

THESIS PROPOSAL

**Robust Algorithms for Winding Numbers
and Signed Distance**

Nicole Feng

October 2025

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:

Keenan Crane (CMU), Chair
Nancy Pollard (CMU)
Ioannis Gkioulekas (CMU)
Christopher Wojtan (ISTA)

*Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.*

Abstract

This thesis presents robust algorithms for inside-outside computation and curve reconstruction (via *winding numbers*) and signed distance computation. These algorithms make geometric inferences from imperfect data, where such imperfect data includes noisy, incomplete, or inaccurate observations or representations of shapes that result from either acquisition or authoring of geometry. A theme is that robustness and versatility can often be achieved by processing smooth, globally-defined functions encoding the geometry of interest, that are more amenable to robust computation than the original, defective curve or surface. For both inside-outside and signed distance computation we can unlock further control over geometry and topology by processing higher-order derivatives of these functions. In many cases, we can also re-cast our algorithms, formulated in terms of smooth functions, onto different discretizations and geometric data structures. Another theme is that robust reconstruction and robust signed distance computation are closely related problems; towards this end, we provide a formalization of their relationship that justifies the design of our algorithms.

Contents

1	Introduction	1
2	Background	3
2.1	Notation	3
2.2	Preliminaries	4
2.3	The inside and outside of curves and surfaces	7
2.3.1	Curve orientations	7
2.3.2	Homology & cohomology	8
2.3.3	Winding numbers	9
2.4	Geodesic distance	10
3	Inside-outside	13
3.1	Winding numbers as a jump harmonic function	13
3.2	Derivative processing	14
3.2.1	Differentiating and integrating jump harmonic functions	15
3.3	Results	16
4	From inside-outside to signed distance	23
4.1	A unification of winding numbers, Poisson Surface Reconstruction, and signed distance	24
4.2	Convolutional distance approximations	27
4.3	Fundamental limitations of convolutional distance	29
4.4	Relation to kernel methods	32
5	Signed distance	35
5.1	The asymptotics of vector diffusion	35
5.2	Algorithm	38
5.3	Results	40
6	Proposed Work	45
6.1	Instant generalized signed distance	45
6.2	Timeline	47

7	Bibliography	49
A	The signed Hopf-Cole transformation	59
A.1	From the (signed) eikonal equation to a (jump) screened Laplace equation . . .	59
A.2	Analysis of the screened Laplace equation as a boundary-layer problem	63

CHAPTER 1

Introduction

From natural phenomena (material structure, plant growth, geological formation, etc.) to human-generated data (manufactured objects, art, digital assets), our lives are defined by geometry. Algorithms for geometric problems enable modeling, simulation, and performance analysis, and thus are key to successfully manipulating the world around us; these algorithms drive technological progress across numerous domains including engineering, animation, finite-element analysis, and other design-based industries.

In general, problems involving geometry can be hard to solve, not necessarily because the problem itself is difficult to model or understand, but because of the difficulty in doing computation with geometry. Virtually all geometric data suffers quality issues, due to imperfect acquisition, reconstruction, or modeling, and unfortunately standard geometric operations are often built on assumptions that break down as soon as the input geometry fails to satisfy “ideal” standards. In practice, geometry can and does suffer from missing data, noise, self-intersections, non-manifold features, and myriad other defects. These low-level defects are unpredictable and easily frustrate higher-level design and optimization tasks such as reliable physics analysis or machine learning pipelines that demand perfect, clean data. Explicit repair of data is possible but time-consuming and tedious, with no guarantees of perfection; we instead need algorithms that remain stable under perturbations in their input and hence allow direct computation of geometric quantities from defective data, which requires answering fundamental questions about geometry.

In other words, we need *robust geometry processing*: we need versatile algorithms that work reliably across varying degrees of quality in their input. This thesis addresses the robust computation of two fundamental geometric quantities, inside-outside and signed distance, that underlie key problems in design and engineering.

- Chapter 3 addresses the question of what it means to be “inside” or “outside” a curve or surface, and uses *winding numbers* to compute inside-outside on general surface domains of arbitrary topology.
- Chapter 4 extends the theory underpinning winding numbers to signed distance computation, which outlines fundamental tradeoffs to be made by robust algorithms for signed

distance.

- Chapter 5 describes a robust algorithm for computing signed distance to defective curves and surfaces in 2D and 3D.

Finally, Chapter 6 proposes a project extending robust signed distance computation to pointwise queries, and my planned timeline for graduation.

CHAPTER 2

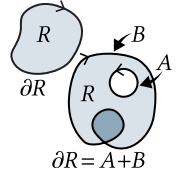
Background

This chapter presents mathematical background for understanding inside-outside classification and signed distance computation, the two fundamental geometric quantities that form the core of this thesis.

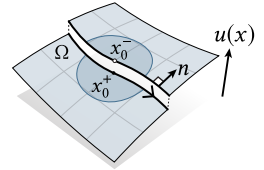
2.1 Notation

We use $|\cdot|$ and $\langle \cdot, \cdot \rangle$ to denote the standard Euclidean norm and inner product for vectors in $\mathbb{R}^2$ and $\mathbb{R}^3$. We use $J : \mathbb{R}^2 \rightarrow \mathbb{R}^2; (x, y) \mapsto (-y, x)$ to denote a quarter turn in the counter-clockwise direction. For any two vectors $u, v \in \mathbb{R}^2$, we define a scalar-valued cross product $u \times v := u_1 v_2 - u_2 v_1$; note that $\langle Ju, v \rangle = u \times v$. For any function $f(t)$ of a single parameter t , we let $\dot{f}(t) := \frac{d}{dt}f(t)$. We use Δ to denote the negative-semidefinite *Laplace-Beltrami* operator on M , which locally behaves like the ordinary Laplace operator $\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$. A function $u : M \rightarrow \mathbb{R}$ is *harmonic* if it is in the kernel of the Laplacian, *i.e.*, if $\Delta u = 0$.

Throughout we use Ω to denote a curve or surface, though we also use Γ to specifically denote a 1D curve. We also frequently discuss the boundaries of regions. Intuitively, the boundary ∂R of a 3D region R is the oriented surface enclosing R , and may have multiple components. Similarly, the boundary of a surface is the oriented curve enclosing the surface, and the boundary of a curve is its set of oriented endpoints. We call Ω *closed* if $\partial\Omega = \emptyset$.

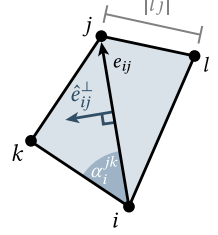


One-sided limits. We make frequent use of signed functions whose value depends on from which side we approach a given curve or surface Ω . We denote one-sided limits of such functions as follows. On $\mathbb{R}^d$, we let $x^\pm := \lim_{s \rightarrow 0} x \pm sn(x)$ denote a point on $x \in \Omega$ as approached from the positive or negative side of Ω , where $n(x)$ is the outward-pointing normal of $\Omega \subset \mathbb{R}^d$ at x that points from the negative to positive side of Ω (see inset). We then denote the corresponding one-sided limits of a function $u(x)$ as $u^\pm(x) := u(x^\pm) := \lim_{s \rightarrow 0} u(x \pm sn(x))$.



For functions defined on smooth manifolds (Section 2.2), an analogous definition holds where the limits are instead taken under an appropriate chart that maps from the surface to $\mathbb{R}^2$.

Triangle meshes. We often represent a 2D surface domain in the discrete setting as a triangle mesh $M = (V, E, F)$, with no restrictions on connectivity; we use C for the set of all triangle corners, and ∂M to denote the boundary of M . We denote k -simplices by $(k+1)$ -tuples of vertex indices, *i.e.*, vertices $i \in V$, edges $ij \in E$, and faces $ijk \in F$. Likewise, we denote the corner of triangle ijk at vertex i as $j_i^k \in C$. These indices are also used to express quantities stored on mesh elements — for instance, corner angles are denoted by α_i^{jk} . We use $|\cdot|$ to denote the volume of a simplex — for example, $|ij|$ is an edge length and $|ijk|$ is a triangle area. For brevity, we often assume that any interior, manifold, oriented edge ij is contained in two triangles labeled ijk , jil , where k and l sit to the left and right of ij , *resp.*



We use $<$ and $>$ to indicate summation over all elements contained by or containing another element (*resp.*). For instance $\sum_{ijk > ij}$ sums over all triangles ijk containing edge ij . For each edge ij , we let e_{ij} be a vector parallel to the edge with arbitrary (but fixed) orientation, and magnitude equal to the edge length. We let $e_{ij}^\perp$ be the 90° rotation of e_{ij} in the counter-clockwise direction, and use $\hat{e}_{ij}$, $\hat{e}_{ij}^\perp$ for the corresponding unit vectors.

2.2 Preliminaries

This section establishes the differential geometric foundations necessary for understanding the geometry processing algorithms developed in subsequent chapters. We focus on the key concepts of smooth manifolds and differential forms, which enable our function-based approach to geometry processing. Here we give only a high-level overview of the concepts needed to understand the algorithms in this thesis; for thorough treatment of these concepts, see Lee [2012] or do Carmo [1992].

Manifolds. Manifolds are topological spaces that locally look like $\mathbb{R}^n$. In particular, a *topological manifold* M has enough topological structure to allow uniqueness of limits and a meaningful notion of function continuity; M is also “locally Euclidean” in the sense that each point $p \in M$ lies in an open subset $V \subset M$ homeomorphic to an open subset of $V \subset \mathbb{R}^n$, *i.e.* there exists a continuous bijection $\varphi : U \rightarrow V$ with continuous inverse.

These homeomorphisms are called *coordinate charts* and let one locally analyze manifolds by comparing pieces of M with pieces of the more well-understood space $\mathbb{R}^n$. For example, one might think to call a function $f : M \rightarrow \mathbb{R}$ differentiable at $p \in M$ if its image under a coordinate chart is differentiable in $\mathbb{R}^n$. For purely topological manifolds, however, this notion of differentiability depends on the particular choice of coordinate chart. Thus we consider *smooth manifolds*, topological manifolds that can be covered with a collection of charts such that any two charts are *smoothly compatible*: that is, for any two charts $\varphi_\alpha, \varphi_\beta$ such that $\varphi_\alpha(U_\alpha) \cap \varphi_\beta(U_\beta) \neq \emptyset$, the transition map $\psi^{-1} \circ \varphi \circ \varphi^{-1}$ is a smooth diffeomorphism. This compatibility ensures that

the notion of differentiability is well-defined, in the sense that the differentiability of functions does not depend on the choice of coordinate chart, so calculus can be done consistently on the manifold.

Tangent spaces and vector fields. Intuitively, *tangent vectors* are vectors that “lie flat” along a smooth manifold M . In more detail, each point $p \in M$ can be associated with a *tangent space* $T_p M$, an n -dimensional vector space isomorphic to $\mathbb{R}^n$ that captures all possible directions of motion through p . Tangent vectors can be characterized, for example, as equivalence classes of smooth curves through p , where two curves γ_1, γ_2 are equivalent if they have the same tangent vectors at a given point when pulled back to pieces of the plane. Because the smooth structure of M means derivatives of curves on M are well-defined, we can define tangent vectors on M .

The *tangent bundle* TM of M takes a union of all tangent spaces on M , and has a smooth structure inherited from the smooth structure of M . A *smooth vector field* can then be defined as a *section* of TM , meaning a smooth map $X : M \rightarrow TM$, where X assigns to each point $p \in M$ a tangent vector $X(p) \in T_p M$ such that the assignment varies smoothly.

Differential forms. Differential forms provide a framework for integration on manifolds, and are built up of objects called *covectors* that act as “meter sticks” with which to measure the signed length of tangent vectors along certain directions.

In more detail, given an n -dimensional vector space V , a *covector* is a linear function $\omega : V \rightarrow \mathbb{R}$, and the space of all covectors forms a dual space V^* to V . Each point $p \in M$ of a smooth manifold can be associated with a *cotangent space* $T_p^* M := (T_p M)^*$ which consists of linear functionals $T_p M \rightarrow \mathbb{R}$, and the collection of all cotangent spaces on M is called the *cotangent bundle* $T^* M$; a smooth section of $T^* M$ is called a smooth *covector field* or *1-form*. A 1-form ω can be used to integrate along a curve $\gamma : [a, b] \rightarrow M$ using $\int_\gamma \omega = \int_a^b \omega_{\gamma(t)}(\gamma'(t)) dt$.

More generally, one can define (covariant) *k-tensors*, multilinear functions that take in k vectors, and define *k-covectors* as alternating *k-tensors*; “alternating” means the *k-tensor* is negated when two arguments are swapped. At each $p \in M$, one can consider the space $\Lambda^k(T_p^* M)$ of all *k-tangent covectors*, and consider a union of all such spaces to form the vector bundle $\Lambda^k(T^* M) = \coprod_{p \in M} \Lambda^k(T_p^* M)$; a section of $\Lambda^k(T^* M)$ is called a (differential) *k-form*. In words, a *k-form*, evaluated at $p \in M$, measures signed *k-dimensional* volumes in *k-dimensional* linear subspaces of $T_p M$.

The integration of differential forms on M also depends on the *orientability* of M : the manifold M is *orientable* if it has an atlas of coordinate charts whose transition maps all have positive Jacobian determinant. Classic examples of non-orientable manifolds include the Möbius strip and Klein bottle, where attempting to define a consistent orientation while traversing certain loops leads to contradictions.

The *exterior derivative* d acts on *k-forms* to produce $(k + 1)$ -forms, and generalizes the differential of a function (0-form). On a smooth manifold M , the exterior derivative is the unique differential operator satisfying several fundamental properties, such as linearity over $\mathbb{R}$ and $d \circ d = 0$. The latter property, nilpotency, gives rise to the distinction between closed forms

($d\omega = 0$) and exact forms ($\omega = d\alpha$), which is central to de Rham cohomology (Section 2.3.2).

The fundamental relationship between differentiation and integration is captured by *Stokes' theorem*: for an oriented smooth n -manifold M with boundary and $(n - 1)$ -form ω with compact support, $\int_{\partial M} \omega = \int_M d\omega$.

Hodge decomposition. Throughout we make use of *Hodge decomposition* acting on 1-forms. In particular, any 1-form ω on a closed Riemannian manifold can be uniquely decomposed into a sum of an exact component $d\alpha$, coexact component $\delta\beta$, and a harmonic component γ ,

$$\omega = d\alpha + \delta\beta + \gamma.$$

On manifolds with boundary, the *Hodge-Friedrichs-Morrey decomposition* applies [Schwarz 2006]. As detailed in Crane et al. [2013a, Chapter 8], this decomposition can be computed by solving a pair of Poisson equations

$$\Delta_0 \alpha = \delta_1 \omega \quad \text{and} \quad \Delta_2 \beta = d_1 \omega, \quad (2.1)$$

where $\Delta_0 := *^{-1}_0 d_0^T *_1 d_0$ and $\Delta_2 := d_1 *_1^{-1} d_1^T *_2$ are the discrete 0- and 2-form Laplacians, *resp.*, with their usual zero-Neumann boundary conditions.

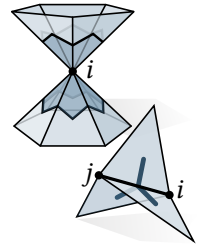
Hodge decomposition is a generalization of *Helmholtz decomposition* from vector calculus, and implies that solving for the scalar potential f that gives the best least-squares approximation of a given vector field X ,

$$f := \operatorname{argmin}_{f'} \|\nabla f' - X\|^2,$$

has a residual $\nabla f - X$ that consists of a most a coexact plus a harmonic component.

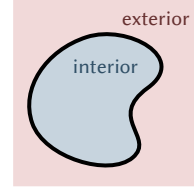
Discrete differential forms. On triangle meshes, we use *discrete exterior calculus* to represent differential forms on simplicial complexes [Desbrun et al. 2006; Crane et al. 2013a]. On triangle meshes, k -forms are represented as values assigned to k -dimensional mesh elements: 0-forms correspond to values on vertices, 1-forms on edges, and 2-forms on faces. The *discrete exterior derivative* d_k maps discrete k -forms to discrete $(k + 1)$ -forms, and can be represented as a sparse matrix equal to the transpose of the boundary operator. The *discrete Hodge star* $\star_k$ takes k -forms to $(n - k)$ -forms, incorporating geometric information arising from edge lengths. These discrete operators preserve the fundamental structure of their smooth counterparts: for example, $d_{k+1} \circ d_k = 0$, and discrete versions of Stokes' theorem hold.

For general, possibly nonmanifold triangle meshes, we follow Sharp et al. [2019a] and define the discrete Hodge star operators by taking volume ratios involving all incident elements (see inset), yielding diagonal matrices with entries $(\star_0)_i := \frac{1}{3} \sum_{ijk \in F} |ijk|$ for all $i \in V$, $(\star_1)_{ij} := w_{ij}$ for all $ij \in E$, where $w_{ij} := \frac{1}{2} \sum_{ijk \in F} \cot \alpha_k^{ij}$ are *cotan weights* [MacNeal 1949, Section 3.2], and $(\star_2)_{ijk} := 1/|ijk|$ for all $ijk \in F$. Otherwise, we use the standard discrete exterior derivative matrices d_k ; the *discrete codifferential* is then $\delta_k := \star_{k-1}^{-1} d_{k-1}^T \star_k$.



2.3 The inside and outside of curves and surfaces

Intuitively, closed objects — such as a capped bottle, soccer ball, or sealed building — separate space into an “inside” region comprised of all the points which cannot be connected to the “outside” without intersecting the object. This intuition can be formalized, for instance using the *Jordan curve theorem* which says that every simple closed curve in the plane — that is, every non-self-intersecting continuous loop — indeed decomposes $\mathbb{R}^2$ into two connected components, a region R bounded by the curve, called the “interior”, and its unbounded complement $\mathbb{R}^2 \setminus R$, called the “exterior” of the curve. The *Jordan–Brouwer separation theorem* gives an analogous statement in higher dimensions.

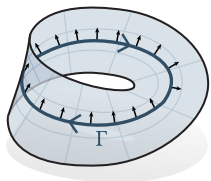
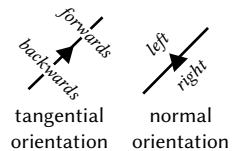


Determining the interior (or equivalently, determining the exterior) of a curve or surface is a fundamental problem in computer vision and computer graphics. For example, *surface reconstruction* is a perennial task that aims to infer, or reconstruct, the surface from which a collection of discrete points or polygons were sampled, whose output is a well-defined interior and exterior of a shape [Berger et al. 2014]. Other applications that depend on reliable interior/exterior computation include motion planning, computer-aided design, geographic information systems.

The interior and exterior of a shape is often represented as a scalar function, such as an *occupancy function*, which typically has value 1 inside the shape and 0 outside. In Section 2.3.3 and Chapter 3, we consider the more general *winding number function*, which gives not just a binary inside/outside classification, but rather an integer-valued function that counts how many times a curve or surface encloses a given point.

2.3.1 Curve orientations

For a surface embedded in $\mathbb{R}^d$, we can specify its orientation as a choice of a continuously-varying surface normal $n(x)$ at every point. For curves, one can also specify orientation via normal direction, picking “left” and “right” sides of the curve. We can also orient along the curve’s tangent direction, picking forward and backward directions (inset). For curves on an oriented surface, the two notions of orientation are equivalent: the normal of the curve, and hence the left/right sides, is usually defined via a 90° counterclockwise rotation of the tangent.



However, if M is non-orientable, then one cannot pick a consistent counterclockwise direction at all points of the surface, and the two types of orientation are not equivalent: for example, the inset shows a curve on a Möbius strip that cannot be given a consistent normal orientation, but can be given a consistent tangential orientation.

2.3.2 Homology & cohomology

On domains besides $\mathbb{R}^d$, a closed curve or surface may not necessarily be the boundary of a well-defined region (inset). The failure of a curve to be a boundary of a region can be studied using *homology* and *cohomology* (Munkres [1984, Chs. 1 & 5]).

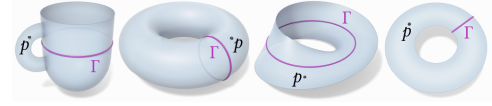
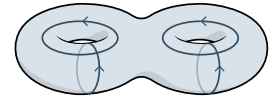


Fig. 2.1: Is the point p “inside” or “outside” the curve Γ ? On surfaces, this question does not always have a meaningful answer.

Homology. On orientable surfaces without boundary, two closed curves Γ_1 and Γ_2 are *homologous* if they form the boundary of a region R , meaning if $\Gamma_1 - \Gamma_2 = \partial R$ (inset). A curve that is the boundary of a region is called *nullhomologous*. In the plane, all closed curves are nullhomologous and hence have well-defined insides and outsides. However, on surfaces or even in subsets of the plane, there may be closed curves which are not the boundary of any region; that is, these curves are congruent to zero in the homology group $H_1(M) = \ker(\partial_1) \setminus \text{Im}(\partial_2)$. For clarity, we call a closed curve *bounding* if it is nullhomologous, and *nonbounding* if it is non-nullhomologous (a.k.a. a *separating cycle*.) Even if individual loops do not bound regions, however, they can still conspire to define a meaningful partition — see the right inset, which reproduces an example from Riso et al. [2022, Figure 4].

