualitatively, the best result is obtained with the normalized p-Laplacian approach ($p = 10$). Both the iterative method (18) and its solution by ADMM (section 5) provide also good results. For both methods 15 iterations were used. For ADMM, a sparse linear system (24) was solved every 5 iterations. The heat approach from ([CWW13]) is fast, solving only two sparse linear systems. Finally, the L_p-distance is also quite accurate and can be evaluated analytically in contrary to all the other methods that require numerical evaluation.

9.8. Approximate distances to surfaces

All the considered approaches naturally work in 3D for computing the distance to a surface. Figure 14 visualizes the point-wise absolute error between the true distance and the approximate distance fields computed using the geodesics-in-heat method [CWW13] (first column), ADMM (second column), and the p-Laplacian method with $p = 8$ (the third column) for polyhedral volumetric Stanford Bunny and Fertility models. In terms of the approximation accuracy, the p-Laplacian method demonstrates the best performance, while the ADMM and geodesics-in-heat method share the second place: ADDM outperforms geodesics-in-heat for the volumetric Fertility model and the situation is opposite for the volumetric Stanford bunny model.

The computational times taken by these approaches for three volumetric models (the two tetrahedral meshes shown in Figure 14 and the Armadillo model) are indicated in Table 1. In this table, n_{nodes} and n_{ele} correspond respectively to the number of nodes and the number of elements (tetrahedra) in the volumetric mesh used for computing the solutions by FEM. The code is written in C++ without any particular effort for optimization. The timings were obtained on a
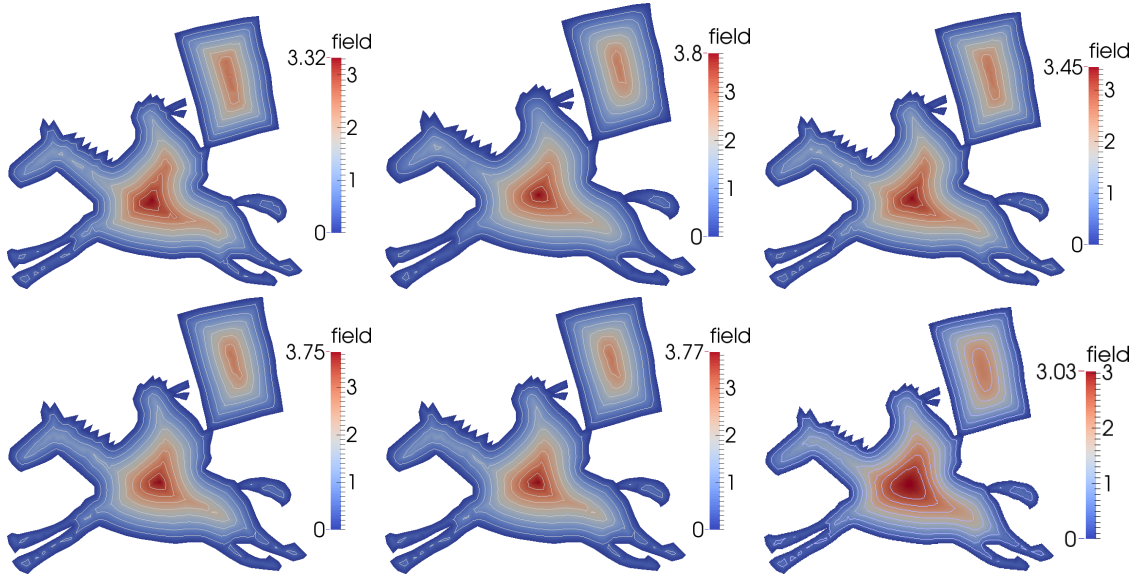
Figure 13: *Top row: exact distance; geodesics-in-heat [CWW13]; p-Laplacian (with $p = 10$). Bottom row: relaxation; relaxation with ADMM (15 iterations); L_p-distance ($p = 5$).*
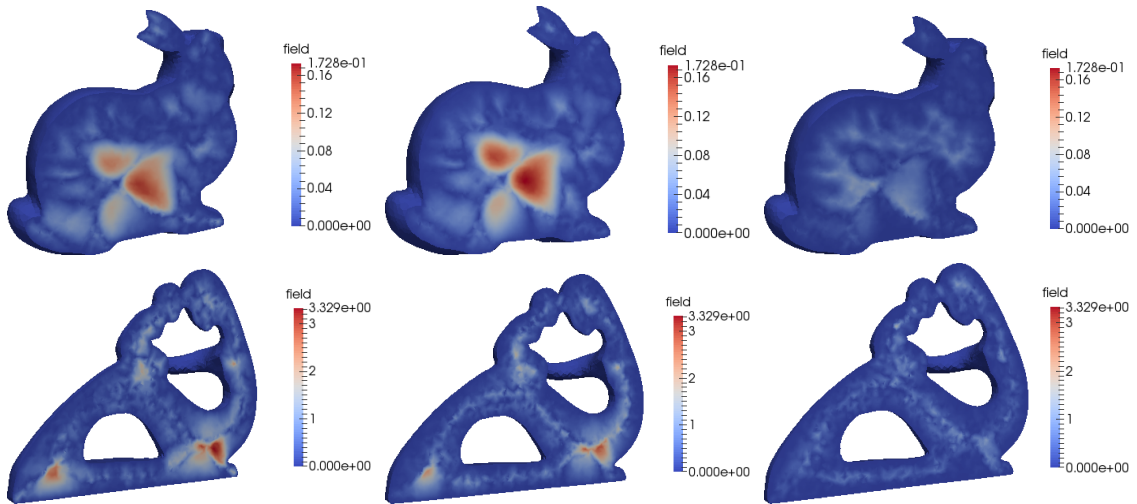
Figure 14: *Point-wise absolute error $|u(x) - \text{dist}(x)|$ computed for tetrahedral 3D meshes, where $\text{dist}(x)$ is the true distance function and $u(x)$ corresponds to the geodesics-in-heat [CWW13] approach (first column); the iterative scheme with ADMM splitting (Section 5) after 15 iterations (second column); the p-Laplacian for $p = 8$ (third column);*