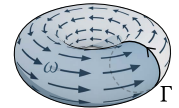


The *homology class* of Γ is the set of all curves homologous to Γ . A set of *homology generators* for M is a set of closed curves $\{\eta_i\}$ from which we can construct a curve in every homology class. On a closed surface of genus g there are $2g$ homology generators, which can be organized in pairs around each handle (inset).



Cohomology. Alternatively, we can study curves and their bounding properties by studying 1-forms dual to the curves. In *de Rham cohomology*, curves Γ are replaced by 1-forms ω , and the boundary operator is replaced by the *exterior derivative* d . We say that a 1-form ω is *closed* if $d\omega = 0$. Two closed 1-forms ω_1, ω_2 are said to be *cohomologous* if $\omega_1 - \omega_2 = d\alpha$, mirroring the condition for curves. Whereas the *Poincaré lemma* states that all 1-forms in $\mathbb{R}^2$ are cohomologous to zero, there can be multiple cohomology classes on surfaces. A set of *cohomology generators* for M is a collection of 1-forms $\{\omega_i\}$ allowing us to represent the cohomology class of any exact 1-form ξ via a sum of generators. On a closed surface of genus g , there are $2g$ cohomology generators, matching the number of homology generators.

A 1-form ω can be paired with a curve Γ via integration, yielding a value $\int_{\Gamma} \omega$. Using this pairing, a 1-form ω is *dual* to a closed curve Γ if integration against ω counts intersections with Γ , i.e. if $\int_{\Gamma} \omega$ is the signed number of intersections between Γ and Γ' for any closed Γ' .

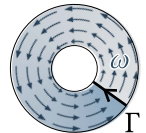
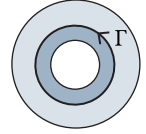


Cohomology is closely related to the theory of harmonic functions and differential forms. In particular, there is a unique harmonic form in each cohomology class, allowing us to represent cohomology classes concretely using harmonic forms. We make use of this duality in Chapter 3.

Relative homology & cohomology. So far we have only discussed surfaces M with no boundary, in which case nonbounding (non-nullhomologous) curves are categorized by the *absolute homology group* $H_1(\Omega)$ [Erickson and Whittlesey 2005].

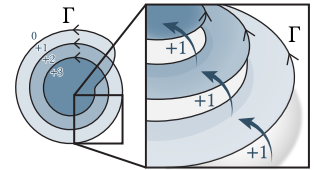
On surfaces with boundary, however, curves may be nonbounding in the sense that they separate the domain into two components, but cannot be represented as the boundary of some region in an absolute sense. For instance, the annulus has a single homology generator: a loop Γ wrapping around the hole in the middle. While Γ separates the annulus into two components, it is not the boundary of any region of the annulus since the boundary of e.g. the inside component includes the inner circle of the annulus' boundary in addition to Γ . On surfaces with boundary, nonseparating cycles are instead described by the *relative homology group* $H_1(M, \partial M)$, whose elements are closed loops in the space obtained by collapsing all of ∂M to a single point [Munkres 1984, §9].

On surfaces with boundary, one must also distinguish between absolute and relative cohomology. The absolute cohomology group $H^1(\Omega)$ consists of harmonic 1-forms tangent to the boundary, whereas the relative cohomology group $H^1(\Omega, \partial\Omega)$ consists of harmonic 1-forms normal to the boundary [Poelke and Polthier 2016]. The dual of a relative homology generator is an absolute cohomology generator and vice versa; for example, the dual of the nonseparating relative homology generator is the 1-form which circulates around the annulus, tangent to the boundary (inset).



2.3.3 Winding numbers

Winding numbers are a basic concept from differential geometry that give a natural extension of the binary notion of inside-outside to more than two regions [Do Carmo 2016, Section 5.7]. In the plane, winding number is a piecewise constant function that jumps by +1 as one crosses the curve from the right (Figure 2.2, inset). The classic winding number is a special case of the signed *solid angle* function, which is itself a particular *harmonic function*, i.e.,



winding number $\subset$ solid angle $\subset$ harmonic functions.

Connections between winding numbers, solid angles, and harmonic functions have long appeared in mathematics, physics, and scientific computing [Binysh and Alexander 2018]. Both Euler [1781] and Lagrange [1798] give formulas for the solid angle of a triangle; Gauss [1838, Sections 37-38] notes the relationship of solid angle to magnetic potential; Maxwell [1881, Articles 409-11, 417-21] further makes connections to jump conditions. Methods for approximating solid angles also play an integral role in *boundary element methods* (BEM) for the Laplace equation [Ning et al. 2010].

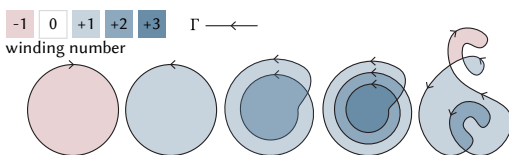


Fig. 2.2: For curves Γ in the plane, the winding number function $w_\Gamma(p)$ gives the number of times the curve Γ wraps around any given point p .

In computer graphics, winding numbers were first applied to *point-in-polygon queries* [Shimrat 1962; Haines 1994a]. Solid angle also plays a key role in rendering, e.g., for finite element radiosity [Goral

et al. 1984] or importance sampling for direct illumination [Veach and Guibas 1995, Section 2.1]. In geometry processing, the utility of the solid angle function for broken geometry has been rediscovered twice, via both *Poisson surface reconstruction (PSR)* [Kazhdan et al. 2006] and the *generalized winding number (GWN)* [Jacobson et al. 2013]. These methods are in turn key components of a wide variety of applications [Hu et al. 2018; Zhou et al. 2016; Chi and Song 2021; Müller et al. 2021; Dvořák et al. 2022; Collet et al. 2015; Chang et al. 2017].

Owing to their rich history and prevalence in math and physics, winding numbers can be characterized through several different perspectives [Feng et al. 2023]. These perspectives do not, however, have a standard extension to curves on surfaces, due to the possible presence of nonbounding loops (Section 2.3.2). Several authors have instead considered generalizations of *turning number* on surfaces, a quantity distinct from winding number despite the historical confusion of terminology [Reinhart 1960, 1963; Chillingworth 1972; Humphries and Johnson 1989; McIntyre and Cairns 1993]. McIntyre and Cairns [1993, Lemma 2] does describe a function that behaves like the winding number for bounding curves, but for nonbounding curves must introduce arbitrary discontinuities to keep this function piecewise constant, and Chernov and Rudyak [2009] define a so-called *affine winding number* useful only for curves within a common homotopy class. More recently, Riso et al. [2022] give a method for computing winding numbers but only on perfectly closed curves already partitioned into distinct loops. In Chapter 3, we give an algorithm that gives a well-behaved generalization of winding number, given minimal structure on the input curves.

2.4 Geodesic distance

Distance computation on curved spaces requires additional geometric structure beyond the purely topological notion of manifolds. We consider an n -dimensional *Riemannian manifold* (M, g) with metric g , which for each $p \in M$ gives rise to the norm of tangent vectors $v \in T_p M$ through $\|v\|_g := g_p(v, v)^{1/2}$, and angle between two tangent vectors $v, w \in T_p M$ through $\cos \theta = \frac{g_p(v, w)}{\|v\|_g \|w\|_g}$.

The metric lets us measure lengths of curves: for a (piecewise) smooth curve $\gamma : [a, b] \rightarrow M$, its length is

$$L_g(\gamma) = \int_a^b \|\gamma'(t)\|_g dt$$

(This definition is independent of parameterization.) The *Riemannian distance* between points $p, q \in M$ is then defined as the infimum of lengths over all curves connecting p and q . (Note that “distance” is also almost always used to refer to minimum distance, a convention we will also adopt throughout this thesis unless otherwise noted.) Curves that are locally length-minimizing are called *geodesics*. In $\mathbb{R}^n$, geodesics are straight lines; geodesics generalize straight lines to arbitrary Riemannian manifolds.

Parallel transport. Geodesics can also be defined to be curves whose tangent vectors remain parallel to the curve as they are transported along the curve. To define what it means for vectors to “remain parallel” across different tangent spaces, we need the concept of a *connection*, which provides a way to differentiate vector fields along curves. The *covariant derivative* generalizes the notion of a directional derivative to curved manifolds, and an *affine connection* ∇ on M assigns to a pair of vector fields X, Y a new vector field $\nabla_X Y$ that gives the covariant derivative of Y in the direction X . The fundamental theorem of Riemannian geometry states that given a Riemannian metric g , there exists unique affine connection on M , the *Levi-Civita connection*, that is both *metric-compatible* ($\nabla g = 0$) and *torsion-free*.

The connection enables definition of *parallel transport* along curves: given a curve $\gamma : I \rightarrow M$ and initial vector $V_0 \in T_{\gamma(t_0)}M$, there exists a unique vector field V along γ such that $V(t_0) = V_0$ and $\nabla_{\gamma'} V = 0$. Parallel transport of V along γ maps tangent vectors in $T_{\gamma(t_0)}M$ to tangent vectors in $T_{\gamma(t_1)}M$. A geodesic is a curve γ whose velocity vector field γ' is parallel-transported along itself, meaning $\nabla_{\gamma'} \gamma' = 0$.

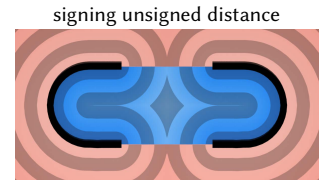
Signed distance. Consider a codimension-1 submanifold $\Omega \subset M$ (for example, curves on a surface, or surfaces within a volume). If Ω separates M into an interior A and an exterior $M \setminus A$, we can define a *signed distance function* (SDF) $\phi : M \rightarrow \mathbb{R}$ as

$$\phi(x) = \begin{cases} d_g(x, \Omega) & x \in M \setminus A \\ -d_g(x, \Omega) & x \in A \end{cases}$$

where the sign of $\phi(x)$ indicates whether x is in the interior or exterior of Ω . SDFs hence encode both geometric information (how far is x from Ω) and topological information (on which side of Ω lies x) in a single scalar function.

Signed distance is essential to many problems across graphics and vision, forming a basic component of numerous algorithms from geometric modeling [Museth et al. 2002], physical simulation [Osher et al. 2004], rendering [Quilez 2008], path planning [Oleynikova et al. 2016], geometric learning [Yariv et al. 2023] and computer vision [Vicini et al. 2022]. For watertight geometry in $\mathbb{R}^n$, there are many ways to compute signed distance: for example, unsigned distance can first be computed via fast, exact closest point queries [Sawhney et al. 2020], then signed via basic inside-outside tests like ray shooting [Haines 1994b]. Alternatively, one can sample geometry onto a grid and use methods like fast sweeping [Osher et al. 2004]; a few methods consider unsigned distance to curves embedded in surfaces [Bommes and Kobbelt 2007; Trettner et al. 2021], which can then be signed using, e.g., flood fill.

For non-watertight, noisy, self-intersecting, or otherwise broken geometry, the operations of computing distance and signing no longer commute; for one, simply signing unsigned distance can yield a function quite different from the SDF to completed geometry (inset). Likewise, wavefront-based methods like *fast marching* [Kimmel and Sethian 1998] and learning-based variants [Lichtenstein et al. 2019; Huberman et al. 2023] propagate sign errors. Some past works consider regularized signed distance for broken geometry, though they suffer from various



downsides that affect their accuracy, robustness, or generality [Bærentzen 2005; Mullen et al. 2010; Calakli and Taubin 2011; Xu and Barbič 2014; Brunton and Rmaileh 2021]. In Chapter 5, we present an algorithm for signed distance that is robust to broken geometry, and generalizes to curved surfaces and alternative spatial discretizations.

CHAPTER 3

Inside-outside

As introduced in Section 2.3.3, the winding number is the number of times a curve or surface wraps around a given point. Winding numbers are a basic component of geometric algorithms such as point-in-polygon tests, and their generalization to data with noise or topological errors has proven valuable for geometry processing tasks ranging from surface reconstruction to mesh booleans. However, standard definitions on $\mathbb{R}^d$ do not immediately apply on surfaces, where not all curves bound regions. Here, we develop a meaningful generalization, starting with the well-known relationship between winding numbers and harmonic functions. By processing the derivatives of such functions, we can robustly filter out components of the input that do not bound any region. The key idea of our method is to turn the difficult problem of determining inside/outside of an unstructured collection of broken curves, into an easier but equivalent problem about vector fields.

Ultimately, our algorithm yields (i) a closed, completed version of the input curves, (ii) integer labels for regions that are meaningfully bounded by these curves, and (iii) the complementary curves that do not bound any region. Because we work with smooth functions defined globally on the domain, our algorithm is much more robust than if we had tried to work directly with sparse, singular curves. The algorithm is guaranteed to work if the input is “perfect” (no noise, gaps, etc.), and otherwise degrades gracefully in the presence of imperfections. The main computational cost is solving a standard Poisson equation, or for surfaces with nontrivial topology, a sparse linear program.

3.1 Winding numbers as a jump harmonic function

We use M to denote a surface domain, and Γ a collection of oriented curves on M . As input, the algorithm takes in M and Γ .

Classic winding numbers jump by 1 when one crosses the curve from the right, yielding a piecewise constant function for closed curves. For broken curves, we must consider the more general solid angle function, which Jacobson et al. [2013] call the *generalized winding number* (GWN). In turn, solid angle is not well-defined for curves on surfaces, leading us to consider

the broader class of harmonic functions with jumps, or *jump harmonic functions* for short. For a non-self-intersecting curve Γ , a jump harmonic function u satisfies a Laplace equation with jump boundary conditions, which we call a *jump Laplace equation*:

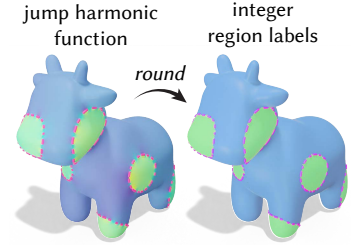
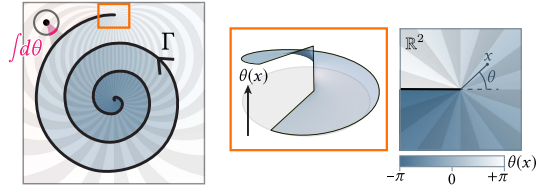
$$\begin{aligned} \Delta u &= 0, & \text{on } M \setminus \Gamma, \\ u^+ - u^- &= 1, & \text{on } \Gamma, \\ \partial u^+ / \partial n &= \partial u^- / \partial n, & \text{on } \Gamma. \end{aligned} \quad (3.1)$$

More generally, u can jump by a different integer value across Γ if Γ wraps onto itself multiple times.

Harmonic functions continuous up to jumps also arise naturally in surface parameterization, for instance, as conjugate harmonic functions in conformal mapping [Gu and Yau 2003; Sawhney and Crane 2017]; our treatment of such functions is similar to

Tong et al. [2006]. A jump harmonic function u is in fact harmonic everywhere, including at points on the curve, modulo the integer jumps; the angle-valued function u only fails to be harmonic at curve endpoints, where there is a branch-point singularity (inset) — see [Krutitskii 2001] for a careful treatment in the case $M = \mathbb{R}^2$.

If the domain M is simply-connected, meaning there cannot exist nonbounding loops on M , then we can simply solve Equation 3.1 for the function u . If Γ is closed, u is analogous to the ordinary winding number in the plane, which yields a piecewise constant, integer-valued function (inset). If Γ is not closed, but M is still simply-connected, then u yields a “soft” real-valued (rather than purely integer-valued) indicator function, analogous to solid angle.



3.2 Derivative processing

If M is not simply-connected, and the input curves Γ have nonbounding components, then the solution u to Equation 3.1 will not look like a region labeling (Figure 3.1). The crux of our method is to filter nonbounding components by processing the derivative ω of u .

If $\omega = 0$, then $u(x)$ is piecewise constant, meaning that $u(x)$ is already a valid (piecewise constant) region labeling. Conversely, if $\omega \neq 0$, then there are nonbounding components of Γ . In particular, nonbounding components of Γ , which are noncongruent to zero in the first homology group $H_1(M)$, are encoded by the harmonic component γ of the 1-form ω , which is noncongruent to zero

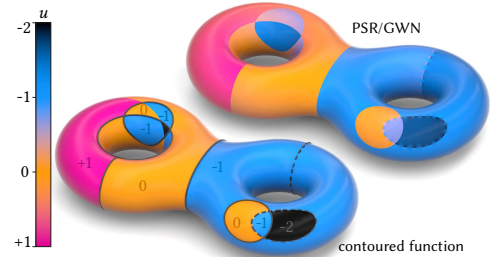
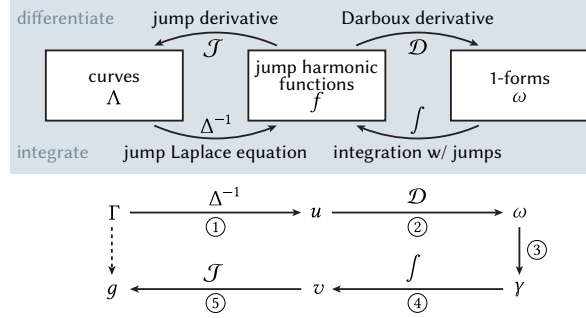


Fig. 3.1: On surfaces, contouring the solution to Equation 3.1, equivalent to Poisson surface reconstruction (PSR) and generalized winding numbers (GWN), can yield regions that do not follow the input curves, and/or jump across nonbounding curves.

in the first cohomology group $H^1(M) = \ker(d_1) \setminus \text{im}(d_0)$ (Section 2.3.2).



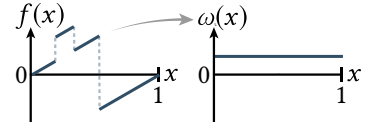
Our key insight is that jump harmonic functions form the “bridge” through which we can transform curves into functions, and vice versa. In particular, jump harmonic functions can both encode curves (where the function jumps in value) as well as a curve’s homology class (through its derivative). Our algorithm will amount to a round-trip around the diagram to the left, where we first translate the input curve

Γ into a jump harmonic function (Equation 3.1), then differentiate this function to obtain a differential 1-form. We then translate the resulting 1-form back into a jump harmonic function and curve, which will correspond to the final winding number function and curve decomposition, respectively. Performing these translations amounts to using appropriate notions of differentiation and integration.

3.2.1 Differentiating and integrating jump harmonic functions

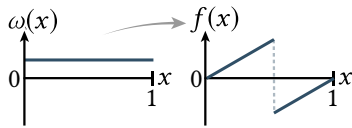
Discontinuous functions $u : M \rightarrow \mathbb{R}$ that jump by an integer across certain curves can also be represented as continuous angle-valued functions $\varphi := e^{2\pi i u}$. Since $e^{2\pi i s} = e^{2\pi i (s+1)}$, angle-valued functions “forget” about any integer jumps, yielding a continuous function that enables one to define a suitable derivative for u even at points where u is discontinuous.

In a bit more detail, the notions of differentiation and integration simply correspond to their usual notions from ordinary calculus, except we have to take extra care when dealing with the discontinuities of jump harmonic functions. For simplicity, we’ll first consider a periodic 1-dimensional function $f(x)$ defined on the unit interval $[0, 1]$. The distributional derivative of any such function can be expressed as



$$f'(x) = \omega(x) + \sum_i \Lambda_i \delta_{x_i}$$

where ω is a periodic piecewise smooth function, and Λ_i is the size of the jump at x_i (Figure 3.2). Likewise, we decompose the change in a jump harmonic function f into a 1-form describing continuous change in f , given by the *Darboux derivative* $\omega := Df$, which can be thought of as the ordinary differential df “modulo jumps”, and a 1-chain describing discontinuous jumps, given by the jump derivative $\Lambda := Jf$. Just as $\omega(x)$ “forgets” about the jumps in a 1D piecewise linear function (see inset), Df forgets about jumps across region boundaries on a surface.



Because the Darboux derivative of a jump harmonic function “forgets” the jumps, there is no unique inverse to Darboux differentiation. In particular, we can only integrate “up to jumps”: a canonical choice is perhaps a piecewise linear function with constant slope, but in general there are many possible piecewise differentiable functions f such that the continuous part of f' equals ω . Ordinarily this function would be determined (up to a constant) via standard integration, but for a periodic function there may be no continuous solution — for example, if ω is strictly positive. Instead, we must decide where f should jump.

Our understanding of the 1D case more or less extends directly to the 2D case. On surfaces M of non-trivial topology, after solving for a jump harmonic function u via Equation 3.1, we compute the Darboux derivative $\omega = \mathcal{D}u$. We use Hodge decomposition (Section 2.2) to extract the harmonic part γ of ω . In this case, only $\delta\beta$ will be nonzero, due to singular behavior near interior endpoints. Hence, we need only solve a single Poisson equation $\Delta_2\beta = d\omega$, then evaluate $\gamma \leftarrow \omega - \delta\beta$.