desktop computer with an Intel Core i3-3220 at $3.30\,\text{GHz}$ and $4\,\text{GB}$ of RAM. The matrices involved in all numerical schemes are sparse. The sparse matrix representation and the sparse matrix solvers from the library Eigen [GJ*10] were used in these computations. The timings indicated in Table 1 correspond to the time taken by all the steps (preparation of data-structure, matrix and vector assembly, call to the solver) except IO.

While being the fastest one, the geodesics-in-heat method of Crane et al. [CWW13] is less capable to deliver an accurate approximation of the distance function far from the boundary. The p-Laplacian method provides us with the most accurate but time consuming approximation. In our opinion, ADMM offers the best combination of the computational speed and accuracy.

Note that, in terms of the computational time, all the tested

methods demonstrate a worse performance when dealing with tetrahedral volumetric meshes to compare with triangular planar ones. The reason is that 2D discrete Laplacian matrices are sparser than their 3D counterparts.

9.9. Numerical analysis of convergence properties

For a numerical analysis of convergence properties of the iterative methods introduced in this paper we consider the relative error $\frac{\|u_k - \text{dist}\|_p}{\|\text{dist}\|_p}$, where dist denotes the true distance function and u_k is the k-th iteration obtained by either using Laplacian iterative iterative scheme (13), the relaxation and splitting scheme (18), or by ADMM (see Section 5). As a representative example, we use the 2D rider shape. Same as before, for the solution of (5) by ADMM, we solve a linear system only every 5 iterations.

As expected, ADMM demonstrates excellent convergence properties for both the L^2 and L^∞ norms. The relaxation and splitting scheme (18) shows the best performance for the L^2 norm, while its convergence w.r.t. the L^∞ norm is questionable. Finally, Laplacian iterations (13) demonstrate very good convergence results for the first several dozens of iterations and then bounce back very slightly.

10. Conclusion and future work

We have proposed, studied, and evaluated several new variational and PDE-based distance function approximation schemes. Each scheme has its advantages and disadvantages. The p-Laplacian scheme is computationally expensive but it can be adapted for approximating optimal transportation problems [EG99, Amb03] which are currently a subject of intensive interest in geometry processing and computer graphics [SRGB14]. The L_p-distance fields approximate the signed distance function. They deliver very accurate approximations near the boundary and can be used within the Kantorovich method [Kan41, KK58] and its extensions [Rva82] for numerical solving PDEs with higher-order boundary conditions [Sha07]. Our fast iterative schemes can be further accelerated by using advanced optimization algorithms (see, for instance, [HLY13, Chapter 4]).

We think that our methods have a good potential to enrich each other. For example, a rough distance function approximation generated by one method can be used as a warm start for another, more accurate but computationally expensive method.

In our numerical study of the proposed distance approximation schemes we rely on very simple error analysis tools: a visual comparison with the exact distance function and a comparison of the maximal values attained by the distance approximations and the exact distance (in spite of its simplicity, the latter seems to be a quite reliable indicator of how accurately the distance function is approximated far from the boundary). For a more serious error analysis study, we need to reduce the discretization error. This can be done by using a moving mesh technique which is capable to align mesh edges with distance function singularities. This constitutes another direction for future research.

Finally, extending some of our schemes to the curvilinear metric case is also a topic for future research.