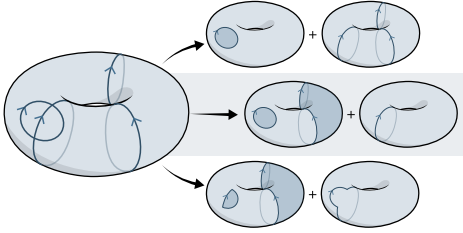


Fig. 3.3: A collection of loops can be decomposed into bounding and nonbounding components in many different ways. We look for the decomposition whose residual is shortest (*middle*).

solving for a jump harmonic function with jumps encoded by Γ , minus the jumps in the residual function v .

3.3 Results

Incomplete oriented curves arise in many settings, ranging from curves projected onto noisy surfaces, to strokes painted on a noisy mesh, to imperfect user selections. We apply our method, abbreviated as *surface winding numbers (SWN)* to several such tasks.

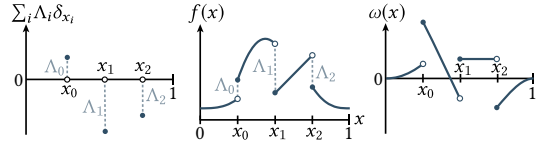


Fig. 3.2: The derivative of any piecewise smooth function $f(x)$ on a periodic interval (*center*) can be decomposed into a piecewise smooth function $\omega(x)$ (*right*) plus a sum of delta functions (*left*). The former captures continuous changes in f while the latter captures jumps in f .

Next, our goal is now to find a *residual function* v whose Darboux derivative looks like γ , and hence describes the nonbounding part of our input curves. If we imagine this nonbounding part is a curve Γ' , then v must jump across Γ' , and should not jump across the complementary bounding component $\Gamma \setminus \Gamma'$. However, the choice of Γ' is in general ambiguous (Figure 3.3). Hence, we look for the *minimal* jumps needed for v to integrate γ . Ultimately, the residual function v is another jump harmonic function whose jumps encode a completion of the nonbounding components of Γ . Finally, we solve for the final winding number function w by

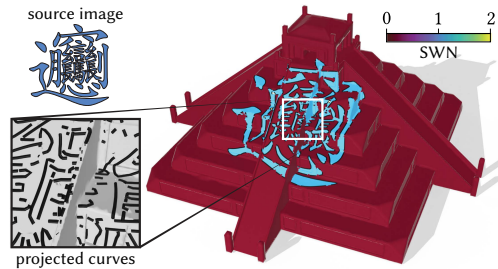


Fig. 3.4: A complicated shape is projected onto a ziggurat with sharp overhangs, creating broken curves; SWN nicely fills in the bounded regions.

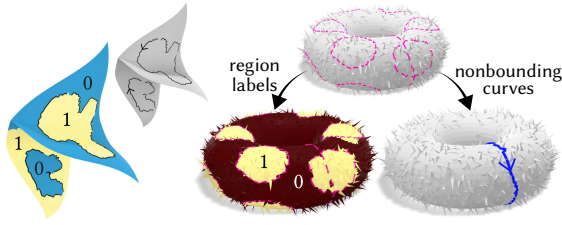


Fig. 3.5: Even on highly non-manifold meshes, SWN can produce an effective region labeling and completions of nonbounding curves.

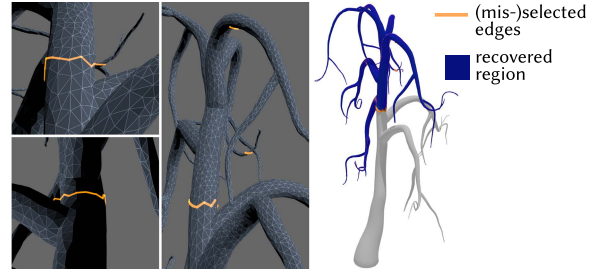


Fig. 3.6: A common frustration with screen-space selection is that distant edges are often selected unintentionally. SWN filters out spurious parts of the selection, and completes loops to yield the expected segmentation.

Robustness. SWN is robust to defects in the input curves (Figures 3.14 and 3.4). Especially if gaps are reasonably small, we generally recover the same regions as for equivalent closed curves (Figure 3.7). In practice our method is also robust to low-quality geometry (Figure 3.13), meshes with low-quality elements (Figure 3.8), and highly nonmanifold connectivity (Figure 3.5), owing to the strong regularity of elliptic problems. Since it is purely intrinsic, surface self-intersections do not result in region misclassification. Moreover, an intrinsic formulation also enables us to use robust methods for *intrinsic retriangulation* if the mesh is particularly bad [Sharp et al. 2019b; Gillespie et al. 2021; Sharp et al. 2021], as illustrated in Figure 3.8.

Sketching on surfaces. SWN robustly handles imperfect broken curves, like ones drawn in surface sketching and painting where user input is imprecise. For instance, in Figure 3.13 a user sketches very reasonable yet broken curves; SWN yields a nice coloring of the sketched regions, which can be further refined by the user. Figure 3.14 demonstrates the utility of SWN even in 2D, where a user draws rough strokes to segment a complex shape. Here, GWN yields undesirable results — despite being a 2D method — since the influence of open strokes leaks across the domain boundary, whether or not the boundary itself is included in Γ . Likewise, GWN may not produce the expected result for 2D regions with holes — for instance, directly rounding the function u in Figure 3.9 would yield the same kind of phantom curves seen in Figure 3.1.

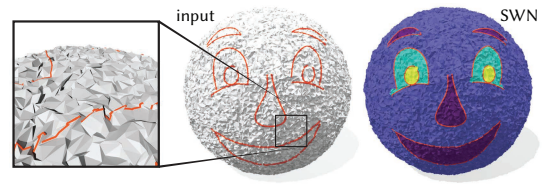


Fig. 3.13: We can robustly identify regions even on geometry with severe noise, intersections, and fold-over. Here, several strokes quickly painted in screen space are used to color regions on the surface.

Stamping and booleans. We can also perform robust boolean operations on surfaces, even for defective domains and/or curves. To get initial shapes, we can for instance “stamp” existing vector graphics onto the surface (Figures 3.4 and 3.11). Rather than worry about numerically

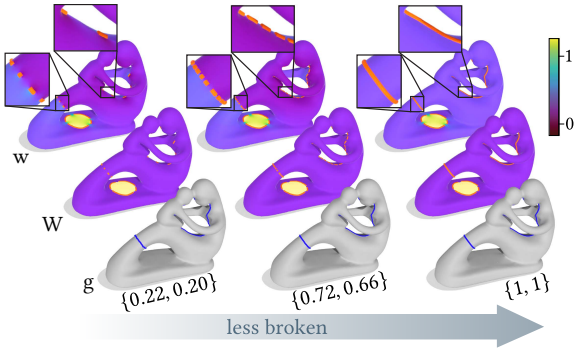


Fig. 3.7: As Γ becomes less broken, w approaches the expected winding number function, and the coefficients on nonbounding loops g approach 1. Throughout, the rounded winding number W yields the correct inside-outside classification, filtering out nonbounding components even for very broken inputs.

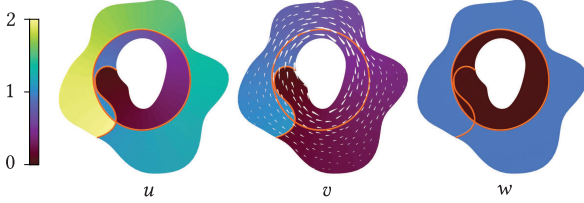


Fig. 3.9: Even for planar regions, one must think carefully about how curves do (or do not) bound regions. Here, SWN correctly filters out the influence of a nonbounding curve connecting two boundary components.

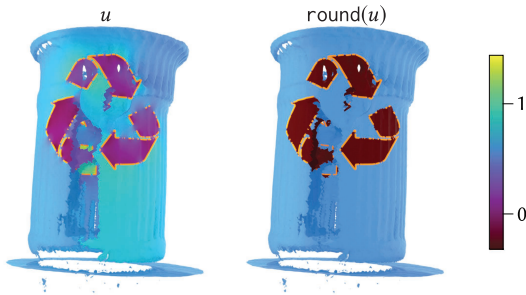


Fig. 3.11: A recycling logo is projected onto a noisy 3D scan of a trash can from [Choi et al. 2016], creating a highly broken curve. Despite large holes in the scan, SWN produces a reasonable region labeling.

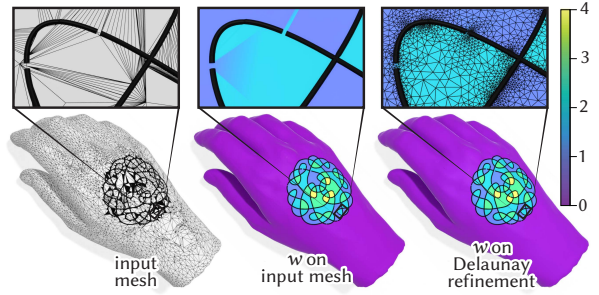


Fig. 3.8: Even on meshes with low element quality, SWN can produce reasonable region labels (*center*). Since our formulation is intrinsic, any remaining artifacts can be eliminated via *intrinsic Delaunay refinement* (*right*).

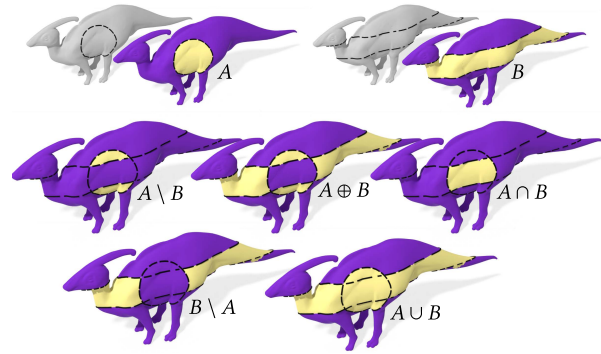


Fig. 3.10: Unlike previous methods, we can compute boolean operations on regions defined by imperfect, broken curves on surfaces.

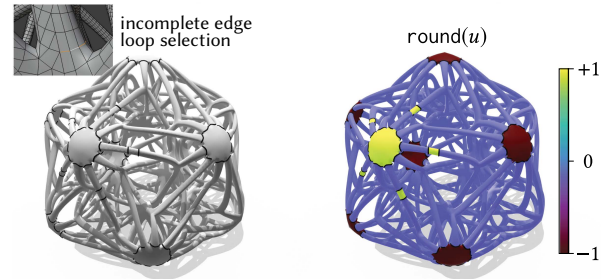


Fig. 3.12: Many 3D modeling tools provide edge loop selection tools, but are easily tripped up by irregular connectivity such as this mixed quad-triangle mesh (*left*). By reasoning about functions rather than edges, we robustly infer user intent (*right*), even on this topologically complex model.

robust intersection, we can lean on SWN to ensure we obtain well-defined regions. Boolean operations are then trivially computed via element-wise logical operations (Figure 3.10). Unlike *BoolSurf* [Riso et al. 2022], we can perform these operations for imperfect, broken curves — albeit at larger computational cost. Note also that unlike extrinsic mesh booleans [Zhou et al. 2016], we need not worry about self-intersections of the surface itself.

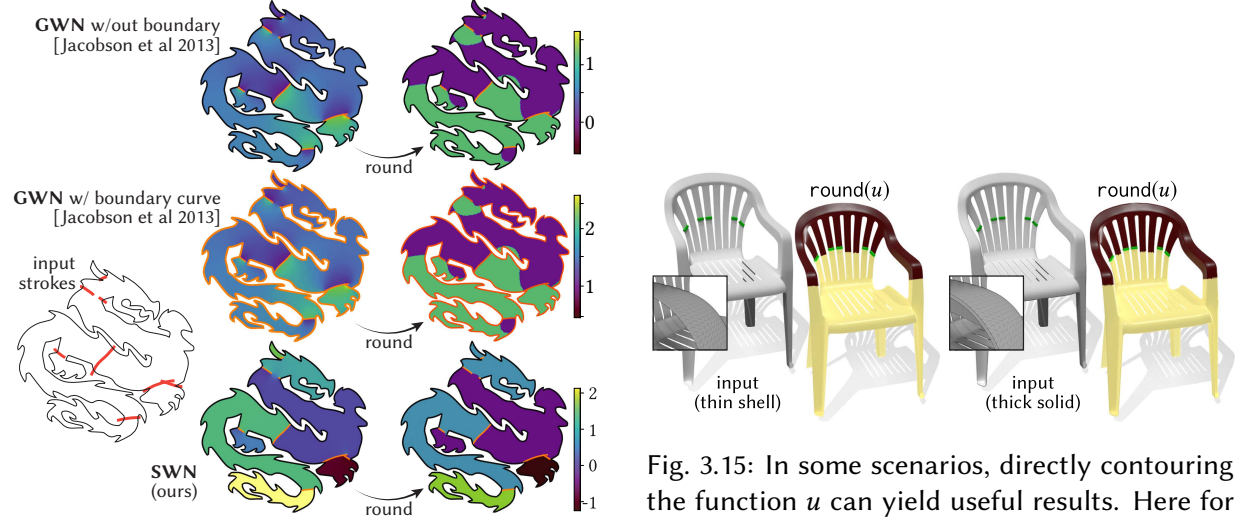


Fig. 3.14: *Left*: a user makes rough strokes to select regions of a 2D shape. *Top right*: GWN produces the wrong result, since the influence of strokes “leaks” across the domain boundary. *Middle right*: including the boundary curve just shifts GWN’s solution by +1. *Bottom right*: SWN produces the desired result, robustly handling gaps, misclicks, and intersecting strokes.

Fig. 3.15: In some scenarios, directly contouring the function u can yield useful results. Here for instance a user avoids the misery of selecting every small loop in a complex model. We also obtain a similar segmentation whether working with a solid (*right*) or shell-like model (*left*).

Region selection. Selection of regions on geometrically or topologically complex 3D models is a challenging user interface design problem. SWN is a valuable component for building such tools. For instance, Figure 3.6 highlights a common frustration when selecting mesh edges in screen space; here SWN automatically filters out misselected edges, capturing the user intent. Similarly, Figure 3.12 shows how SWN can be used to repair loops that are not easily chosen via edge-based selection tools common to 3D modelers. Other tools provide facilities for directly selecting regions rather than curves, *e.g.*, using a lasso or “fat” paintbrush. Here, however, one encounters the same problem: a region selected in screen space might inadvertently highlight distant, unintentional pieces of the surface. One could likewise use SWN to filter the *boundary* of such a selection. Finally, Figure 3.15 shows an example where one might *not* want to filter out nonbounding loops. Here, rather than process the function u , we simply apply the contouring procedure from Section ??, yielding loops that did not belong to the input, yet automatically complete the implied segmentation.

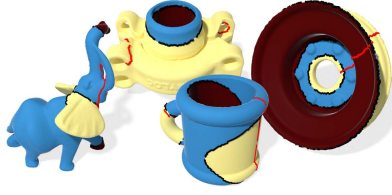


Fig. 3.16: Four of the 934 test cases in our synthetic benchmark. Each model is assigned ground truth region labels (indicated by colors), along with broken boundaries for those regions (black), and additional broken nonbounding loops (red).

Performance and accuracy. To measure the success rate of our algorithm, we constructed a synthetic dataset of models with ground truth regions and nonbounding loops (Figure 3.16). We started with the meshes from Myles et al. [2014], remeshed them to resolutions between 10k and 90k vertices, and generated random regions by taking sublevelsets of low-frequency Laplacian eigenfunctions. To obtain nonbounding loops, we computed a greedy homology basis [Erickson and Whittlesey 2005], picked a random subset of the loops, and straightened them slightly using *FlipOut* [Sharp and Crane 2020b] before snapping them back to mesh edges. We then deleted random segments from these curves. In total, we obtained 934 test cases of which 451 were defined on nonsimply-connected surfaces (*i.e.*, those with nontrivial topology). For each test case,

we quantify error as the percentage of surface area mislabeled by our method.

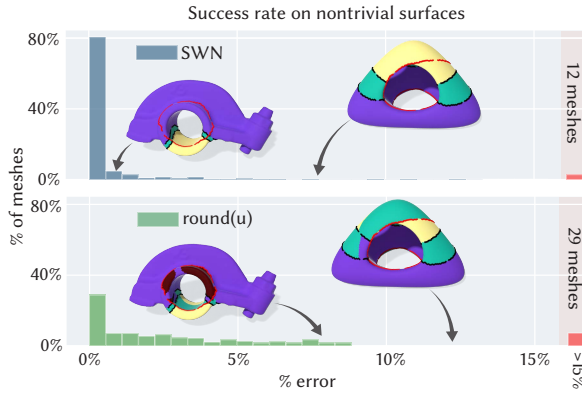


Fig. 3.17: Error rates for SWN (*top*) compared to naïve rounding of u à la GWN (*bottom*). Error is quantified as percentage of mislabeled surface area. The two highlighted examples show how naïve rounding can fail to filter out nonbounding loops (in red) which are correctly identified by SWN.

On simply-connected surfaces our method typically takes less than two seconds (see Figure 3.18, *top*), and achieves a mean/max error of only 0.14%/5%. On nonsimply-connected surfaces, there was occasionally fundamental ambiguity in the input, yielding results quite different from the ground truth (Figure 3.19), but in general our method remains quite accurate, achieving

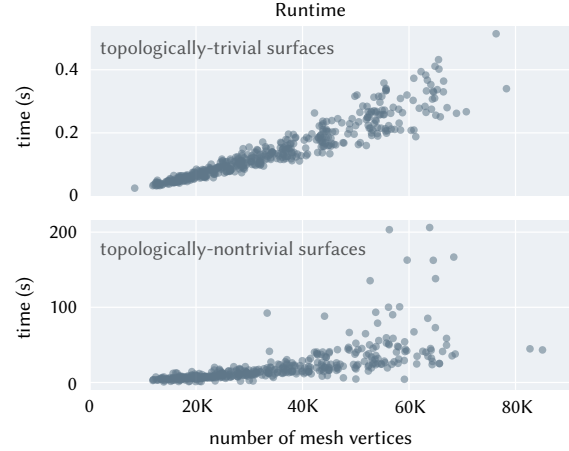


Fig. 3.18: *Top*: on topologically trivial surfaces, our method boils down to a quick linear solve. *Bottom*: on surfaces with nontrivial topology we must also solve a linear program, which becomes the computational bottleneck. These timings were measured using an Intel Xeon W-1250 CPU with 16 GB of RAM.

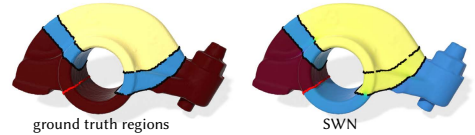


Fig. 3.19: Since SWN always extracts the shortest collection of nonbounding curves (Figure 3.3), it may not always reproduce the ground truth — but still gives a reasonable segmentation.

errors under 0.5% on 80% of models. More importantly, SWN performs much better than naïve rounding of the function u à la GWN, which can create phantom curves (Figure 3.1) which significantly degrade the accuracy of the final labels (Figure 3.17). The linear program takes much longer than the single linear solve, but still runs in a matter of minutes (Figure 3.18, *bottom*); see Section ?? for a more approximate, but faster alternative.

CHAPTER 4

From inside-outside to signed distance

Reconstruction of curves and surfaces, and signed distance computation, represent two fundamental problems in geometry processing. Of course, the two problems are highly related, because a meaningful notion of signed distance depends on a meaningful notion of inside and outside, which is often provided by reconstruction. Chapter 5 in fact describes a robust algorithm for computing signed distance that works by essentially achieving simultaneous reconstruction and distance.

The connection between reconstruction and distance computation can be formalized: in Section 4.1, we show that winding numbers-based reconstruction and Poisson Surface Reconstruction are related to signed distance via a relatively simple change of variables. This relationship gives rise to *convolutional distance approximations*: a class of algorithms that approximate the minimum distance to a given shape through a summation of kernels concentrated on the shape’s boundary. Such approximations for unsigned distance have been re-discovered several times throughout image processing, signal processing, computer vision, and computer graphics. Signed variants, though they are not described as much in the literature, can also be derived. At a high level, these (un)signed distance approximations may be obtained as viscosity solutions to eikonal equations, or through asymptotic analysis using Laplace’s method, encompassing the well-known Hopf-Cole transformation for PDEs, Varadhan formulas for geodesic distance, and LogSumExp methods and softmax functions used in machine learning.

However, while such convolutional distance approximations produce good results for densely sampled boundaries such as polygon meshes and image contours, we also show that no convolutional approximation can yield good results on sparsely sampled boundary data consisting of isolated point samples, such as point clouds. Instead, it is necessary to decouple regression from distance computation to some extent, resulting in the multi-step but especially robust *Signed Heat Method* described in Chapter 5. It remains an open problem in vision and geometry processing whether there exists a both fast and accurate convolutional algorithm that computes robust (un)signed distance to point clouds (Section 6.1).