Mesh	n_{nodes}	n_{ele}	heat	ADMM	p-Lap.
Bunny	38k	217k	5.5	11.6	21.2
Fertility	22k	115k	2.6	5.3	111.0
Armadillo	17k	89k	2.0	4.0	51.8

Table 1: *Timing in seconds for computing the distance to a surface with the geodesics-in-heat (heat) method [CWW13] and some of the methods introduced in this paper. These timings correspond to the time taken by all the steps except IO.*

References

[AA12] AUBERT G., AUJOL J.-F.: Poisson skeleton revisited: a new mathematical perspective. *Journal of Mathematical Imaging and Vision* (2012), 1–11.

[Amb03] AMBROSIO L.: Lecture notes on optimal transport problems. In *Mathematical Aspects of Evolving Interfaces, Lecture Notes in Mathematics Volume 1812* (2003), Springer, pp. 1–52.

[BDM89] BHATTACHARYA T., DIBENEDETTO E., MANFREDI J.: Limits as $p \to \infty$ of $\Delta_p u_p = f$ and related extremal problems. *Rend. Sem. Mat. Univ. Pol. Torino, Fascicolo Speciale Nonlinear PDEs* (1989), 15–68.

[BF10] BRUVOLL S., FLOATER M. S.: Transfinite mean value interpolation in general dimension. *J. Comp. Appl. Math. 233* (2010), 1631–1639.

[BFP13] BELYAEV A., FAYOLLE P.-A., PASKO A.: Signed L_p-distance fields. *Computer-Aided Design 45* (2013), 523–528.

[BPE*11] BOYD S., PARIKH N., E. C., PELEATO B., ECKSTEIN J.: Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends in Machine Learning 3*, 1 (2011), 1–122.

[BST04] BISWAS A., SHAPIRO V., TSUKANOV I.: Heterogeneous material modeling with distance fields. *Comput. Aided Geom. Des. 21* (2004), 215–242.

[Cga] CGAL, Computational Geometry Algorithms Library. http://www.cgal.org.

[CHK13] CAMPEN M., HEISTERMANN M., KOBBELT L.: Practical anisotropic geodesy. *Computer Graphics Forum 32*, 5 (2013), 63–71. SGP 2013 issue.

[CSCRP10] CRIMINISI A., SHARP T., CARSTEN ROTHER C., PÉREZ P.: Geodesic image and video editing. *ACM Transactions on Graphics 29* (2010), 134.
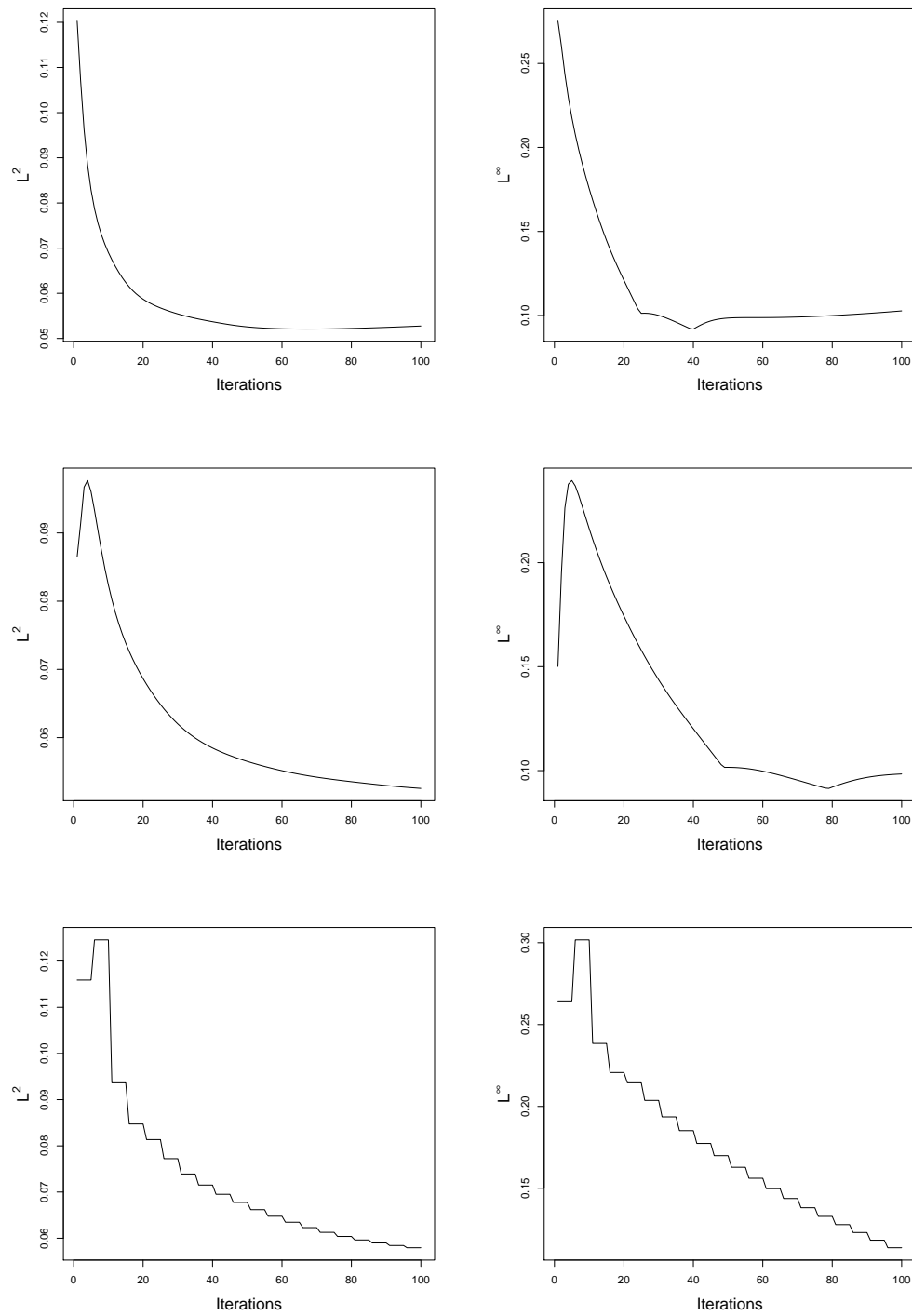
Figure 15: *L^2 (left column) and L^∞ (right column) relative errors for Laplacian iterations (13) (top row), simple relaxation and splitting scheme (18) (middle row), and ADMM (bottom row). The rider domain is used.*