4.1 A unification of winding numbers, Poisson Surface Reconstruction, and signed distance

In this section, we describe the connection between winding numbers, Poisson Surface Reconstruction, and signed distance.

We start by considering the following eikonal equation for unsigned distance:

$$\begin{aligned} \|\nabla u\|^2 &= 1 & x \notin \Omega \\ u(x) &= 0 & x \in \Omega \\ \frac{\partial u^\pm}{\partial n}(x) &= \pm 1 & x \in \Omega. \end{aligned} \quad (4.1)$$

This equation describes an unsigned distance function $u(x)$ to Ω in some domain M (inset). The boundary conditions consist of Dirichlet boundary conditions that state the distance should be zero at the source geometry Ω , and double-sided Neumann conditions that state $u(x)$ should be increasing on either side away from Ω .

Distance functions, however, are not differentiable at the cut locus of the source geometry Ω , the locus of points at which the minimizer of distance to Ω is non-unique. For x on the cut locus, $\nabla u(x)$ is hence undefined and the meaning of Equation 4.1 is unclear. Distance functions instead satisfy Equation 4.1 in a “viscosity sense”: whereas a true distance function is not everywhere differentiable, it can be obtained as the limit of the solution to a perturbed version of the eikonal equation as the amount of perturbation goes to zero. In particular, Equation 4.1 is perturbed by adding a small amount of scalar diffusion, obtaining a “viscous” eikonal equation

$$\begin{aligned} \|\nabla u\|^2 - 1 &= \frac{1}{\lambda} \Delta u(x) & x \notin \Omega \\ u(x) &= 0 & x \in \Omega \\ \frac{\partial u^\pm}{\partial n}(x) &= \pm 1 & x \in \Omega. \end{aligned} \quad (4.2)$$

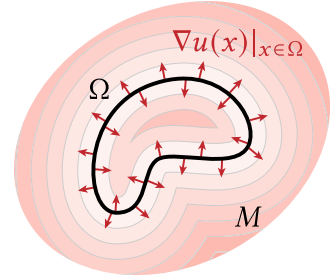
The seminal work by Crandall and Lions [1983] introduced *viscosity solutions* as a type of generalized solution unique for Hamilton-Jacobi equations.

Equation 4.2 is an example of a time-independent, viscous *Burgers’ equation*, for which there is a well-known change-of-variables called the *Hopf-Cole transformation* (sometimes called *Cole-Hopf transformation*) [Hopf 1950; Cole 1951]. Evans [1998, Section 4.4] gives a derivation of the transformation, which through exponentiation turns a nonlinear time-dependent viscous Burgers’ equation into linear heat equation. A variant of this transformation,

$$w(x) = \exp(-\lambda u(x)) \quad (4.3)$$

turns the nonlinear, viscous eikonal equation in Equation 4.2 into a linear screened Poisson equation, where the viscosity now acts as a screening term that controls the amount of damping on a diffusive process [Belyaev and Fayolle 2015]:

$$\begin{aligned} \Delta w(x) - \lambda^2 w(x) &= 0 & x \notin \Omega \\ w(x) &= 1 & x \in \Omega \\ \frac{\partial w^+}{\partial n}(x) &= -\frac{\partial w^-}{\partial n}(x) & x \in \Omega. \end{aligned} \quad (4.4)$$



The Hopf-Cole transformations, as applied to the heat equation and the screened Laplace equation in Equation 4.4, are identical to the two *Varadhan's formulas* for geodesic distance [Varadhan 1967].

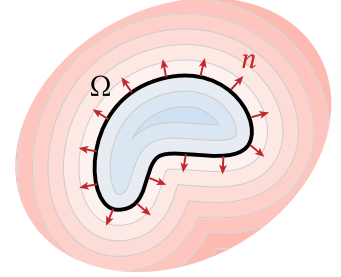
Taking the Hopf-Cole transformation further, we now consider a *signed* eikonal equation, which solves for signed rather than unsigned distance:

$$\begin{aligned} \|\nabla u(x)\|^2 &= 1 & x \notin \Omega \\ u(x) &= 0 & x \in \Omega \\ \frac{\partial u}{\partial n}(x) &= 1 & x \in \Omega. \end{aligned} \quad (4.5)$$

Compared to Equation 4.1 for unsigned distance, Equation 4.5 has Neumann conditions that are continuous across Ω (inset).

We can likewise consider a signed eikonal equation with viscosity,

$$\begin{aligned} \text{sign}_\Omega(x) (\|\nabla u(x)\|^2 - 1) &= \frac{1}{\lambda} \Delta u(x) & x \notin \Omega \\ u(x) &= 0 & x \in \Omega \\ \frac{\partial u}{\partial n}(x) &= 1 & x \in \Omega, \end{aligned} \quad (4.6)$$



and a *signed* variant of the Hopf-Cole transformation

$$w(x) = \text{sign}_w(x) \exp(-\lambda \text{sign}_w(x) u(x)) \quad (4.7)$$

which we call the *signed Hopf-Cole transformation*. Applying Equation 4.7 to Equation 4.6 yields a *jump* screened Laplace equation,

$$\begin{aligned} \Delta w(x) - \lambda^2 w(x) &= 0 & x \notin \Omega \\ w^\pm(x) &= \pm 1 & x \in \Omega \\ \frac{\partial w^+}{\partial n}(x) &= \frac{\partial w^-}{\partial n}(x) & x \in \Omega. \end{aligned} \quad (4.8)$$

Compared to the screened Laplace equation in Equation 4.4, the jump screened Laplace equation in Equation 4.8 has boundary conditions that make the solution jump across the source geometry Ω . A derivation of the signed Hopf-Cole transformation, valid for both closed and open Ω , is given in Appendix A.

The relationship between distance functions and screened Laplace solutions via the Hopf-Cole transformation can be described intuitively as follows. The jump screened Laplace equation in Equation 4.8 diffuses double-sided boundary values from Ω , where larger values of λ means greater screening (damping) so that boundary data is diffused less and instead more concentrated around Ω . Diffusion yields a function that is monotonic in distance, and in particular, its decay is roughly exponential in distance, so applying the inverse signed Hopf-Cole transformation

$$u(x) = -\frac{1}{\lambda} \text{sign}_w(x) \log |w(x)|$$

essentially reverses this exponential decay by taking a log, giving an approximation of distance (Figure 4.1, right).

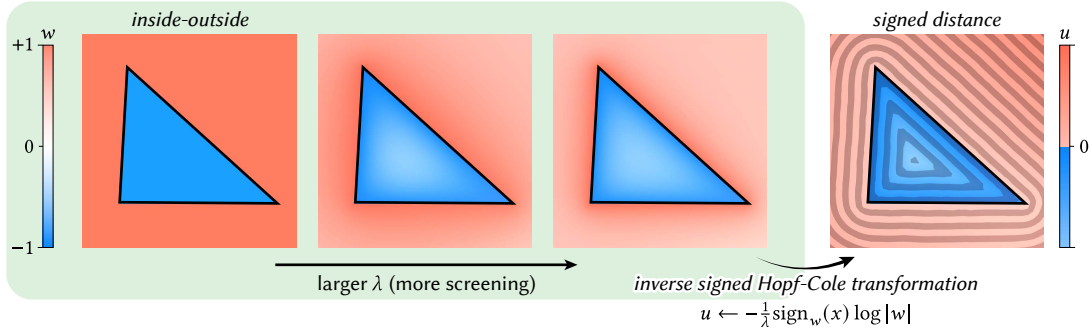


Fig. 4.1: As the screening amount λ goes to 0, the solution w to the jump screened Laplace equation in Equation 4.8 converges to an inside-outside function (*left*). As the screening amount increases, w converges to a signed distance function via a log-transform. (*right*).

The jump screened Laplace equation in Equation 4.8 can further be rewritten as the screened Poisson equation

$$\Delta w(x) - \lambda^2 w(x) = -2 (\nabla \cdot n(x) \mu_\Omega(x)) \quad (4.9)$$

where $n(x)$ denotes the outward-pointing unit normals to Ω , and μ_Ω is a measure concentrated on Ω . A derivation of Equation 4.9 is given in Appendix ??.

The Green's function $G^\lambda(x, y)$ of the screened Laplace operator, called the *Yukawa potential* or *screened Coulomb potential*, has the expression

$$G^\lambda(x, y) := \frac{\exp(-\lambda \|x - y\|)}{4\pi \|x - y\|} \quad (4.10)$$

and can be used to express the solution $w(x)$ of Equation 4.8 as the boundary integral equation

$$w(x) = 2 \int_\Omega \frac{G^\lambda(x, z)}{\partial n(z)} dz = \int_\Omega \frac{(\lambda \|x - z\| + 1) \langle x - z, n(z) \rangle}{2\pi \|x - z\|^3} \exp(-\lambda \|x - z\|) dz, \quad x \notin \Omega. \quad (4.11)$$

Unlike the distance function u for which $|u(x)| \rightarrow \infty$ as $\|x\| \rightarrow \infty$, the solution w of Equation 4.8 exhibits exponential decay at infinity, and so can be represented by the boundary integral in Equation 4.11.

Fascinatingly, these derivations establish a close relationship between a few classic occupancy methods and signed distance. On one hand, as the screening parameter $\lambda \rightarrow 0$, the jump screened Laplace equation in Equation 4.8 becomes a jump Laplace equation whose solutions — so called *jump harmonic functions* (Chapter 3) — describe generalized winding number (Figure 4.1, *left*). In turn, generalized winding number is a special case of *Poisson Surface Reconstruction* [Kazhdan et al. 2006]: Poisson Surface Reconstruction is equivalent to a regularized version of winding numbers, which corresponds to convolving the right-hand side of the Poisson equation in Equation 4.9 with a Gaussian [Chen et al. 2024] (and taking $\lambda \rightarrow 0$). In summary, in the smooth setting at least, we can obtain signed distance from occupancy methods simply by introducing a screening term into their PDEs.

Practical issues. On discrete surfaces, Crane et al. [2013b] advocate against approximating geodesic distance by solving a screened Laplace or heat equation then applying a Hopf-Cole transformation. In particular, for the distance approximation to be accurate, one needs to take the diffusion time $t \rightarrow 0$, but on discrete surfaces one cannot take t too small else the approximation converges to the graph metric [Crane et al. 2013b, Appendix A]. On $\mathbb{R}^n$, however, we can use the boundary integral in Equation 4.11 uses not a discrete Laplacian, but Green’s function for the screened Laplace operator without discretizing $\mathbb{R}^n$ – hence we can take $t = \lambda^{-2}$ much smaller (barring precision issues).

A more significant drawback of applying Hopf-Cole transformations/Varadhan’s formulas is that they can be numerically unstable. First, the distance approximation can be inaccurate near Ω where $|w| \leq 1$, and $\log |w|$ hence has a very large derivative – so any existing errors near Ω become exacerbated after applying a log transform. These errors may arise, for example, due to the Yukawa potential’s singular behavior at Ω . Second, one quickly runs into numerical precision issues far away from Ω due to the exponential decay of the Yukawa potential and heat kernel.

It may be possible to improve distance accuracy with additional computation, for example using the higher-order corrections proposed by Belyaev and Fayolle [2024]. But in Section 4.3, we will also see that these distance approximations are also fundamentally flawed when acting on sampled data, such as point clouds, severely limiting their direct applicability to real-world data.

4.2 Convolutional distance approximations

Before discussing further limitations of the distance formula obtained in the previous section, we first observe that the formula can be generalized to a broader class of distance formulas.

The boundary integral formulation of signed distance in Equation 4.11 implies that signed distance can be obtained by convolving a particular exponential kernel over the source geometry Ω , then applying the appropriate log transformation. The kernel used for convolution, however, does not have to be the Yukawa potential (Equation 4.10). In fact, for *any* exponential kernel with parameter λ appearing inside the exponential, and for any continuous function $h : \mathbb{R}^d \rightarrow \mathbb{R}$ and twice-differentiable function $\varphi : \mathbb{R}^d \rightarrow \mathbb{R}$, we have the following asymptotic behavior:

$$\int_{\Omega} h(z) \exp(-\lambda\varphi(z)) \, dz \stackrel{\lambda \rightarrow +\infty}{\sim} (2\pi/\lambda)^{d/2} \det(\nabla^2\varphi(x^*))^{-1/2} h(x^*) \exp(-\lambda\varphi(x^*)) \quad (4.12)$$

where $x^* := \operatorname{argmin}_{z \in \Omega} \varphi(z)$ is the minimizer of the exponential argument φ , assumed to be unique [Tibshirani et al. 2024, Eq. 11]. The observation in Equation 4.12 is an example of *Laplace’s method*, a classic technique in asymptotic analysis [Evans 1998, §4.5], [Bender and Orszag 1999, §6.4].

Intuitively, the integral on the left-hand side of Equation 4.12 becomes increasingly peaked where φ is largest, such that in the limit all other contributions become *subdominant*, that is, exponentially small with respect to this peak contribution. Applying $-\frac{1}{\lambda} \log(\cdot)$ to both sides of

Equation 4.12, asymptotically we obtain a direct estimate of the minimum of φ :

$$-\frac{1}{\lambda} \log \left(\int_{\Omega} h(z) \exp(-\lambda \varphi(z)) dz \right) \sim \varphi(x^*) + \frac{d \log \lambda}{2\lambda} - \frac{h(x^*)}{\lambda} + O(\lambda^{-1}), \quad \lambda \rightarrow +\infty.$$

All terms beyond the first go to 0 as $\lambda \rightarrow \infty$.

Taking $\varphi(z)$ to be the minimum distance function to x , that is, $\varphi(z) = \operatorname{argmin}_{z \in \Omega} \|x - z\|$, and taking the domain of integration to represent the source geometry Ω to which we compute distance, we define

$$\tilde{d}(x) = -\frac{1}{\lambda} \log \left(\int_{\Omega} h_x(z) \exp(-\lambda \|x - z\|) dz \right) \quad (4.13)$$

as the general form of a *convolutional distance formula* that estimates the minimum distance from a point x to Ω . A special case Equation 4.13 is the formula we derived in Section 4.1 via the Hopf-Cole transformation, where $h_x(z) = \frac{(\lambda \|x - z\| + 1) \langle x - z, n(z) \rangle}{2\pi \|x - z\|^3}$. Similar asymptotic analysis was also used by Varadhan [1967], whose formulas yield the same exponential change-of-variables as Hopf-Cole.

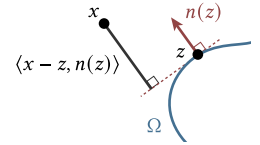
We can also obtain a *self-normalized* convolutional formula (terminology we take from Belyaev and Fayolle [2024])

$$\hat{d}(x) = \frac{\int_{\Omega} g_x(z) \exp(-\lambda \|x - z\|) dz}{\int_{\Omega} \exp(-\lambda \|x - z\|) dz} \quad (4.14)$$

that gives a smooth approximation of $g_x(x^*)$ as $\lambda \rightarrow \infty$, where $x^* = \operatorname{argmin}_{z \in \Omega} \varphi_x(z)$ is again the minimizer of $\varphi_x(z) = \|x - z\|$. Equation 4.14 can be obtained either by applying Laplace's method twice (once to the numerator, and once to the denominator), or by taking the derivative of Equation 4.13 with respect to λ . When used to approximate signed distance, most authors use the *pseudonormal distance* [Alexa et al. 2001; Boissonnat and Cazals 2002; Kolluri 2008; Öztireli et al. 2009; Yang et al. 2025]

$$g_x(z) = \langle x - z, n(z) \rangle$$

that simply uses the query point x 's projection onto a tangent plane (inset), though of course other local distance approximations may be used.



Equation 4.14 is an example of a *kernel density estimator*, and offers greater flexibility than Equation 4.13 in the sense that it instead interpolates the function g , which need not be a distance function or even scalar-valued. For example, Sharp et al. [2019c] use a version of Equation 4.14 to compute closest-point interpolation of both scalar- and vector-valued data on manifolds. Equation 4.14 can also be seen as a partition-of-unity method that computes an expected value of $g_x(z)$, where the local estimate at $z = z'$ has probability $\frac{\exp(-\lambda \|x - z'\|)}{\int_{\Omega} \exp(-\lambda \|x - z\|)}$.

Instances of the two convolutional formulas in Equation 4.13 and Equation 4.14 appear widely across mathematics, computer science, and engineering. For instance, we have already mentioned that Varadhan’s formulas are examples of Equation 4.13. In computer vision, the *Schrödinger distance transform* uses the exponential asymptotics of the Schrödinger equation to arrive at an instance of Equation 4.14 [Gurumoorthy and Rangarajan 2009; Sethi et al. 2012]. Various versions of Equation 4.13 or Equation 4.14 were further observed by Kolluri [2008], Karam et al. [2019], Madan and Levin [2022], and Abgrall [2022]. More recently, the relationship between eikonal and screened Laplace equations has been used in neural reconstruction methods [Lipman 2021; Wang et al. 2025].

More generally, the exponential asymptotics that underlie convolutional distance formulas also underlie common logistic regression techniques used for classification problems. Equation 4.13, when applied to discrete data, is a generalization of the *LogSumExp* function, which is commonly used as a smooth relaxation of the minimum or maximum operator in machine learning. The LogSumExp function is also known as the *Kreisselmeier-Steinhauser function*, particularly in the systems and control community [Kreisselmeier and Steinhauser 1980]. The gradient of the LogSumExp function, called the *softmax function* in machine learning, is an instance of a self-normalized convolutional formula in Equation 4.14, and has the same form as the *Boltzmann distribution* commonly used as a probability distribution in statistical mechanics. Tibshirani et al. [2024] gives an excellent survey of further connections between Laplace’s method and smooth minimizers throughout the fields of convex optimization, statistics, machine learning.

Other properties. Equation 4.13, the first convolutional distance formula, gives an underestimate of the true distance to Ω if both sides of Equation 4.12 are multiplied by at least $\lambda^{d/2}$, and $h \geq 1$. (Note that the dimension d will be the dimension of Ω , i.e. $d = 2$ if Ω is a surface in $\mathbb{R}^3$.) Also, if x is on the cut locus of Ω , then φ no longer has a unique minimizer, in which case the distance approximation in Equation 4.13 has additional error, but one that gets absorbed into the $O(\lambda^{-1})$ term.

It remains an open question whether there are closed-form expressions for the integrals of Yukawa potentials or other exponential kernels over (rational) parametric curves or triangles, analogous to closed-form expressions for winding numbers [Jacobson et al. 2013; Liu et al. 2025]. In particular, the example in Figure 4.1 was obtained by solving a screened Laplace equation on a triangle mesh as opposed to closed-form integration along the source curves.

4.3 Fundamental limitations of convolutional distance

The asymptotic analysis in Section 4.2 implies that both convolutional distance formulas Equation 4.13 and Equation 4.14 are completely determined by the behavior of $\varphi_x(z)$ or $g_x(z)$, respectively, around the global minimizer x^* of the exponential argument $\varphi_x(z)$. Indeed, on some level, Equation 4.13 and Equation 4.14 can be interpreted as constructing a global distance approximation simply by blending exponentially-weighted local distance approximations centered

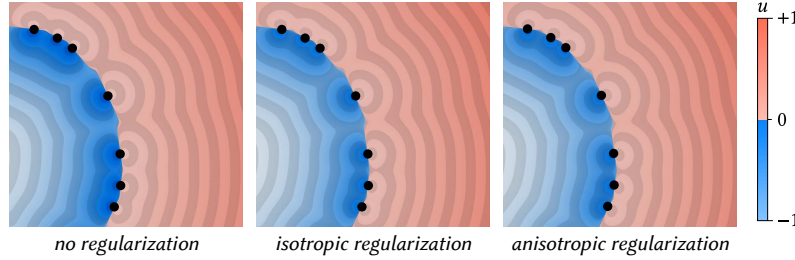


Fig. 4.2: Regularizing the kernel in Equation 4.13 is equivalent to choosing the function $h_z(x)$, which unfortunately has no effect asymptotically. Here, we use an exponential kernel with no regularization (*left*), with isotropic Gaussian regularization (*center*), and with anisotropic Gaussian regularization (*right*), which all produce non-robust distance for $\lambda = 100$.

on the geometry Ω , in the style of countless other classic meshless interpolation methods that use exponential radial basis functions, such as moving least squares surfaces, partition-of-unity methods, and smoothed particle hydrodynamics [Blinn 1982; Alexa et al. 2001; Carr et al. 2001; Boissonnat and Cazals 2002; Ohtake et al. 2005; Yu and Turk 2013].

However, the fact that convolutional distance approximations are essentially single-point approximations severely hinders their robustness to noisy, incomplete, imperfect geometry: it means that creating a convolutional distance approximation that gives generalized signed distance — that is, globally accurate signed distance to the true, uncorrupted geometry underlying the corrupted input — boils down determining a good function $\varphi_x(z)$ or $g_x(z)$, which is just as difficult as the general problem of computing generalized signed distance. For if we knew a good function $\varphi_x(z)$ or $g_x(z)$ whose global minimizer produced generalized signed distance, then it would be the solution to the general global problem (and vice versa).

Concretely, Equation 4.13 fares especially poorly on point-sampled data, where one obtains a distance function whose level sets are to the *sampled* geometry, rather than to the underlying surface from which the discrete geometry is sampled (Figure 4.3). As $\lambda \rightarrow \infty$, one simply “snaps” to the closest point in the input point set.

We might try regularizing the exponential kernel, based on the success of Chen et al. [2024] at regularizing the kernels used in winding number methods. These regularized kernels were introduced to avoid numerical and interpolation issues caused by the singularity of the ordinary, unregularized Poisson kernel used for winding numbers, and can be interpreted as adopting a stochastic model of the point cloud geometry. However, the choice of regularizing kernel is equivalent to choice of the $h_x(z)$ function, which has no effect asymptotically as $\lambda \rightarrow \infty$ (Figure 4.2). In other words, while we need very high λ to get good distance properties, the asymptotic properties of the exponential function rapidly negate the effect of any regularization we might choose.

Ultimately, we can either obtain better regression quality, or better eikonality, but not both: the parameter λ is coupled to both regression quality and distance accuracy, in opposite directions, and empirically there are no values of λ that yield acceptable results in both. On one end of the spectrum, λ is small and one obtains winding number methods, Poisson Surface

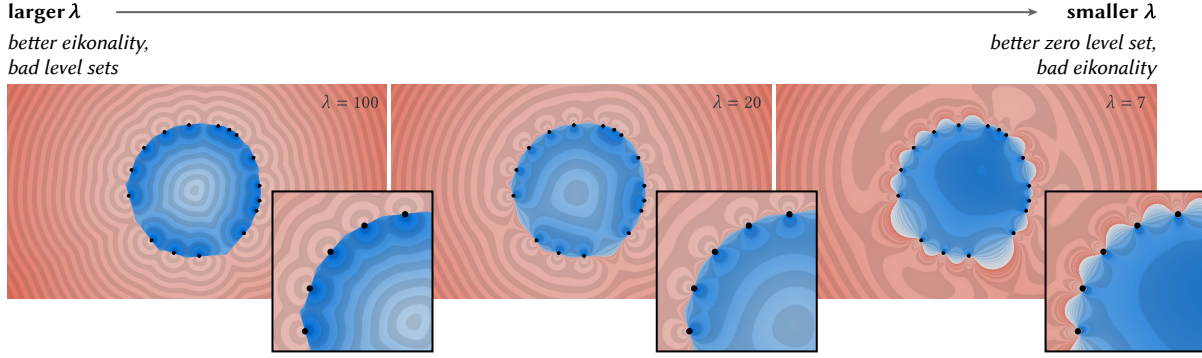


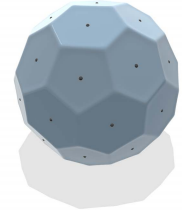
Fig. 4.3: For convolutional distance approximations, the parameter λ is coupled to both regression quality and distance accuracy, in opposite directions: large λ values are needed to obtain better eikonal, but overfits distance to the point set (*left*); and conversely, smaller λ 's give better reconstruction of the zero level set, in the sense that it at least interpolates the data points, but isn't close to a distance function (*center, right*). Here we show an example using Equation 4.13.

Reconstruction, and the method of Madan and Levin [2022], which essentially sacrifice distance accuracy to focus on regression quality (reconstruction). On the other end of the spectrum, one has the convolutional distance transforms applied in image processing [Gurumoorthy and Rangarajan 2009; Sethi et al. 2012; Karam et al. 2019], where issues of robustness likely never occurred: these transforms operate on dense image contours whose sampling resolution matches the resolution of the image on which distance is being computed.

We cannot alter the exponential part of the kernel, because the exponential factor is the key to the asymptotic behavior that enables distance approximation. Kernels that decay slower — for example, kernels of the form $1/\|x - z\|^\lambda$ as used in *Shepard interpolation* — need even larger values of λ to achieve good distance, while suffering from the same drawbacks since they are essentially approximations of the exponential. Kernels that decay faster, such as the Gaussian $\exp(-\lambda\|x - z\|^2)$ suffer immensely from numerical instability, and anisotropic kernels alter the metric with which distance is measured.

Convolutional distance approximations to point-sampled curves and surfaces do converge to the correct solution as the sampling gets more and more dense, assuming no noise in the samples. However, accurately evaluating signed distance to noisy, partially-observed point clouds, at arbitrary points in space, remains an important problem to be solved, especially in vision and robotics.

The self-normalized variant of convolutional distance approximations (Equation 4.14) holds more promise, though it is still difficult to design a suitable function $g_x(z)$ to describe the local estimates to be blended together. For one, using the naive pseudonormal distance $g_x(z) = \langle x - z, n(z) \rangle$ common in the literature yields simple linear continuation of surfaces (inset). Building distance functions out of higher-order local approximations is a focus of current work (Section 6.1).



4.4 Relation to kernel methods

The self-normalized convolutional distance formula presented in Equation 4.14 is an example of a kernel density estimator, which finds broad use in regression tasks.

Broadly speaking, kernel density estimation uses *kernel smoothing* to estimate the unknown probability density function of a random variable: one makes inferences about a population by constructing a smooth interpolant or regressor through a finite number of data samples. This interpolant is constructed as a weighted average, where weights are determined by a kernel typically defined to assign higher weights to points closer in distance. For example, the *Nadaraya-Watson* estimator of a real-valued variable $f(x)$, using data points $\{p_i\}_{i=1}^N$, is defined as

$$\widehat{f}(x) := \frac{\sum_i f(p_i)k(x, p_i)}{\sum_i k(x, p_i)} \quad (4.15)$$

where the kernel $k(x, y)$ is often chosen to be a Gaussian kernel, and so has asymptotic behavior analogous to that of Equation 4.14. We can likewise think of convolutional distance approximations as a “nonparametric kernel method”, distinct from “model-based” methods that rely on, for example, surface fitting to point clouds (jet fitting, polynomial fitting, etc.)

In image processing, kernel methods appear as digital filters used for, for example, smoothing or denoising. Kernel methods are also used in *manifold learning* (a.k.a. *nonlinear dimensionality reduction*), where one assumes that some high-dimensional data lies on a lower-dimensional manifold, and tries to learn this manifold. More recently, kernel methods underpin the “attention” mechanism used in transformer neural network architecture [Vaswani et al. 2017]. These examples are applications of *kernel regression*, which specifically aims to find the relation between two random variables (input data and output).

While Section 4.3 showed why exponential asymptotics are undesirable for robust distance approximation, they can in fact be useful for clustering problems, since the incredibly strong attraction of the exponential kernel to minimizers aids the partition of data into discrete clusters. For example, applying *maximum log-likelihood estimation* to *Gaussian mixture models*, a method common in statistical learning, yields an iterative kernel density estimator in the form of Equation 4.15. Similarly, the SoftMax function is used as an activation function in classifier neural networks precisely because it is likely to pick a single “winning” category. The same principle underlies the method of *exponential tilting* and other sampling methods motivated by statistical mechanics.

In contrast, exponential asymptotics can be detrimental in non-clustering regression problems, which includes not just convolutional distance approximation but also many generative models. One such class of generative models aim to sample from a target probability distribution ρ_1 by transforming samples from an easy-to-sample distribution ρ_0 . *Diffusion models* in particular parameterize the transformation as a time-dependent differential equation

$$\frac{dz}{dt} = \frac{1}{t}z + \left(\frac{1}{t} - 1\right) \nabla \log \rho_t(z) \quad t \in (0, 1]$$

where z is a random variable that represents a sample $x \sim \rho_1$ that has been perturbed by some noise; one usually applies this model to training data to learn how to sample from the target distribution [Liu et al. 2022; Scarvelis et al. 2023].

When the source distribution is a normal distribution from which we sample Gaussian noise, at each timestep the transformed distribution is simply a mixture of (increasingly lower-variance) Gaussians centered at the training points, and the score function has a closed-form expression where

$$\nabla \log \rho_t(z) \propto \frac{\sum_{i=1}^N t x_i \exp\left(-\frac{1}{2} \frac{\|z - t x_i\|^2}{(1-t)^2}\right)}{\sum_{i=1}^N \exp\left(-\frac{1}{2} \frac{\|z - t x_i\|^2}{(1-t)^2}\right)} - z$$

where the first term can be seen as an approximation of $\operatorname{argmin}_{t x_i} \|z - t x_i\|^2$. Hence this expression represents a vector that points from the current sample z to an estimate of $\operatorname{argmin}_{t x_i} \|z - t x_i\|^2$, meaning the above ODE can be interpreted as an over-relaxation step that gradually pushes samples towards the closest training point.

Raw diffusion models can hence only “memorize” their training data, and cannot generalize effectively beyond their training data to generate novel results. In other words, just like how convolutional distance approximations behave on point clouds, the point samples making up the training data act as infinite attractors to which samples are inevitably drawn to, simply reproducing the distribution represented by the discrete training data, rather than the true distribution from which the training data is sampled, an observation made by many authors [Liu et al. 2022; Somepalli et al. 2022; Pidstrigach 2022; Yoon et al. 2023; Scarvelis et al. 2023; Carlini et al. 2023; Jain et al. 2024; Gu et al. 2025; Biroli et al. 2024]. The current state-of-the-art involves training an extremely expensive neural network to approximate the score function, trying to maintain a precarious balance between learning and overfitting.

In summary, many statistical methods fundamentally rely on the asymptotic behavior of the exponential function, whose strong attraction to extrema can be useful for some problems, but frustrating for others. Over the past several decades, researchers have tackled different versions of this problem in different settings: in the context of surface reconstruction, for instance, Kolluri [2008] derives sampling requirements under which the approximation converges, though doesn’t propose a robust algorithm; other authors suggest anisotropic kernels [Levin 1998; Adamson and Alexa 2006; Zagorchev and Goshtasby 2012], spatially-varying kernel bandwidths [Wang et al. 2008; Öztireli et al. 2009; Fuhrmann and Goesele 2014], hierarchical schemes [Ohtake et al. 2003], or regularized kernels [Chen et al. 2024]. The latter approach is similar to techniques encouraging generalization in generative models [Scarvelis et al. 2023], though adding noise to the model can easily lead to lower-quality output [Arjovsky et al. 2017]. The upshot is that these asymptotics are based on scalar diffusion: to gain robust behavior, we must develop more sophisticated strategies that, for example, use higher-order information.

CHAPTER 5

Signed distance

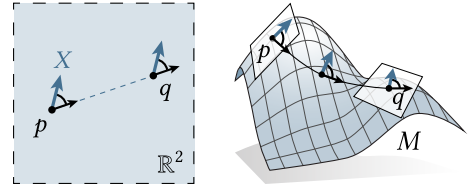
We now present our algorithm, the *signed heat method (SHM)*, for computing generalized signed distance directly from broken geometry, providing not only an inside/outside classification but also distance information. At a high level, the SHM is based on short-time heat diffusion, but introduces an intermediate step that avoids the diffusion-based pitfalls of convolutional distance approximations (Section 4.3). As a PDE-based approach, it also can compute (geodesic) distance on curved domains, applies to many discretizations (Figures 5.4, 5.6), and is robust to broken input with corrupted topology, geometry, and orientation (Figure 5.5) and to errors in the underlying domain (Figure 5.1). More broadly, its variational, PDE-based approach enables extensions not possible with other geodesic distance algorithms.

5.1 The asymptotics of vector diffusion

We consider an n -dimensional Riemannian manifold M with metric g , and want to compute the signed distance function ϕ for a submanifold $\Omega \subset M$.

We begin by observing a well-known property of the gradient of SDFs: at points $x \in M$ away from the cut locus of Ω , the gradient of an SDF equals the normal to Ω at the closest point to x on Ω . Motivated by this property, we consider *parallel transport* of vectors along shortest geodesics between points on Ω and points in M . In the plane, for example, the shortest geodesic $\gamma_{p \rightarrow q}$ between two points p and q is simply a straight line connecting the two points; parallel-transporting a tangent vector $X \in T_p \mathbb{R}^2$ along $\gamma_{p \rightarrow q}$ means transporting X so that it remains parallel to its initial state throughout its travel, and thus amounts to translating the initial vector X along $\gamma_{p \rightarrow q}$. On a Riemannian manifold M , we can similarly ask that a tangent vector $X \in T_p M$ maintain a constant angle with the tangent of the geodesic during transport (inset). (For more description of parallel transport, see Chapter 2.)

A remarkable fact is that *vector heat diffusion* yields parallel transport along shortest geodesics, as diffusion time goes to zero. On $\mathbb{R}^d$, this relation is intuitive: vector heat diffusion is equivalent



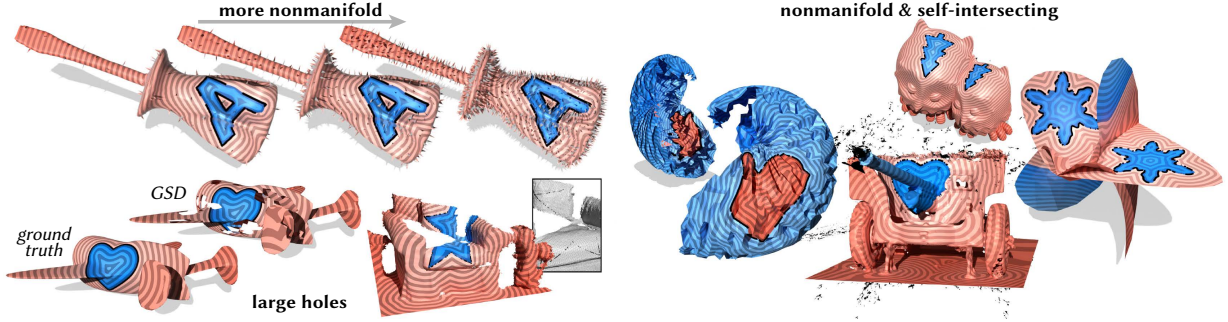


Fig. 5.1: Our method is robust not only to errors in the source geometry, but also in the domain mesh itself. Here we obtain well-behaved SDFs even for meshes found “in the wild,” such as amateur-created 3D scans [Choi et al. 2016]. Even in cases where a notion of inside and outside is meaningless (such as the rightmost mesh), our method fails gracefully — still producing a good signed distance approximation near the input curve.

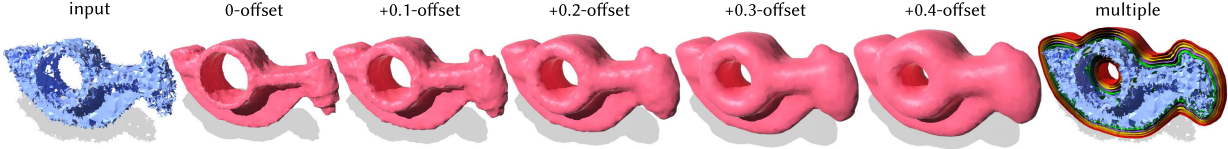


Fig. 5.2: By extracting level sets of generalized signed distance, we can convert broken, noisy, and nonmanifold input geometry (*far left*) into closed, regular, manifold surfaces and evenly-spaced offset surfaces.

to simply applying the scalar heat diffusion component-wise to an initial vector field, meaning vectors get translated throughout $\mathbb{R}^d$, while their magnitudes decay according to the scalar heat kernel $k_t(x, y) = (4\pi t)^{-d/2} \exp(-\|x - y\|^2/4t)$. On manifolds, where the tangent space at every point locally looks Euclidean, the vector heat kernel has the asymptotic expansion

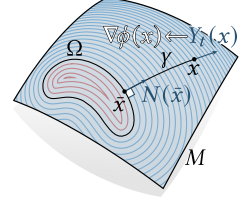
$$k_t^\nabla \sim (4\pi t)^{-1} \exp(-\text{dist}(x, y)^2/4t) c(x, y) \sum_{i=0}^{\infty} t^i \Phi_i^\Delta(x, y)$$

where the magnitude is governed by the same exponential decay that describes the scalar heat kernel in Euclidean space, and the leading term $\Phi_0^\Delta(x, y)$ is the parallel transport map $P_{\gamma_{x \rightarrow y}}$ that takes vectors at a point x to vectors at another point y along the shortest geodesic $\gamma_{x \rightarrow y}$ connecting the two ($c(x, y)$ is a surface-related scalar-valued term) [Berline et al. 1992, Theorem 2.30].

Hence the first step of our algorithm is to diffuse the normals N of Ω to the rest of the domain M for a short time $t > 0$, which extends information about surface orientation to the rest of the domain. In particular we solve a vector-valued diffusion equation

$$\begin{aligned} \frac{d}{dt} X_t &= \Delta^\nabla X_t, \quad t > 0, \\ X_0 &= N\mu_\Omega \end{aligned} \tag{5.1}$$

where Δ^∇ denotes the negative-definite *connection Laplacian* on M , and $N\mu_\Omega$ is a vector-valued measure equal to zero away from Ω , and determined by N for points in Ω ¹. As $t \rightarrow 0$, the diffused vector $X_t(x)$ at each point $x \in M$ aligns with the vector obtained via parallel transport of the normal $N(\bar{x})$ at the closest point $\bar{x} \in \Omega$, along a minimal geodesic γ . Since parallel transport along geodesics preserves tangency, this vector will be tangent to γ itself — and since traveling along γ is the quickest way back to Ω , it must be parallel to the unsigned distance gradient $\nabla\phi$. Moreover, since we transport *oriented* normals, we get the correct sign. We can hence normalize X_t to obtain an approximation $Y_t := X_t/\|X_t\|$ of the signed distance gradient.



The vector field Y_t will not describe exact gradients for any SDF, due to both the diffusion approximation — and more significantly — errors in the input. However, we can still look for the function ϕ whose gradient is as close as possible, in a least-squares sense, to Y_t . In particular, we seek a minimizer for the problem

$$\min_{\phi: M \rightarrow \mathbb{R}} \int_M \|\nabla\phi - Y_t\|^2. \quad (5.2)$$

Using integration by parts, one can show that a minimizer satisfies the Poisson equation

$$\begin{aligned} \Delta\phi &= \nabla \cdot Y_t \text{ on } M \\ \frac{\partial\phi}{\partial n} &= n \cdot Y_t \text{ on } \partial M, \end{aligned} \quad (5.3)$$

where Δ denotes the negative-definite Laplace-Beltrami operator on M . The solution of Equation 5.3 is determined up to a constant shift, leaving us the freedom to enforce either exact or approximate interpolation of Ω by a level set of ϕ (Section ??).

In summary, we arrive at the following algorithm:

1. Solve a vector diffusion equation $\frac{d}{dt}X_t = \Delta^\nabla X_t$ (Equation 5.1), which diffuses the normals of Ω for a small time $t > 0$.
2. Evaluate the vector field $Y_t = X_t/\|X_t\|$, yielding a unit vector field that approximates the gradient of the (unknown) SDF.
3. Solve a Poisson equation $\Delta\phi = \nabla \cdot Y_t$ (Equation 5.3), to find the function whose gradient best matches Y_t .

A diffusion-based approach is valuable because it extends in a robust way to imperfect geometry. For instance, if a curve has gaps, or a surface has holes, diffusion averages together normals at nearby points, providing smooth interpolation of the observed data. Normalization

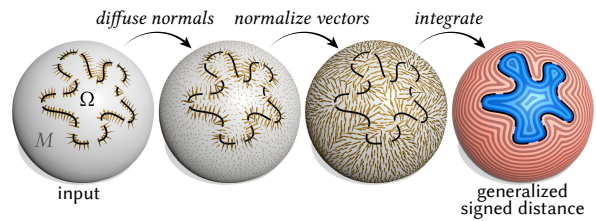


Fig. 5.3: The three steps of the signed distance method.
¹Intuitively, we use μ_Ω to denote a measure concentrated on Ω , similar in spirit to an indicator function. More formally, for any Borel measurable set $U \subset M$, $\mu_\Omega(U) := \int_{\Omega \cap U} dV$, where dV is the usual volume measure on Ω .