[CT11] CALAKLI F., TAUBIN G.: SSD: smooth signed distance surface reconstruction. *Computer Graphics Forum 30*, 7 (2011), 1993–2002.

[CWW13] CRANE K., WEISCHEDEL C., WARDETZKY M.: Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics 32* (2013), 152:1–152:11.

[DF09] DYKEN C., FLOATER M. S.: Transfinite mean value interpolation. *Comp. Aided Geom. Design 26* (2009), 117–134.

[EG99] EVANS L. C., GANGBO W.: Differential equations methods for the Monge-Kantorovich mass transfer problem. *Mem. Amer. Math. Soc. 131* (1999).

[Eva98] EVANS L. C.: *Partial Differenetial Equations*. American Mathematical Society, 1998.

[EZL*12] ESTELLERS V., ZOSSO D., LAI R., OSHER S., THIRAN J.-P., BRESSON X.: Efficient algorithm for level set method preserving distance function. *IEEE Transactions on Image Processing 21* (2012), 4722–4734.

[Flo03] FLOATER M. S.: Mean value coordinates. *Computer Aided Geometric Design 20*, 1 (2003), 19–27.

[FST11] FREYTAG M., SHAPIRO V., TSUKANOV I.: Finite element analysis in situ. *Finite Elem. Anal. Des. 47*, 9 (2011), 957–972.

[GBS*07] GORELICK L., BLANK M., SHECHTMAN E., IRANI M., BASRI R.: Actions as space-time shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence 29*, 12 (2007), 2247–2253.

[GGS*06] GORELICK L., GALUN M., SHARON E., BASRI R., BRANDT A.: Shape representation and classification using the Poisson equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence 28*, 12 (2006), 1991–2005.

[Giu84] GIUSTI E.: *Minimal Surfaces and Functions of Bounded Variation*. Monographs in Mathematics, Vol. 80. Birkhäuser, 1984.

[GJ*10] GUENNEBAUD G., JACOB B., ET AL.: Eigen v3. http://eigen.tuxfamily.org, 2010.

[GR09] GURUMOORTHY K. S., RANGARAJAN A.: A Schrödinger equation for the fast computation of approximate Euclidean distance functions. In *Scale Space and Variational Methods in Computer Vision (SSMV 2009). LNCS, vol. 5567* (2009), Springer, pp. 100–111.

[HLY13] HAN Z., LI H., YIN W.: *Compressive Sensing for Wireless Networks*. Cambridge University Press, 2013.

[JSW05] JU T., SCHAEFER S., WARREN J.: Mean value coordinates for closed triangular meshes. *ACM Trans. Graph. 24*, 3 (2005), 561–566. ACM SIGGRAPH 2005.

[Kan41] KANTOROVICH L. V.: Some remarks on Ritz's method. *Trydy vysshego voenno-morskogo inzhenerno-stroitel'nogo uchilishcha*, 3 (Leningrad, 1941). (Russian).

[Kaw90] KAWOHL B.: On a family of torsional creep problems. *J. reine angew. Math.s 410*, 1 (1990), 1–22.

[KK58] KANTOROVICH L. V., KRYLOV V. I.: *Approximate Methods of Higher Analysis*. Interscience Publishers, 1958. Chapter 4, § 2.

[LB13] LARSON M., BENGZON F.: *The Finite Element Method*. Springer, 2013.

[LRF10] LIPMAN Y., RUSTAMOV R., FUNKHOUSER T.: Biharmonic distance. *ACM Trans. Graph. 29*, 3 (2010), 27:1–27:11.

[LXGF05] LI C., XU C., GUI C., FOX M. D.: Level set evolution without re-initialization: a new variational formulation. In *Computer Vision and Pattern Recognition (CVPR)* (2005), pp. 430–436, Vol. 1.

[OF01] OSHER S., FEDKIW R. P.: Level set methods: An overview and some recent results. *Journal of Computational Physics 169* (2001), 463–502.

[PBDSH13] PANOZZO D., BARAN I., DIAMANTI O., SORKINE-HORNUNG O.: Weighted averages on surfaces. *ACM Trans. Graph. 32* (2013), 60:1–60:12. ACM SIGGRAPH 2013.

[PHD*10] POTTMANN H., HUANG Q., DENG B., SCHIFTNER A., KILIAN M., GUIBAS L., WALLNER J.: Geodesic patterns. *ACM Transactions on Graphics 29*, 4 (2010), 43:1–43:10. Proc. SIGGRAPH 2010.

[RC13] ROUCHDY Y., COHEN L. D.: Geodesic voting methods: overview, extensions, and application to blood vessel segmentation. *Computer Methods in Biomechanics and Biomedical Engineering: Imaging and Visualization 1*, 2 (2013), 79–88.

[RLF09] RUSTAMOV R. M., LIPMAN Y., FUNKHOUSER T.: Interior distance using barycentric coordinates. *Computer Graphics Forum 28*, 5 (2009), 1279–1288. Symposium on Geometry Processing 2009 issue.

[RS13] ROGET B., SITARAMAN J.: Wall distance search algorithm using voxelized marching spheres. *Journal of Computational Physics 241* (2013), 76–94.

[RSST01] RVACHEV V. L., SHEIKO T. I., SHAPIRO V., TSUKANOV I.: Transfinite interpolation over implicitly defined sets. *Computer Aided Geometric Design 18* (2001), 195–220.

[Rus07] RUSTAMOV R. M.: *Boundary element formulation of harmonic coordinates*. Tech. rep., Department of Mathematics, Purdue University, November 2007.

[Rva74] RVACHEV V. L.: *Methods of Logic Algebra in Mathematical Physics*. Naukova Dumka, 1974. In Russian.

[Rva82] RVACHEV V. L.: *Theory of R-functions and Some Applications*. Naukova Dumka, 1982. In Russian.

[Sha07] SHAPIRO V.: Semi-analytic geometry with R-functions. *Acta Numerica 16* (2007), 239–303.

[She96] SHEWCHUK J. R.: Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator. In *Applied Computational Geometry: Towards Geometric Engineering*, vol. 1148 of *LNCS*. Springer, 1996, pp. 203–222.

[SOG14] SABELNIKOV V., OVSYANNIKOV A. Y., GOROKHOVSKI M.: Modified level set equation and its numerical assessment. *J. Comput. Phys.* (2014). http://dx.doi.org/10.1016/j.jcp.2014.08.018.

[Spa94] SPALDING D. B.: Calculation of turbulent heat transfer in cluttered spaces. In *Proc. 10th Int. Heat Transfer Conference* (Brighton, UK, 1994).

[SRGB14] SOLOMON J., RUSTAMOV R., GUIBAS L., BUTSCHER A.: Earth mover's distances on discrete surfaces. *ACM Transactions on Graphics 33*, 4 (2014), 67:1–67:12. Proc. SIGGRAPH 2014.

[Tau94] TAUBIN G.: Distance approximations for rasterizing implicit curves. *ACM Transactions on Graphics 13* (1994), 3–42.

[TLHD03] TONG Y., LOMBEYDA S., HIRANI A. N., DESBRUN M.: Discrete multiscale vector field decomposition. *ACM Trans. Graph. 22*, 3 (2003), 445–452. SIGGRAPH 2003.

[Tuc98] TUCKER P. G.: Assessment of geometric multilevel convergence and a wall distance method for flows with multiple internal boundaries. *Applied Mathematical Modelling 22* (1998), 293–311.

[Tuc11] TUCKER P. G.: Hybrid Hamilton-Jacobi-Poisson wall distance function model. *Computers & Fluids 44*, 1 (2011), 130–142.