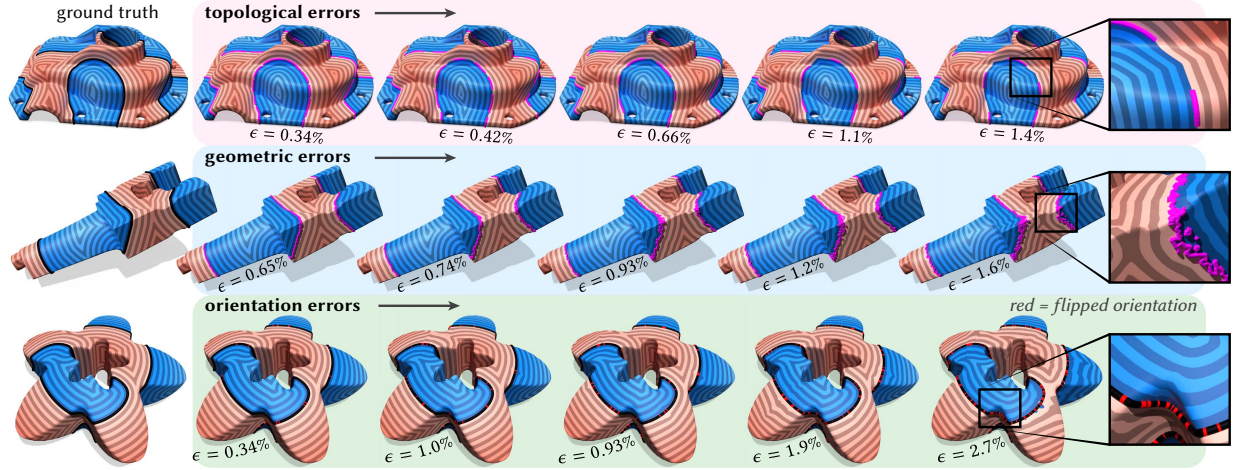


Fig. 5.5: Our method provides robust and reliable signed distance approximation, failing gracefully in the presence of significant topological, geometric, or orientation errors. Errors ϵ in geodesic distance are displayed relative to the exact polyhedral SDF of a finely sampled version of the original curve.

of gradients then ensure that we recover a distance approximation, rather than just a smooth function.

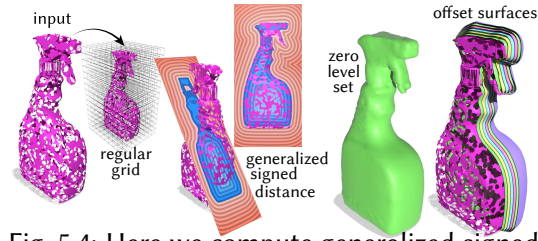


Fig. 5.4: Here we compute generalized signed distance to a badly broken surface (left) by solving on a regular grid in $\mathbb{R}^3$. Contouring this function yields well-behaved and evenly-spaced offset surfaces (right).

ity, while achieving especially good reconstruction by diffusing vector-valued rather than scalar-valued data. And while the asymptotic behavior of short-time vector diffusion means that in the limit $t \rightarrow 0$, points simply inherit the normal at the closest point like in pseudonormal distance (Section 4.3), in practice we take finite diffusion times $t > 0$ that yield well-behaved shape completions.

5.2 Algorithm

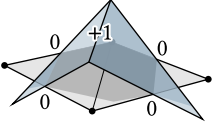
The signed heat method mainly involves solving two sparse linear systems, corresponding to Steps 1 and 3.

The SHM ultimately relies on the exponential asymptotics of heat kernel, much in the same way as convolutional distance approximations (Chapter 4). However, whereas convolutional distance approximations tightly couple both distance accuracy and reconstruction, and thus sacrifice their quality in both (Section 4.3), the SHM instead achieves good quality in both. The SHM’s intermediate step of normalizing gradient vectors, like the original heat method for unsigned distance [Crane et al. 2013b], partially decouples distance and reconstruction quality.

Time discretization. As in past heat methods we discretize Equation 5.1 in time via one step of backward Euler [Crane et al. 2013b; Sharp et al. 2019c], and solve a linear equation

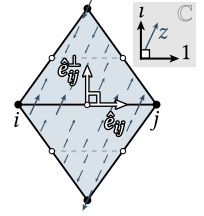
$$(\text{id} - t\Delta^\nabla)X_t = X_0 \quad (5.4)$$

for a single, fixed time $t > 0$ (where id is the identity).



Spatial discretization on triangle meshes. On triangle meshes, we use edge-based operators for the vector diffusion step (Step I), which makes it straightforward to discretize curve sources: we don't have to map from the tangent space of curve segments, which generically lie within triangle faces, to the tangent space of vertices. We can instead discretize within single face using only intrinsic operations, opening the possibility of further improving accuracy and robustness using *intrinsic Delaunay refinement* [Gillespie et al. 2021].

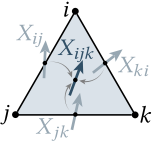
We use *Crouzeix-Raviart (CR) basis functions*, which associate each edge $ij \in E$ with a facewise linear function $\varphi_{ij} : M \rightarrow \mathbb{R}$ interpolating the value 1 at the midpoint of ij , and 0 at all other edge midpoints (inset). A corresponding basis for vector fields is expressed by identifying tangent vectors with complex numbers. In particular, at each edge ij we choose a coordinate system such that 1 and the imaginary unit i correspond to $\hat{e}_{ij}$ and $\hat{e}_{ij}^\perp$, *resp.* The function $\psi_{ij} := \varphi_{ij} + 0i$ then defines a basis vector field parallel to the edge, and z/ψ_{ij} describes a locally supported vector field parallel to $z \in \mathbb{C}$ (right inset).



The discrete algorithm amounts to solving two sparse linear systems. First we solve the discrete vector heat equation, which uses the discrete Crouzeix-Raviart connection Laplacian L^∇ and mass matrix M ,

$$(M + tL^\nabla)X = X_0, \quad (5.5)$$

obtaining a diffused vector field X . Following Sharp et al. [2019c, Section 7.3], we let $t = h^2$, where h is the mean distance between nodes — in our case, edge midpoints, yielding half the mean edge length.



Next, we average the diffused vectors X to each face $ijk \in F$ via $X_{ijk} := (X_{ij} + X_{jk} + X_{ki})/3$ (taking care to express all vectors in the same basis), and compute unit vectors $Y_{ijk} := X_{ijk}/\|X_{ijk}\|$ which represent the gradient of our (generalized) SDF. Finally, to obtain the SDF $\phi \in \mathbb{R}^{|V|}$ at vertices, we solve a sparse linear system

$$C\phi = b \quad (5.6)$$

where C is the cotan Laplacian, $b \in \mathbb{R}^{|V|}$ is a vector of discrete divergences.

Additional discretizations. Since the method is based purely on intrinsic PDEs, these equations can be applied not just on triangle meshes, but virtually any data structure, in any dimension: all we need is a divergence operator and Laplacian defined on the data structure, and we simply take advantage of Laplacians developed in previous work [Bunge et al. 2020; Coeurjolly and Lachaud 2022; Sharp and Crane 2020a; Sharp et al. 2019c].

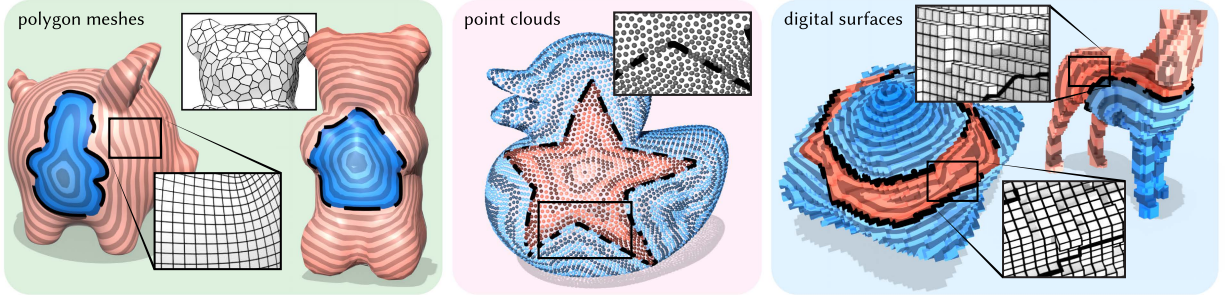


Fig. 5.6: Our method extends to polygon meshes, point clouds, and digital surfaces. Digital surface meshes are from Coeurjolly and Levallois [2015].

5.3 Results

Our method applies out of the box to nonmanifold and nonorientable meshes (inset), since all our differential operators are local and defined per-face, and hence oblivious to any nonmanifold features.

One natural use case for generalized signed distance is to contour broken geometry, and generate accurate fixed-distance offsets (Figures 5.12, 5.2, 5.4). Generating such offsets from imperfect geometry is useful, for example, for 3D printing, or for downstream mesh processing tasks that require closed or manifold surfaces. One can also “inflate” or “shrink” shapes by taking positive or negative offsets – and combining these two operations in sequence can be used to simplify high-frequency features of broken shapes as if they were whole (Figure 5.7).

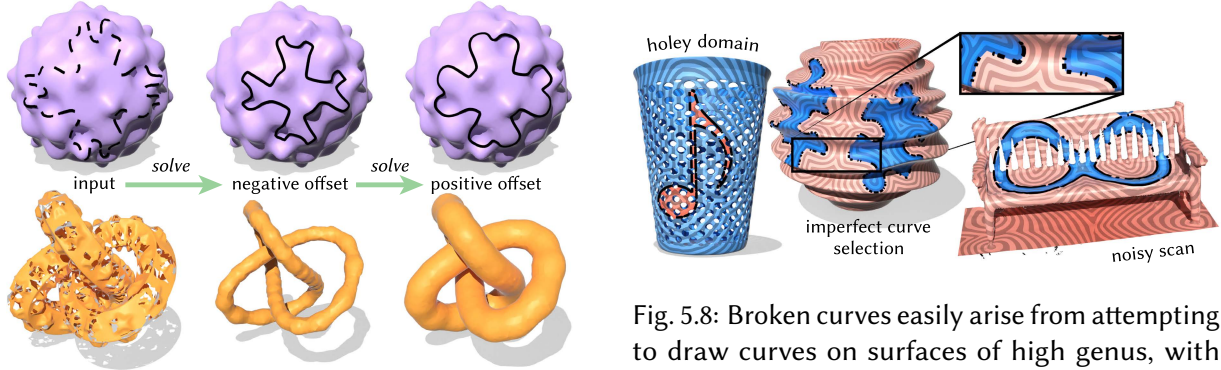


Fig. 5.7: One can simplify high-frequency features of broken curves and surfaces by taking consecutive positive/negative offsets of a generalized signed distance function, akin to dilation/erosion.

Fig. 5.8: Broken curves easily arise from attempting to draw curves on surfaces of high genus, with overhangs, and with holes and scanner noise. Our method yields signed distance functions robust to these challenges. (Scanned bench from Choi et al. [2016].)

Accuracy and performance. We compare against the unsigned heat method (UHM) of Crane et al. [2013b], which provides a useful reference point since nearly all work on geodesic distance algorithms from the past decade compares against this method. As a more recent reference

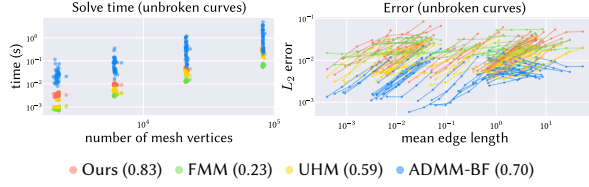


Fig. 5.9: We observe approximately linear convergence in distance accuracy on a benchmark of unbroken (closed) curves on 44 different meshes. The legend shows median orders of accuracy. Note that if we omit the zero set constraint, enabling us to re-use both factorizations, our method and UHM become 1–2 orders of magnitude faster.

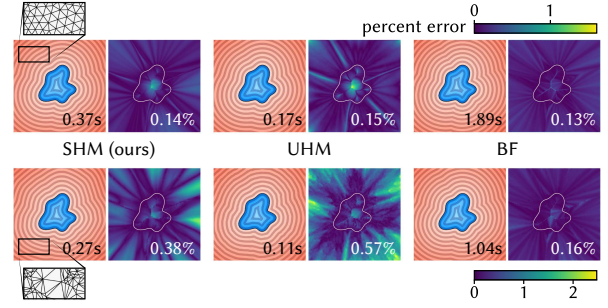


Fig. 5.10: Distribution of error in distance approximation for a perfect, unbroken curve on a high- and low-quality mesh (*top/bottom*). Both meshes have about 100k faces. Inset numbers on SDF and error plots indicate compute time and mean error (*resp.*). Overall our method is quite comparable to the original heat method, and less accurate than BF on the low-quality mesh — but about 4–5x faster.

point, we also compare with the convex formulation of [Belyaev and Fayolle \[2020\]](#) (labeled BF). Since neither method directly handles curve sources, we either integrate the initial scalar heat distribution against hat functions (for UHM) or simply use the set of curve vertices as the source set (for BF). (Methods that directly handle curve sources do not have an open source implementation [[Bommes and Kobbelt 2007](#)], or do not include curve sources in their public release [[Trettner et al. 2021](#)].) Finally, since BF must constrain the zero set, we impose the same constraints on UHM/SHM, and do not pre-factor any matrices. Note, however, that for multiple source terms, heat methods can achieve about two orders of magnitude speedup by omitting factorization [[Crane et al. 2013b](#), Table 1].

Planar domains. As noted by [Crane et al. \[2013b, Figure 21\]](#), even exact polyhedral distance (including MMP) provides only a 2nd-order accurate estimate of true (smooth) geodesic distance, due to errors in the approximation of the domain itself. To avoid conflating these two sources of error, we first consider closed, planar curves — where the exact SDF is easily computed via closest-point queries [[Sawhney et al. 2020](#)], and sign can unambiguously be determined via standard inside/outside tests [[Haines 1994b](#)]. As seen in Figure 5.10, our method is slightly slower but slightly more accurate than UHM. Without sharpening, it is not as accurate as BF— but is an order of magnitude faster. Moreover, BF must trade off between bias near the boundary [[Edelstein et al. 2023, Figure 3, left](#)], or distortion in the presence of curve sources, depending on whether Hessian regularization is omitted or included (*resp.*).

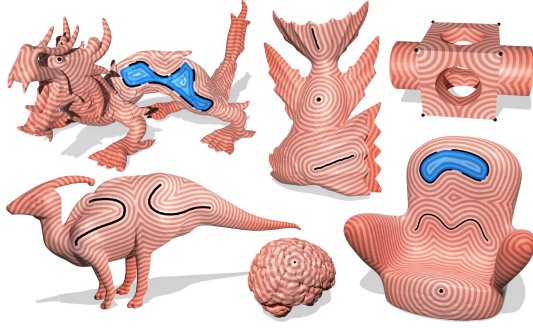


Fig. 5.11: We can mix and match signed and unsigned distance, selectively treating open curves as either broken region boundaries, or as literal open curves. We can also incorporate distance to isolated points.

Surface domains. We next consider closed curves on surface meshes. Here we can no longer obtain the true distance on an unknown underlying smooth surface; we hence compute “ground truth” distance as the exact polyhedral distance to a finely-sampled version of the input curve (100 samples per curve edge) using MMP [Mitchell et al. 1987], which itself has $O(h^2)$ error. We plot convergence and solve times in Figure 5.9. We observe that our method has approximately linear convergence in mean edge length, with better consistency on curve sources compared to other methods. We find the same trend in solve times as in Section 5.3.

Completion of broken geometry. Finally, we compare against the end-to-end pipeline of repairing broken geometry, computing unsigned distance to the fixed geometry, and signing the unsigned distance. In particular, we use the surface winding numbers (SWN) introduced in Chapter 3 to contour broken curves, and compute exact polyhedral distance using MMP [Mitchell et al. 1987] using the curve vertices as the source set. As input surface domains, we use the meshes with $\sim 5k$ vertices from the same dataset as above. As input curves, we take level sets of five different low-frequency Laplacian eigenfunctions, and add geometric and topological errors by taking the union of the curves with their offsets (found by taking boundaries of triangle strips), and deleting about 50% of the curve at random intervals.

The repair-distance-sign pipeline is particularly sensitive to errors in the input, since any errors made during contouring are permanent and will destroy the quality of the final SDF no matter how accurate the subsequent distance computation. In particular, contouring the winding number is notoriously difficult, and often leads to misclassified regions (Figure 5.14). Though Section ?? suggests a rounding procedure, it remains unclear *which* half-integer level set to take as the boundary between inside and outside; we use the values of the rounded winding number function that appear most often along the input curves, though this results in 12% of examples with $>50\%$ of surface area misclassified. We also try contouring winding number according to the average of the winding number function along the input curves, but we find similar results. In contrast, our method achieves greater robustness

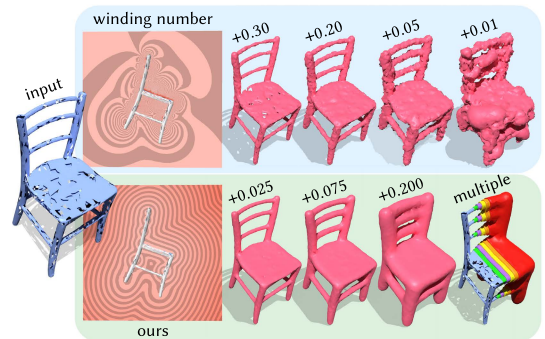


Fig. 5.12: *Top:* Generalized winding number (GWN) cannot be used for offset surfaces, since it provides only a smooth indicator function — and not a signed distance. *Bottom:* Our generalized signed distance (GSD) provides much nicer offsets on the same broken mesh.

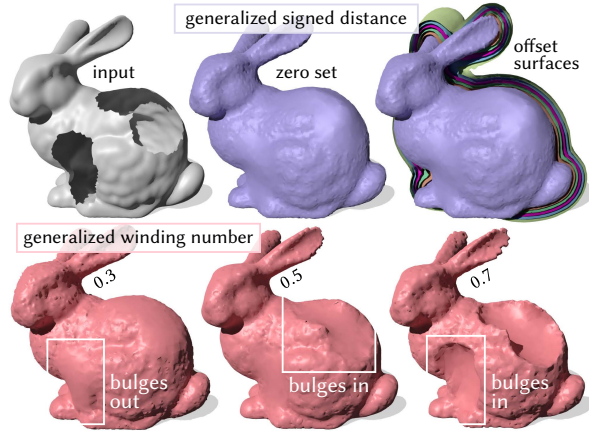


Fig. 5.13: GWN completes surfaces with saddle-shaped harmonic patches that exhibit poor normal continuity with the observed geometry (across many contour values). Our method directly incorporates normal information, providing more plausible reconstruction even for large holes.

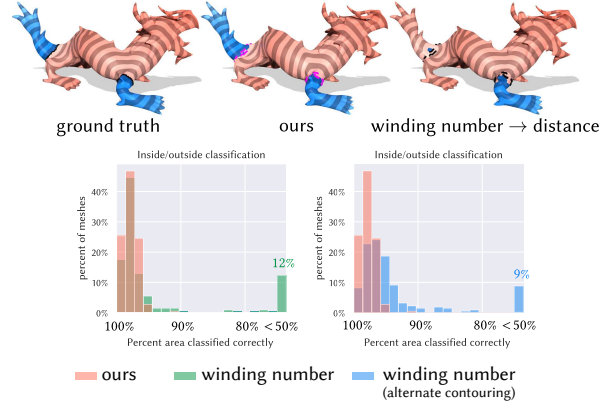


Fig. 5.14: We compare, on 220 examples, the accuracy of GSD versus a hybrid scheme that repairs broken geometry using winding numbers, computes unsigned distance to the repaired geometry, then signs the unsigned distance. Winding numbers are worse at classifying inside/outside, regardless of contouring method. As a result, the hybrid scheme yields about 3x lower distance accuracy on average (0.11 vs. 0.04 L_2 error, *resp.*), even though it benefits from exact distance (via MMP).

by averaging normal vectors over the whole domain. The repair-distance-sign pipeline is also about 10x more expensive due to its intermediate steps; SWN alone has cost asymptotically equivalent the SHM.

As seen in Figure 5.13 we also obtain more natural surface completion than GWN, which for general surfaces is hard to contour with any single level set value. Similar to spline interpolation, the zero set of GSD nicely matches both positions and normals along hole boundaries. Here meshes for both methods are extracted using the marching tetrahedra implementation in *libigl* [Jacobson et al. 2018].

CHAPTER 6

Proposed Work

Section 6.1 proposes a pointwise evaluation method for computing generalized signed distance similar to the output of [Feng and Crane \[2024\]](#). In Section 6.2, I present my planned timeline to finish this work in time for graduation in spring 2026.

6.1 Instant generalized signed distance

The robustness of the Signed Heat Method in Chapter 5 comes from solving a global problem, in particular by making use of a well-behaved, globally-supported extension of higher-order information about the input geometry. Currently we solve this global problem by solving a global linear system defined on a spatial discretization of the domain: this computational paradigm is especially effective for dense evaluations of generalized signed distance — meaning simultaneous evaluation at many query points throughout the domain — since it makes use of spatial coherence between query points, and allows improved amortized performance over repeated solves re-using the same matrix factorizations.

On the other hand, many applications in computer vision, simulation, and rendering would benefit from an output-sensitive algorithm for generalized signed distance that answers isolated pointwise queries, where each pointwise query is optimized to be as fast as possible. Since we wish to support queries at arbitrary points, this latter paradigm also implies no spatial discretization of the domain, which would be especially beneficial for the large-scale, intricate scenes often encountered in vision applications — where meshing would be especially memory- and compute-heavy, perhaps even infeasible.

Unfortunately, directly adapting the Signed Heat Method to support efficient pointwise queries proves to be impossible. At first, it appears the Signed Heat Method boils down to solving the Poisson problem

$$\Delta\phi = \nabla \cdot Y_t,$$

where $Y_t := X_t/\|X_t\|$ is the normalized version of the vector field solution obtained from the initial vector diffusion step. It’s tempting to turn this Poisson equation into the corresponding integral equation to gain the computational flexibility we ask for, but Green’s representation

formula can only be extended to infinite regions if the solution decays to zero at infinity [Brebbia et al. 1984, §2.10], whereas a distance function goes to infinity at infinity. Likewise, the integral equation

$$\phi(x) = \int_{\partial M} \left[\frac{\partial G(x, z)}{\partial n(z)} \phi(z) - G(x, z) \frac{\partial \phi}{\partial n}(z) \right] dz + \int_M G(x, y) (\nabla \cdot Y_t)(y) dy,$$

while valid on any finite domain M using an appropriate fundamental solution $G(x, y)$, is recursive in ϕ : the first term assumes we already know the value of the solution ϕ on the domain boundary ∂M . (In contrast, the convolutional distance formulas in Chapter 4 do exist as boundary integrals, because through exponentiation the solution does have proper decay at infinity.)

While the convolutional distance approximations introduced in Chapter 4 exhibit the performance benefits we desire, unfortunately Section 4.3 showed that they do not provide generalized signed distance to point clouds. Below are two approaches I am currently pursuing to extend these methods to ones that do produce generalized signed distance:

Local fitting of geometric proxies. Another interpretation of convolutional formulas is that they give a good global estimate as long as the local estimates they blend together — in particular, the ones represented by h in Equation 4.14 — are accurate. The self-normalized convolutional formula in Equation 4.14 is simply a general-purpose method of smoothly blending together local function estimates (the h 's) into a smooth global function. So we can solve our problem if we compute the local approximations really well; we also sidestep the issue of how to set λ 's value, since if we are confident the local estimates are accurate, we should simply set λ as large as possible.

Our first experiments in this vein fit torii to neighborhoods around each point, where each torus locally approximates the surface while representing a geometry proxy to which it is easy to compute distance (Figure 6.1).

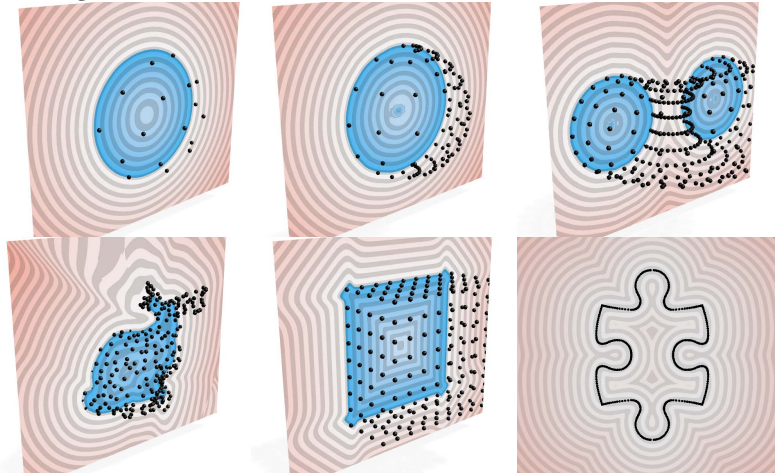


Fig. 6.1: Experiments, implemented in a shader, of blending together distance approximations to torii.

The results are promising, but it is also clear that the method of local approximation needs to be improved to handle sharp features and other complex geometry. There is a plethora of classical statistical tools for data fitting, but picking just one method feels risky: for any estimator, one can probably find a class of data for which that particular estimator doesn't work well. Thus, next steps focus on improving local data-fitting by learning from data.

Learning spatially-varying kernel bandwidths. Another approach to generalizing convolutional distance approximations is to compute spatially-varying kernel bandwidths that ideally interpolate points if the query point is close to the input point cloud, or where the point cloud is sparsely sampled, while using smaller bandwidth (higher λ) further away or in regions of high sample density. For now, I think this approach is less promising than the former, since it necessarily sacrifices distance accuracy.

6.2 Timeline

Below is a timeline for my proposed work, culminating with an expected thesis defense in spring of 2026.

OCTOBER 2025 – JANUARY 2026: Finish ongoing work on instant signed distance, and submit to SIGGRAPH 2026.

JAN. 2026 – APRIL 2026: Continue investigating “maximum distance functions”, a topic I started in Spring 2025, and finish an unrelated project on curve reconstruction (and possibly a project on neural Laplacians).

APRIL 2026 – MAY 2026: Write thesis and prepare thesis presentation.

MAY 2026: Defend thesis.

CHAPTER 7

Bibliography

- Rémi Abgrall. 2022. Evaluating a distance function. arXiv:2211.02319 [math.NA] <https://arxiv.org/abs/2211.02319> 4.2
- Anders Adamson and Marc Alexa. 2006. Anisotropic point set surfaces. In *Proceedings of the 4th International Conference on Computer Graphics, Virtual Reality, Visualisation and Interaction in Africa* (Cape Town, South Africa) (*AFRIGRAPH '06*). Association for Computing Machinery, New York, NY, USA, 7–13. <https://doi.org/10.1145/1108590.1108592> 4.4
- M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, and C.T. Silva. 2001. Point set surfaces. In *Proceedings Visualization, 2001. VIS '01*. 21–29, 537. <https://doi.org/10.1109/VISUAL.2001.964489> 4.2, 4.3
- Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70* (Sydney, NSW, Australia) (*ICML '17*). JMLR.org, 214–223. 4.4
- J Andreas Bærentzen. 2005. Robust generation of signed distance fields from triangle meshes. In *Fourth International Workshop on Volume Graphics, 2005*. IEEE, 167–239. 2.4
- Alexander Belyaev and Pierre-Alain Fayolle. 2020. An ADMM-based scheme for distance function approximation. *Numerical Algorithms* 84 (2020), 14 pages. <https://doi.org/10.1007/s11075-019-00789-5> 5.3
- Alexander Belyaev and Pierre-Alain Fayolle. 2024. Accuracy Improvements for Convolutional and Differential Distance Function Approximations. arXiv:2412.09200 [math.NA] <https://arxiv.org/abs/2412.09200> 4.1, 4.2
- Alexander G Belyaev and Pierre-Alain Fayolle. 2015. On variational and PDE-based distance function approximations. In *Computer Graphics Forum*, Vol. 34. Wiley Online Library, 104–118. 4.1
- Carl M. Bender and Steven A. Orszag. 1999. *Advanced Mathematical Methods for Scientists and Engineers I: Asymptotic Methods and Perturbation Theory*. Springer New York, NY. <https://doi.org/https://doi.org/10.1007/978-1-4757-3069-2> 4.2, A.2

- Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Joshua Levine, Andrei Sharf, and Claudio Silva. 2014. State of the Art in Surface Reconstruction from Point Clouds. In *EUROGRAPHICS star report (EUROGRAPHICS star report, Vol. 1)*. Strasbourg, France, 161–185. <https://doi.org/10.2312/egst.20141040> 2.3
- Nicole Berline, Ezra Getzler, and Michèle Vergne. 1992. *Heat Kernels and Dirac Operators* (1 ed.). Springer Berlin, Heidelberg. 5.1
- Jack Binysh and Gareth P Alexander. 2018. [Maxwell’s theory of solid angle and the construction of knotted fields](#). *Journal of Physics A: Mathematical and Theoretical* 51, 38 (2018), 385202. 2.3.3
- Giulio Biroli, Tony Bonnaire, Valentin de Bortoli, and Marc Mézard. 2024. Dynamical Regimes of Diffusion Models. arXiv:2402.18491 [cs.LG] <https://arxiv.org/abs/2402.18491> 4.4
- James F. Blinn. 1982. A Generalization of Algebraic Surface Drawing. *ACM Trans. Graph.* 1, 3 (July 1982), 235–256. <https://doi.org/10.1145/357306.357310> 4.3
- Jean-Daniel Boissonnat and Frédéric Cazals. 2002. Smooth surface reconstruction via natural neighbour interpolation of distance functions. *Computational Geometry* 22, 1 (2002), 185–203. [https://doi.org/10.1016/S0925-7721\(01\)00048-7](https://doi.org/10.1016/S0925-7721(01)00048-7) 16th ACM Symposium on Computational Geometry. 4.2, 4.3
- David Bommes and Leif Kobbelt. 2007. Accurate Computation of Geodesic Distance Fields for Polygonal Curves on Triangle Meshes. *VMV*, 151–160. 2.4, 5.3
- C.A. Brebbia, J.C.F. Telles, and L. Wrobel. 1984. *Boundary Element Techniques: Theory and Applications in Engineering*. Springer Berlin Heidelberg. 6.1
- Alan Brunton and Lubna Abu Rmaileh. 2021. Displaced Signed Distance Fields for Additive Manufacturing. *ACM Trans. Graph.* 40, 4, Article 179 (jul 2021), 13 pages. <https://doi.org/10.1145/3450626.3459827> 2.4
- Astrid Bunge, Philipp Herholz, Misha Kazhdan, and Mario Botsch. 2020. Polygon Laplacian Made Simple. *Computer Graphics Forum* 39, 2 (2020), 303–313. <https://doi.org/10.1111/cgf.13931> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/cgf.13931> 5.2
- Fatih Calakli and Gabriel Taubin. 2011. SSD: Smooth Signed Distance Surface Reconstruction. In *Computer Graphics Forum*, Vol. 30. Wiley Online Library, The Eurographics Association and Blackwell Publishing Ltd., 1993–2002. <https://doi.org/10.1111/j.1467-8659.2011.02058.x> 2.4
- Nicholas Carlini, Jamie Hayes, Milad Nasr, Matthew Jagielski, Vikash Sehwal, Florian Tramèr, Borja Balle, Daphne Ippolito, and Eric Wallace. 2023. Extracting Training Data from Diffusion Models. arXiv:2301.13188 [cs.CR] <https://arxiv.org/abs/2301.13188> 4.4
- J. C. Carr, R. K. Beatson, J. B. Cherrie, T. J. Mitchell, W. R. Fright, B. C. McCallum, and T. R. Evans. 2001. Reconstruction and representation of 3D objects with radial basis functions. In *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’01)*. Association for Computing Machinery, New York, NY, USA, 67–76. <https://doi.org/10.1145/382640.382658>

- [//doi.org/10.1145/383259.383266](https://doi.org/10.1145/383259.383266) 4.3
- Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niessner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. 2017. *Matterport3d: Learning from rgb-d data in indoor environments*. *arXiv:1709.06158* (2017). 2.3.3
- Hanyu Chen, Bailey Miller, and Ioannis Gkioulekas. 2024. 3D Reconstruction with Fast Dipole Sums. *ACM Trans. Graph.* 43, 6, Article 192 (Nov. 2024), 19 pages. <https://doi.org/10.1145/3687914> 4.1, 4.3, 4.4
- Vladimir Chernov and Yuli B. Rudyak. 2009. *On generalized winding numbers*. *St. Petersburg Mathematical Journal* 20, 5 (2009), 837–849. 2.3.3
- Cheng Chi and Shuran Song. 2021. *GarmentNets: Category-Level Pose Estimation for Garments via Canonical Space Shape Completion*. In *The IEEE International Conference on Computer Vision (ICCV)*. 2.3.3
- David RJ Chillingworth. 1972. *Winding numbers on surfaces, I*. *Math. Ann.* 196, 3 (1972), 218–249. 2.3.3
- Sungjoon Choi, Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. 2016. A Large Dataset of Object Scans. *arXiv:1602.02481* (2016). 3.11, 5.1, 5.8
- David Coeurjolly and Jacques-Olivier Lachaud. 2022. A Simple Discrete Calculus for Digital Surfaces. In *IAPR Second International Conference on Discrete Geometry and Mathematical Morphology*, Étienne Baudrier, Benoît Naegel, Adrien Krähenbühl, and Mohamed Tajine (Eds.). Springer, LNCS. 5.2
- David Coeurjolly and Jérémy Levallois. 2015. VolGallery. <https://github.com/dcoeurjo/VolGallery>. 5.6
- Julian D Cole. 1951. On a quasi-linear parabolic equation occurring in aerodynamics. *Quarterly of applied mathematics* 9, 3 (1951), 225–236. 4.1
- Alvaro Collet, Ming Chuang, Pat Sweeney, Don Gillett, Dennis Evseev, David Calabrese, Hugues Hoppe, Adam Kirk, and Steve Sullivan. 2015. *High-quality streamable free-viewpoint video*. *ACM Trans. Graph.* 34, 4 (2015), 1–13. 2.3.3
- Michael G Crandall and Pierre-Louis Lions. 1983. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American mathematical society* 277, 1 (1983), 1–42. 4.1
- Keenan Crane, Fernando de Goes, Mathieu Desbrun, and Peter Schröder. 2013a. *Digital Geometry Processing with Discrete Exterior Calculus*. In *ACM SIGGRAPH 2013 courses* (Anaheim, California) (*SIGGRAPH '13*). ACM, New York, NY, USA, 126 pages. 2.2, 2.2
- Keenan Crane, Clarisse Weischedel, and Max Wardetzky. 2013b. Geodesics in heat: A new approach to computing distance based on heat flow. *ACM Transactions on Graphics (TOG)* 32, 5 (2013), 1–11. 4.1, 5.1, 5.2, 5.3
- Mathieu Desbrun, Eva Kanso, and Yiyang Tong. 2006. *Discrete differential forms for computational modeling*. In *ACM SIGGRAPH 2006 Courses*. 39–54. 2.2

- Manfredo P. do Carmo. 1992. *Riemannian Geometry* (1 ed.). 2.2
- Manfredo P. Do Carmo. 2016. *Differential Geometry of Curves & Surfaces* (second ed.). Dover, New York. 2.3.3
- Jan Dvořák, Zuzana Káčereková, Petr Vaněček, Lukáš Hruša, and Libor Váša. 2022. [As-rigid-as-possible volume tracking for time-varying surfaces](#). *Computers & Graphics* 102 (2022), 329–338. <https://doi.org/10.1016/j.cag.2021.10.015> 2.3.3
- Michal Edelstein, Nestor Guillen, Justin Solomon, and Mirela Ben-Chen. 2023. A Convex Optimization Framework for Regularized Geodesic Distances. In *ACM SIGGRAPH 2023 Conference Proceedings* (Los Angeles, CA, USA) (SIGGRAPH '23). Association for Computing Machinery, New York, NY, USA, Article 2, 11 pages. <https://doi.org/10.1145/3588432.3591523> 5.3
- Jeff Erickson and Kim Whittlesey. 2005. [Greedy optimal homotopy and homology generators](#). In *SODA*, Vol. 5. 1038–1046. 2.3.2, 3.3
- Leonhard Euler. 1781. De mensura angulorum solidorum. *Acta Academiae Scientiarum Imperialis Petropolitanae* (1781), 31–54. 2.3.3
- L.C. Evans. 1998. *Partial Differential Equations*. American Mathematical Society. 4.1, 4.2
- Sergei Fedotov. 1999. Wave front for a reaction-diffusion system and relativistic Hamilton-Jacobi dynamics. *Phys. Rev. E* 59 (May 1999), 5040–5044. Issue 5. <https://doi.org/10.1103/PhysRevE.59.5040> A.2
- Nicole Feng and Keenan Crane. 2024. A Heat Method for Generalized Signed Distance. *ACM Trans. Graph.* 43, 4, Article 92 (jul 2024), 16 pages. <https://doi.org/10.1145/3658220> 6
- Nicole Feng, Mark Gillespie, and Keenan Crane. 2023. Perspectives on Winding Numbers. *ACM Transactions on Graphics (Supplemental)* (2023). 2.3.3
- Wendell H Fleming and Panagiotis E Souganidis. 1986. PDE-viscosity solution approach to some problems of large deviations. *Annali della Scuola Normale Superiore di Pisa-Classe di Scienze* 13, 2 (1986), 171–192. A.2
- M. I. Freidlin. 1986. Geometric Optics Approach to Reaction-Diffusion Equations. *SIAM J. Appl. Math.* 46, 2 (1986), 222–232. <https://doi.org/10.1137/0146016> arXiv:<https://doi.org/10.1137/0146016> A.2
- Simon Fuhrmann and Michael Goesele. 2014. Floating scale surface reconstruction. *ACM Trans. Graph.* (2014). 4.4
- Carl Friedrich Gauss. 1838. General Theory of Terrestrial Magnetism. (1838). <https://hgss.copernicus.org/articles/5/11/2014/hgss-5-11-2014.pdf> 2.3.3
- Mark Gillespie, Nicholas Sharp, and Keenan Crane. 2021. Integer Coordinates for Intrinsic Geometry Processing. *ACM Trans. Graph.* 40, 6 (2021). 3.3, 5.2
- Cindy M Goral, Kenneth E Torrance, Donald P Greenberg, and Bennett Battaile. 1984. [Modeling the interaction of light between diffuse surfaces](#). *ACM SIGGRAPH computer graphics* 18, 3

- (1984), 213–222. 2.3.3
- Xiangming Gu, Chao Du, Tianyu Pang, Chongxuan Li, Min Lin, and Ye Wang. 2025. On Memorization in Diffusion Models. arXiv:2310.02664 [cs.LG] <https://arxiv.org/abs/2310.02664> 4.4
- Xianfeng Gu and Shing-Tung Yau. 2003. Global conformal surface parameterization. In *Eurographics Symposium on Geometry Processing*. 127–137. 3.1
- Karthik S. Gurumoorthy and Anand Rangarajan. 2009. A Schrödinger Equation for the Fast Computation of Approximate Euclidean Distance Functions. In *Scale Space and Variational Methods in Computer Vision*, Xue-Cheng Tai, Knut Mørken, Marius Lysaker, and Knut-Andreas Lie (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 100–111. 4.2, 4.3
- Eric Haines. 1994a. Point in Polygon Strategies. Academic Press Prof., Inc., USA, 24–46. 2.3.3
- Eric Haines. 1994b. Point in Polygon Strategies. *Graphics Gems* 4, 1 (1994), 24–46. 2.4, 5.3
- Eberhard Hopf. 1950. The partial differential equation $u_t + uu_x = mu_{xx}$. *Communications on Pure and Applied mathematics* 3, 3 (1950), 201–230. 4.1
- Yixin Hu, Qingnan Zhou, Xifeng Gao, Alec Jacobson, Denis Zorin, and Daniele Panozzo. 2018. Tetrahedral Meshing in the Wild. *ACM Trans. Graph.* 37, 4, Article 60 (July 2018), 14 pages. <https://doi.org/10.1145/3197517.3201353> 2.3.3
- Saar Huberman, Amit Bracha, and Ron Kimmel. 2023. Deep Accurate Solver for the Geodesic Problem. In *International Conference on Scale Space and Variational Methods in Computer Vision*. Springer, 288–300. 2.4
- Stephen P Humphries and Dennis Johnson. 1989. A generalization of winding number functions on surfaces. *Proc. of the London Math. Soc.* 3, 2 (1989), 366–386. 2.3.3
- Alec Jacobson, Ladislav Kavan, and Olga Sorkine. 2013. Robust Inside-Outside Segmentation using Generalized Winding Numbers. *ACM Trans. Graph.* 32, 4 (2013). 2.3.3, 3.1, 4.2
- Alec Jacobson, Daniele Panozzo, et al. 2018. libigl: A simple C++ geometry processing library. <https://libigl.github.io/>. 5.3
- Anubhav Jain, Yuya Kobayashi, Takashi Shibuya, Yuhta Takida, Nasir Memon, Julian Togelius, and Yuki Mitsufuji. 2024. Classifier-Free Guidance inside the Attraction Basin May Cause Memorization. arXiv:2411.16738 [cs.CV] <https://arxiv.org/abs/2411.16738> 4.4
- Christina Karam, Kenjiro Sugimoto, and Keigo Hirakawa. 2019. Fast Convolutional Distance Transform. *IEEE Signal Processing Letters* 26, 6 (2019), 853–857. <https://doi.org/10.1109/LSP.2019.2910466> 4.2, 4.3
- Michael Kazhdan, Matthew Bolitho, and Hugues Hoppe. 2006. Poisson Surface Reconstruction. In *Symposium on Geometry Processing (SGP '06)*. Eurographics Association, 61–70. 2.3.3, 4.1
- Ron Kimmel and James A Sethian. 1998. Computing geodesic paths on manifolds. *Proceedings of the national academy of Sciences* 95, 15 (1998), 8431–8435. 2.4
- Ravikrishna Kolluri. 2008. Provably good moving least squares. *ACM Trans. Algorithms* 4, 2,