[Tuc14] TUCKER P. G.: *Unsteady Computational Fluid Dynamics in Aeronautics*. Springer, 2014.

[Var67] VARADHAN S. R. S.: On the behavior of the fundamental solution of the heat equation with variable coefficients. *Comm. Pure Appl. Math. 20* (1967), 431–455.

[WBR14] WHYTOCK T., BELYAEV A., ROBERTSON N. M.: Dynamic distance-based shape features for gait recognition. *Journal of Mathematical Imaging and Visualization* (2014). In press.

[XQYH12] XIN S.-Q., QUYNH D. T. P., YING X., HE Y.: A global algorithm to compute defect-tolerant geodesic distance. In *SIGGRAPH Asia 2012 Technical Briefs* (2012), pp. 23:1–23:4.

[XT11] XIA H., TUCKER P. G.: Fast equal and biased distance fields for medial axis transform with meshing in mind. *Applied Mathematical Modelling 35* (2011), 5804–5819.

[XTC12] XIA H., TUCKER P. G., COUGHLIN G.: Novel applications of BEM based Poisson level set approach. *Engineering Analysis with Boundary Elements 36* (2012), 907–912.

[YWH13] YING X., WANG X., HE Y.: Saddle vertex graph (SVG): a novel solution to the discrete geodesic problem. *ACM Transactions on Graphics 32* (2013), 170.

[Zuc13] ZUCKER S. W.: Distance images and the enclosure field: applications in intermediate-level computer and biological vision. In *Innovations for Shape Analysis*. Springer, 2013, pp. 301–323.

Appendix

A. Normalization for *p*-Laplacian distances

Let us start from the Laplacian in the one-dimensional case and treat variable x as the distance from 0. Consider

$$-fu''(x) = 1, \qquad u(0) = 0, \qquad (49)$$

where we assume that f is constant. We have

$$u(x) = -\frac{1}{2f}(C_0 - x)^2 + C_1,$$

where C_0 and C_1 are constants. The condition $u(0) = 0$ implies that $C_1 = C_0^2/(2f)$. Thus

$$C_0 = fx + u', \quad C_1 = u + (u')^2/(2f)$$

and excluding C_0 and C_1 yields

$$fx^2 + 2u'x - 2u = 0.$$

Assuming that $x > 0$, we arrive at

$$x = -\frac{u'}{f} + \sqrt{\frac{(u')^2}{f^2} + \frac{2u}{f}} = \frac{2u}{u' + \sqrt{(u')^2 + 2fu}}. \quad (50)$$

A multidimensional analogue of the right-hand side of (50) is obtained by replacing u' by $|\nabla u|$ is given by

$$-\frac{|\nabla u|}{F} + \sqrt{\frac{|\nabla u|^2}{F^2} + \frac{2u}{F}} = \frac{2u}{|\nabla u| + \sqrt{|\nabla u|^2 + 2Fu}}, \quad (51)$$

where F is a properly chosen generalization of f from (49).

For example, setting F equal to the Frobenius norm of the Hessian of $u(\mathbf{x})$ yields Taubin normalization (29), while $F = 1$ gives Spalding-Tucker normalization (30).

Let us now assume that $f = 1$ in (49) and consider the one-dimensional *p*-Laplacian case

$$\left((u')^{p-1}\right)' = -1, \quad (u')^{p-1} = C_0 - x, \quad u' = (C_0 - x)^{\frac{1}{p-1}}.$$

Thus

$$u(x) = -\frac{p-1}{p}(C_0 - x)^{\frac{p}{p-1}} + C_1.$$

Now $u(0) = 0$ yields

$$-\frac{p-1}{p}(C_0)^{\frac{p}{p-1}} + C_1 = 0.$$

We also have

$$C_0 = x + (u')^{p-1}, \quad C_1 = u + \frac{p-1}{p}(u')^p.$$

Excluding C_0 and C_1 we arrive at

$$x + (u')^{p-1} = \left[\frac{p}{p-1}u + (u')^p\right]^{\frac{p-1}{p}}.$$

Therefore

$$x = -(u')^{p-1} + \left[\frac{p}{p-1}u + (u')^p\right]^{\frac{p-1}{p}}.$$

For $p = 2$ it gives (50) with $f = 1$.

The suggested normalization procedure for the multidimensional case is

$$-|\nabla u|^{p-1} + \left[\frac{p}{p-1}u + |\nabla u|^p\right]^{\frac{p-1}{p}}.$$
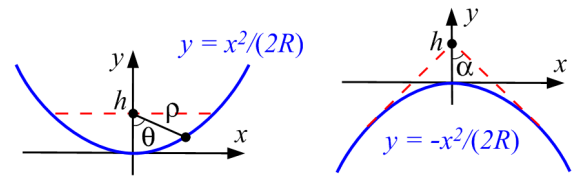
B. Asymptotics of L_p-distance fields near boundary

Figure 16: *Osculating parabolas.*

2-D case. Given a point $\mathbf{x} \in \Omega$ situated at the distance $h \ll 1$ from $\partial\Omega$, let us introduce Euclidean and polar coordinates, as shown in Fig. 16, where the origin of coordinates is located at the closest point to $\mathbf{x}$ on $\partial\Omega$ and the *y*-axis coincides with the direction of the orientation normal.

We start from the case of the positive curvature k of $\partial\Omega$ at

the origin of coordinates. Locally $\partial\Omega$ is approximated by the osculating parabola

$$y = x^2/(2R),$$

where $R = 1/k$ is the curvature radius, as seen in the left image of Fig. 16. In polar coordinates

$$x = \rho\sin\theta, \quad y = h - \rho\cos\theta$$

the parabola becomes

$$\rho^2\sin^2\theta + 2R\rho\cos\theta - 2Rh = 0.$$

Solving this quadratic equation for $\rho > 0$ yields

$$\rho = \frac{-R\cos\theta + \sqrt{D}}{\sin^2\theta}, \quad D = R^2\cos^2\theta + 2Rh\sin^2\theta$$

and we can write

$$\frac{1}{\rho} = \frac{1}{2Rh}\left(R\cos\theta\sqrt{1 + \frac{2h}{R}\tan^2\theta} - R\cos\theta\right)$$

$$= \frac{\cos\theta}{h} + \frac{1}{2R}\frac{\sin^2\theta}{\cos\theta} + O(h).$$

Therefore,

$$\Phi_p(\boldsymbol{x}) \sim \int_{-\pi/2}^{\pi/2}\frac{d\theta}{\rho(\theta)^p} = \frac{c_p}{h^p} + \frac{1}{R}\frac{d_p}{h^{p-1}} + O\left(\frac{1}{h^{p-2}}\right),$$

where

$$c_p = \int_{-\pi/2}^{\pi/2}\cos^p\theta\,d\theta \quad\text{and}\quad d_p = \frac{p}{2}\int_{-\pi/2}^{\pi/2}\cos^{p-2}\theta\sin^2\theta\,d\theta$$

Simple calculations show that $d_p = c_{p-2}/2$.

Above we have assumed that $p > 1$. If $p = 1$ (the case of the mean value coordinates) we have

$$\Phi_1(\boldsymbol{x}) = \int_{-\pi/2}^{\pi/2}\frac{d\theta}{\rho(\theta)} = \frac{2}{h} + \frac{1}{2R}\int_{-\pi/2}^{\pi/2}\frac{\sin^2\theta}{\cos\theta}\,d\theta + O(h)$$

and the integral diverges at $\pm\pi/2$. So we have to consider

$$\Phi_1(\boldsymbol{x}) \sim \frac{1}{2Rh}\int_{-\pi/2}^{\pi/2}\left(\sqrt{R^2\cos^2\theta + 2Rh\sin^2\theta} + R\cos\theta\right)d\theta$$

as $h \to 0$. Note that

$$\int_{-\pi/2}^{\pi/2}\sqrt{R^2\cos^2\theta + 2Rh\sin^2\theta}\,d\theta$$

$$= 2R\int_{0}^{\pi/2}\sqrt{1 - \left(1 - \frac{2h}{R}\right)\sin^2\theta}\,d\theta = 2R\mathcal{E}\left(1 - \frac{2h}{R}\right)$$

where

$$\mathcal{E}(t) = \int_{0}^{\pi/2}\sqrt{1 - t\sin^2\theta}\,d\theta$$

is the complete elliptic integral of the second kind. It can be shown that

$$\mathcal{E}(1 - \varepsilon) = 1 - \frac{1}{2}\varepsilon\ln\varepsilon + \dots \quad\text{as } \varepsilon \to 0.$$

Thus

$$\Phi_1(\boldsymbol{x}) \sim \frac{2}{h} + \frac{2}{R}\ln\frac{1}{h} + O(1), \quad\text{as } h \to 0.$$

Now let us consider the negative curvature case which corresponds to the right image of Fig. 16. We have

$$x = \rho\sin\theta, \qquad y = h - \rho\cos\theta, \qquad y = -x^2/(2R),$$

$$\rho^2\sin^2\theta - 2R\rho\cos\theta + 2Rh = 0,$$

$$\rho = \left(R\cos\theta - \sqrt{R^2\cos^2\theta - 2Rh\sin^2\theta}\right)\Big/\sin^2\theta,$$

$$\frac{1}{\rho} = \frac{1}{2Rh}\left(\sqrt{R^2\cos^2\theta - 2Rh\sin^2\theta} + R\cos\theta\right)$$

$$= \frac{1}{2Rh}\left(R\cos\theta\sqrt{1 - \frac{2h}{R}\tan^2\theta} + R\cos\theta\right)$$

$$= \frac{\cos\theta}{h} - \frac{1}{2R}\frac{\sin^2\theta}{\cos\theta} + O(h).$$

Thus we have arrived at

$$\frac{1}{\rho^p} = \frac{\cos^p\theta}{h^p} - \frac{p}{2R}\frac{\cos^{p-2}\theta\sin^2\theta}{h^{p-1}} + O\left(\frac{1}{h^{p-2}}\right). \quad (52)$$

Now let us study an asymptotic behavior of

$$\int_{-\alpha(h)}^{\alpha(h)}\frac{d\theta}{\rho(\theta)^p},$$

where the integration limits $\alpha(h)$ and $-\alpha(h)$ correspond to the two rays originated from $\boldsymbol{x}$ and tangent to the osculating parabola $y = -x^2/(2R)$, as seen in the right image of Fig. 16. Note that $\alpha(h) = \pi/2 + O(h)$ and $\cos\theta = O(h)$ for $|\theta|$ between $\alpha(h)$ and $\pi/2$, as $h \to 0$. Thus, in view of (52), we have

$$\Phi_p(\boldsymbol{x}) \sim \int_{-\alpha}^{\alpha}\frac{d\theta}{\rho(\theta)^p} = \frac{1}{h^p}\int_{-\alpha}^{\alpha}\cos^p\theta\,d\theta$$

$$- \frac{p}{2Rh^{p-1}}\int_{-\alpha}^{\alpha}\cos^{p-2}\theta\sin^2\theta\,d\theta + O\left(\frac{1}{h^{p-2}}\right)$$

$$= \frac{1}{h^p}\int_{-\pi/2}^{\pi/2}\cos^p\theta\,d\theta - \frac{p}{2Rh^{p-1}}\int_{-\pi/2}^{\pi/2}\cos^{p-2}\theta\sin^2\theta\,d\theta$$

$$+ O\left(\frac{1}{h^{p-2}}\right) = \frac{c_p}{h^p} - \frac{1}{R}\frac{d_p}{h^{p-1}} + O\left(\frac{1}{h^{p-2}}\right)$$

with the same c_p and d_p as for the positive curvature case.

3-D case. Let us introduce spherical coordinates, as shown in the right image of Fig. 17.
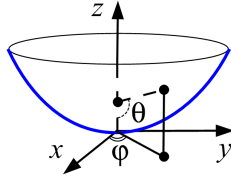
$$x = \rho\sin\theta\cos\varphi, \quad y = \rho\sin\theta\sin\varphi, \quad z = h - \rho\cos\theta.$$

Locally a smooth surface is approximated by a paraboloid

$$z = \frac{1}{2}\left(ax^2 + by^2\right).$$

In the spherical coordinates, the paraboloid is given by

$$2(h - \rho\cos\theta) = \rho^2\left(a\sin^2\theta\cos^2\varphi + b\sin^2\theta\sin^2\varphi\right).$$

Figure 17: *Osculating paraboloid.*

Let us denote by $k(\varphi)$ the directional curvature at the origin of coordinates

$$k(\varphi) = a\cos^2\varphi + b\sin^2\varphi \equiv 1/R(\varphi),$$

then the surface equation is simplified to

$$\rho^2\sin^2\theta + 2\rho R(\varphi)\cos\theta - 2hR(\varphi) = 0.$$

Thus $1/\rho(\theta,\varphi)^p$ is given by

$$\frac{1}{[2hR(\varphi)]^p}\left[\sqrt{R(\varphi)^2\cos^2\theta + 2R(\varphi)h\sin^2\theta} + R(\varphi)\cos\theta\right]^p$$

So, similar to the 2D case, we arrive at

$$\Phi_p(\boldsymbol{x}) \sim \int_S \frac{d\Omega}{\rho(\theta,\varphi)^p} = \frac{c_p}{h^p} + H\frac{d_p}{h^{p-1}} + O\left(\frac{1}{h^{p-2}}\right)$$

where

$$H = \frac{1}{2\pi}\int_0^{2\pi}\frac{d\varphi}{R(\varphi)}$$

is the mean curvature and

$$c_p = \frac{1}{2}\int_\Sigma \cos^p\theta\,d\Sigma \quad\text{with}\quad d\Sigma = \sin\theta\,d\theta\,d\varphi.$$

In particular,

$$c_1 = \int_0^{2\pi} d\varphi \int_{-\pi/2}^{\pi/2}\cos\theta\sin\theta\,d\theta = \pi.$$