- Article 18 (May 2008), 25 pages. <https://doi.org/10.1145/1361192.1361195> 4.2, 4.2, 4.4
- G. Kreisselmeier and R. Steinhauser. 1980. Systematic Control Design By Optimizing A Vector Performance Index. In *Computer Aided Design of Control Systems*, M. A. Cuenod (Ed.). Pergamon, 113–117. <https://doi.org/10.1016/B978-0-08-024488-4.50022-X> 4.2
- Pavel A Krutitskii. 2001. The jump problem for the Laplace equation. *Applied Mathematics Letters* 14, 3 (2001), 353–358. 3.1
- Joseph Louis Lagrange. 1798. *Solutions de quelques problèmes relatifs aux triangles sphériques, avec une analyse complète de ces triangles*. 2.3.3
- John M. Lee. 2012. *Introduction to Smooth Manifolds* (2 ed.). Springer New York, NY. 2.2
- David Levin. 1998. The Approximation Power of Moving Least-Squares. *Math. Comp.* 67, 224 (1998), 1517–1531. <http://www.jstor.org/stable/2584860> 4.4
- Moshe Lichtenstein, Gautam Pai, and Ron Kimmel. 2019. Deep eikonal solvers. In *Scale Space and Variational Methods in Computer Vision: 7th International Conference, SSVM 2019, Hofgeismar, Germany, June 30–July 4, 2019, Proceedings* 7. Springer, 38–50. 2.4
- Yaron Lipman. 2021. Phase Transitions, Distance Functions, and Implicit Neural Representations. arXiv:2106.07689 [cs.LG] <https://arxiv.org/abs/2106.07689> 4.2, A, A.2
- Shibo Liu, Ligang Liu, and Xiao-Ming Fu. 2025. Closed-form Generalized Winding Numbers of Rational Parametric Curves for Robust Containment Queries. *ACM Trans. Graph.* 44, 4, Article 75 (July 2025), 9 pages. <https://doi.org/10.1145/3730886> 4.2
- Xingchao Liu, Chengyue Gong, and Qiang Liu. 2022. Flow Straight and Fast: Learning to Generate and Transfer Data with Rectified Flow. arXiv:2209.03003 [cs.LG] <https://arxiv.org/abs/2209.03003> 4.4
- Richard H MacNeal. 1949. *The solution of partial differential equations by means of electrical networks*. Ph. D. Dissertation. California Institute of Technology. 2.2
- Abhishek Madan and David IW Levin. 2022. Fast evaluation of smooth distance constraints on co-dimensional geometry. *ACM Transactions on Graphics (TOG)* 41, 4 (2022), 1–17. 4.2, 4.3
- James Clerk Maxwell. 1881. *A Treatise on Electricity and Magnetism*. Vol. II. Oxford University Press. 2.3.3
- Margaret McIntyre and Grant Cairns. 1993. A new formula for winding number. *Geometriae Dedicata* 46, 2 (1993), 149–159. 2.3.3
- Joseph S. B. Mitchell, D. Mount, and C. Papadimitriou. 1987. The Discrete Geodesic Problem. *SIAM J. Comput.* 16, 4 (1987), 647–668. 5.3
- Patrick Mullen, Fernando De Goes, Mathieu Desbrun, David Cohen-Steiner, and Pierre Alliez. 2010. Signing the Unsigned: Robust Surface Reconstruction from Raw Pointsets. *Computer Graphics Forum* 29, 5 (2010), 1733–1741. <https://doi.org/10.1111/j.1467-8659.2010.01782.x> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2010.01782.x> 2.4

- Lea Müller, Ahmed A. A. Osman, Siyu Tang, Chun-Hao P. Huang, and Michael J. Black. 2021. [On Self-Contact and Human Pose](#). In *Proceedings IEEE/CVF Conf. on Computer Vision and Pattern Recognition (CVPR)*. 2.3.3
- James R Munkres. 1984. *Elements of Algebraic Topology* (1st. ed.). CRC press. 2.3.2, 2.3.2
- Ken Museth, David E Breen, Ross T Whitaker, and Alan H Barr. 2002. Level set surface editing operators. In *Proceedings of the 29th annual conference on Computer graphics and interactive techniques*. 330–338. 2.4
- Ashish Myles, Nico Pietroni, and Denis Zorin. 2014. [Robust Field-Aligned Global Parametrization](#). *ACM Trans. Graph.* 33, 4 (2014). 3.3
- De-zhi Ning, Bin Teng, Hai-tao Zhao, and Chun-ling Hao. 2010. [A comparison of two methods for calculating solid angle coefficients in a BIEM numerical wave tank](#). *Engineering Analysis with Boundary Elements* 34, 1 (2010), 92–96. 2.3.3
- Yutaka Ohtake, Alexander Belyaev, Marc Alexa, Greg Turk, and Hans-Peter Seidel. 2005. Multi-level partition of unity implicits. In *ACM SIGGRAPH 2005 Courses* (Los Angeles, California) (*SIGGRAPH '05*). Association for Computing Machinery, New York, NY, USA, 173–es. <https://doi.org/10.1145/1198555.1198649> 4.3
- Y. Ohtake, A. Belyaev, and H.P. Seidel. 2003. A multi-scale approach to 3D scattered data interpolation with compactly supported basis functions. In *2003 Shape Modeling International*. 153–161. <https://doi.org/10.1109/SMI.2003.1199611> 4.4
- Helen Oleynikova, Alexander Millane, Zachary Taylor, Enric Galceran, Juan Nieto, and Roland Siegwart. 2016. Signed distance fields: A natural representation for both mapping and planning. In *RSS 2016 workshop: geometry and beyond-representations, physics, and scene understanding for robotics*. University of Michigan. 2.4
- Stanley Osher, Ronald Fedkiw, and K Piechor. 2004. Level set methods and dynamic implicit surfaces. *Appl. Mech. Rev.* 57, 3 (2004), B15–B15. 2.4
- A. C. Öztireli, G. Guennebaud, and M. Gross. 2009. Feature Preserving Point Set Surfaces based on Non-Linear Kernel Regression. *Computer Graphics Forum* 28, 2 (2009), 493–501. <https://doi.org/10.1111/j.1467-8659.2009.01388.x> arXiv:<https://onlinelibrary.wiley.com/doi/pdf/10.1111/j.1467-8659.2009.01388.x> 4.2, 4.4
- Jakiw Pidstrigach. 2022. Score-Based Generative Models Detect Manifolds. arXiv:2206.01018 [stat.ML] <https://arxiv.org/abs/2206.01018> 4.4
- Konstantin Poelke and Konrad Polthier. 2016. [Boundary-aware Hodge decompositions for piecewise constant vector fields](#). *Computer-Aided Design* 78 (2016), 126–136. 2.3.2
- Inigo Quilez. 2008. Raymarching Signed Distance Fields. <https://iquilezles.org/articles/raymarchingdf/>. 2.4
- Bruce L Reinhart. 1960. The winding number on two manifolds. In *Annales de l'institut Fourier*, Vol. 10. 271–283. 2.3.3
- Bruce L Reinhart. 1963. Further remarks on the winding number. In *Annales de l'institut Fourier*,

- Vol. 13. 155–160. 2.3.3
- Marzia Riso, Giacomo Nazzaro, Enrico Puppo, Alec Jacobson, Qingnan Zhou, and Fabio Pellacini. 2022. BoolSurf: Boolean Operations on Surfaces. *ACM Trans. Graph.* 41, 6 (2022), 1–13. 2.3.2, 2.3.3, 3.3
- Rohan Sawhney and Keenan Crane. 2017. Boundary First Flattening. *ACM Trans. Graph.* 37, 1, Article 5 (dec 2017), 14 pages. <https://doi.org/10.1145/3132705> 3.1
- Rohan Sawhney, Ruihao Ye, Johann Korndorfer, and Keenan Crane. 2020. FCPW: Fastest Closest Points in the West. 2.4, 5.3
- Christopher Scarvelis, Haitz Sáez de Ocáriz Borde, and Justin Solomon. 2023. Closed-Form Diffusion Models. arXiv:2310.12395 [cs.LG] <https://arxiv.org/abs/2310.12395> 4.4
- Günter Schwarz. 2006. *Hodge Decomposition-A method for solving boundary value problems*. Springer. 2.2
- Manu Sethi, Anand Rangarajan, and Karthik Gurumoorthy. 2012. The Schrödinger distance transform (SDT) for point-sets and curves. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 198–205. 4.2, 4.3
- Nicholas Sharp and Keenan Crane. 2020a. A Laplacian for Nonmanifold Triangle Meshes. *Computer Graphics Forum (SGP)* 39, 5 (2020). 5.2
- Nicholas Sharp and Keenan Crane. 2020b. You Can Find Geodesic Paths in Triangle Meshes by Just Flipping Edges. *ACM Trans. Graph.* 39, 6 (2020). 3.3
- Nicholas Sharp, Keenan Crane, et al. 2019a. GeometryCentral: A modern C++ library of data structures and algorithms for geometry processing. <https://geometry-central.net/>. (2019). 2.2
- Nicholas Sharp, Mark Gillespie, and Keenan Crane. 2021. Geometry Processing with Intrinsic Triangulations. (2021). 3.3
- Nicholas Sharp, Yousuf Soliman, and Keenan Crane. 2019b. Navigating Intrinsic Triangulations. *ACM Trans. Graph.* 38, 4 (2019). 3.3
- Nicholas Sharp, Yousuf Soliman, and Keenan Crane. 2019c. The Vector Heat Method. *ACM Trans. Graph.* 38, 3 (2019). 4.2, 5.2, 5.2, 5.2
- M. Shimrat. 1962. Algorithm 112: Position of Point Relative to Polygon. *Commun. ACM* 5, 8 (aug 1962), 434. <https://doi.org/10.1145/368637.368653> 2.3.3
- Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2022. Diffusion Art or Digital Forgery? Investigating Data Replication in Diffusion Models. arXiv:2212.03860 [cs.LG] <https://arxiv.org/abs/2212.03860> 4.4
- Ryan J. Tibshirani, Samy Wu Fung, Howard Heaton, and Stanley Osher. 2024. Laplace Meets Moreau: Smooth Approximation to Infimal Convolutions Using Laplace’s Method. arXiv:2406.02003 [math.OC] <https://arxiv.org/abs/2406.02003> 4.2, 4.2
- Y. Tong, P. Alliez, D. Cohen-Steiner, and M. Desbrun. 2006. Designing Quadrangulations with

- Discrete Harmonic Forms.** In *Symposium on Geometry Processing (SGP '06)*. Eurographics Association, 201–210. 3.1
- Philip Trettner, David Bommers, and Leif Kobbelt. 2021. Geodesic distance computation via virtual source propagation. In *Computer graphics forum*, Vol. 40. Wiley Online Library, 247–260. 2.4, 5.3
- Sathamangalam R Srinivasa Varadhan. 1967. On the behavior of the fundamental solution of the heat equation with variable coefficients. *Communications on Pure and Applied Mathematics* 20, 2 (1967), 431–455. 4.1, 4.2, A.1, A.2
- Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), Vol. 30. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf 4.4
- Eric Veach and Leonidas J Guibas. 1995. **Optimally combining sampling techniques for Monte Carlo rendering.** In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*. 419–428. 2.3.3
- Delio Vicini, Sébastien Speierer, and Wenzel Jakob. 2022. Differentiable Signed Distance Function Rendering. *Transactions on Graphics (Proceedings of SIGGRAPH)* 41, 4 (July 2022), 125:1–125:18. <https://doi.org/10.1145/3528223.3530139> 2.4
- Hao Wang, Carlos E. Scheidegger, and Claudio T. Silva. 2008. Optimal bandwidth selection for MLS surfaces. In *2008 IEEE International Conference on Shape Modeling and Applications*. 111–120. <https://doi.org/10.1109/SMI.2008.4547957> 4.4
- Zimo Wang, Cheng Wang, Taiki Yoshino, Sirui Tao, Ziyang Fu, and Tzu-Mao Li. 2025. HotSpot: Signed Distance Function Optimization with an Asymptotically Sufficient Condition. In *CVPR*. 4.2
- Hongyi Xu and Jernej Barbič. 2014. Signed Distance Fields for Polygon Soup Meshes. In *Proceedings of Graphics Interface 2014 (Montreal, Quebec, Canada) (GI '14)*. Canadian Information Processing Society, CAN, 35–41. 2.4
- Kaizhi Yang, Liu Dai, Isabella Liu, Xiaoshuai Zhang, Xiaoyan Sun, Xuejin Chen, Zexiang Xu, and Hao Su. 2025. IMLS-Splatting: Efficient Mesh Reconstruction from Multi-view Images via Point Representation. *ACM Trans. Graph.* 44, 4, Article 83 (July 2025), 11 pages. <https://doi.org/10.1145/3731210> 4.2
- Lior Yariv, Omri Puny, Natalia Neverova, Oran Gafni, and Yaron Lipman. 2023. Mosaic-SDF for 3D Generative Models. *arXiv preprint arXiv:2312.09222* (2023). 2.4
- TaeHo Yoon, Joo Young Choi, Sehyun Kwon, and Ernest K. Ryu. 2023. Diffusion Probabilistic Models Generalize when They Fail to Memorize. In *ICML 2023 Workshop on Structured Probabilistic Inference & Generative Modeling*. <https://openreview.net/forum?id=shciCbSk9h> 4.4

- Jihun Yu and Greg Turk. 2013. Reconstructing surfaces of particle-based fluids using anisotropic kernels. *ACM Trans. Graph.* 32, 1, Article 5 (Feb. 2013), 12 pages. <https://doi.org/10.1145/2421636.2421641> 4.3
- Lyubomir G. Zagorchev and Arthur Ardeshir Goshtasby. 2012. A Curvature-Adaptive Implicit Surface Reconstruction for Irregularly Spaced Points. *IEEE Transactions on Visualization and Computer Graphics* 18, 9 (2012), 1460–1473. <https://doi.org/10.1109/TVCG.2011.2764.4>
- Qingnan Zhou, Eitan Grinspun, Denis Zorin, and Alec Jacobson. 2016. *Mesh Arrangements for Solid Geometry*. *ACM Trans. Graph.* 35, 4, Article 39 (July 2016), 15 pages. <https://doi.org/10.1145/2897824.2925901> 2.3.3, 3.3

APPENDIX A

The signed Hopf-Cole transformation

Here we give a derivation of the signed Hopf-Cole transformation. The derivation in Section A.1 is inspired by the formulation of the viscous signed eikonal equation in Theorems 4 & 5 of [Lipman \[2021\]](#). While [Lipman \[2021\]](#) shows its equivalence to a screened Poisson equation using a signed Hopf-Cole transformation, its boundary conditions and the validity of the transformation when Ω is open are unclear. Section A.2 uses perturbative methods to provide another perspective of the Hopf-Cole transformation.

A.1 From the (signed) eikonal equation to a (jump) screened Laplace equation

We start with the signed viscous eikonal equation

$$\begin{aligned} \sqrt{t}\Delta u(x) - \text{sign}_\Omega(x) (\|\nabla u(x)\|^2 - 1) &= 0 & x \notin \Omega, \\ u(x) &= 0 & x \in \Omega, \\ \frac{\partial u}{\partial n}(x) &= 1 & x \in \Omega \end{aligned} \tag{A.1}$$

where $\text{sign}_\Omega(x) : \mathbb{R}^d \rightarrow \{-1, 1\}$ is a sign function associated with the $(d - 1)$ -dimensional oriented surface Ω , defined to be a piecewise constant function whose boundary conditions satisfy

$$\text{sign}_\Omega \left(\lim_{s \rightarrow 0} x \pm sn(x) \right) = \lim_{s \rightarrow 0} \text{sign}_\Omega (x \pm sn(x)) = \pm 1 \quad x \in \Omega \tag{A.2}$$

where n denotes the outward-facing normal direction to Ω .

We consider the following function of $u(x)$, which we refer to as the *signed Hopf-Cole transformation*,

$$w(x) := \text{sign}_\Omega(x) \exp \left(- \text{sign}_\Omega(x) u(x) / \sqrt{t} \right). \tag{A.3}$$

Equation A.3 may also be considered a signed extension of Varadhan's formula [[Varadhan 1967](#)].

We observe that for all points $x \in \mathbb{R}^d$ not on the locus of points where sign_Ω changes sign, we have that $0 < |w(x)| \leq 1$ for all $t > 0$. The inverse transformation corresponding to Equation A.3 is

$$u(x) = -\text{sign}_\Omega(x) \sqrt{t} \log(\text{sign}_\Omega(x) w(x)). \quad (\text{A.4})$$

If Ω is simple and closed, then Ω bounds a simply-connected region A , and the only reasonable definition of $\text{sign}_\Omega(x)$ on $\{x \mid x \notin \Omega\}$ is such that

$$\text{sign}_\Omega(x) = \begin{cases} -1 & x \in A \\ +1 & x \in \mathbb{R}^d \setminus \bar{A} \end{cases}.$$

Since Ω separates the domain $\mathbb{R}^d$, the solution of the viscous signed eikonal equation in Equation A.1 is equivalent to the union of the solutions of two independent viscous signed eikonal equations: one defined on A with boundary Ω (the “interior”), and one defined on $\mathbb{R}^d \setminus \bar{A}$ with boundary $-\Omega$ (the “exterior”), where $-\Omega$ denotes Ω with opposite orientation.

Let $u^+ : \mathbb{R}^d \setminus \bar{A} \rightarrow \mathbb{R}$ denote the solution to the exterior problem, and $u^- : A \rightarrow \mathbb{R}$ the solution to the interior problem (and similarly for $w^\pm(x)$). Using the signed Hopf-Cole transformation, we obtain the following expressions:

$$\begin{aligned} \nabla u^\pm(x) &= \mp \sqrt{t} \frac{\nabla w}{w} \\ \Delta u^\pm(x) &= \mp \sqrt{t} \left(\frac{\Delta w}{w} - \frac{\|\nabla w\|^2}{w^2} \right). \end{aligned}$$

Applying these to the viscous signed eikonal equation in Equation A.1, we obtain a screened Laplace equation in $w(x)$ for both the interior and exterior regions:

$$\Delta w^\pm(x) - \frac{1}{t} w^\pm(x) = 0. \quad (\text{A.5})$$

The Dirichlet boundary conditions of $w^+(x)$ and $w^-(x)$ are

$$\begin{aligned} w^\pm(x) &= \text{sign}_\Omega(x) \exp\left(-\text{sign}_\Omega(x) u^\pm(x) / \sqrt{t}\right) & x \in \Omega \\ \Rightarrow w^\pm(x) &= \text{sign}_\Omega(x) & \text{since } u^\pm(x) = 0 \text{ for } x \in \Omega. \end{aligned}$$

The Neumann boundary conditions for both $w^+(x)$ and $w^-(x)$ are identical w.r.t. the normals of Ω :

$$\frac{\partial w^\pm(x)}{\partial n(x)} = -\frac{1}{\sqrt{t}}.$$

Putting together the solutions $w^+(x)$ and $w^-(x)$, we obtain the following jump screened Laplace equation for $w(x)$:

$$\begin{aligned} \Delta w(x) - \frac{1}{t} w(x) &= 0 & x \notin \Omega, \\ w^\pm(x) &= \text{sign}_\Omega^\pm(x) & x \in \Omega, \\ \frac{\partial w}{\partial n}(x) &= -\frac{1}{\sqrt{t}} & x \in \Omega. \end{aligned} \quad (\text{A.6})$$

where $w^\pm(x) := w(x^\pm)$ and $\text{sign}_\Omega^\pm(x) := \text{sign}_\Omega(x^\pm)$ (where $x^\pm := \lim_{s \rightarrow 0} x \pm sn(x)$).

When Ω is open, the sign function $\text{sign}_\Omega(x)$ is no longer easily defined away from Ω . However, we always know the Dirichlet boundary conditions of sign_Ω (Equation A.2), which from Equation A.5 we know will always equal those of $w(x)$. Though we have not yet introduced any constraints between $\text{sign}_\Omega(x)$ and $\text{sign}_w(x)$ at points $x \notin \Omega$, we will assume that $\text{sign}_\Omega = \text{sign}_w$ — meaning $\text{sign}_\Omega(x)$ is no longer independent of $w(x)$ and $u(x)$, but coupled to them — and re-write the signed Hopf-Cole transformation in Equation A.3 as

$$w(x) = \text{sign}_w(x) \exp\left(-\text{sign}_w(x)u(x)/\sqrt{t}\right) \quad (\text{A.7})$$

with inverse

$$u(x) = -\text{sign}_w(x)\sqrt{t} \log(\text{sign}_w(x)w(x)) = -\text{sign}_w(x)\sqrt{t} \log|w(x)|.$$

(Note that defining the signed Hopf-Cole transformation basically forces us to assume $\text{sign}_\Omega = \text{sign}_w$, otherwise the inverse transformation may not be well-defined.) Using the updated signed Hopf-Cole transformation, we obtain the following expressions:

$$\begin{aligned} \nabla u &= -\sqrt{t} (\log|w| \nabla \text{sign}_w + \text{sign}_w \nabla \log|w|) \\ \Delta u &= -\sqrt{t} (2\nabla \log|w| \cdot \text{sign}_w + \log|w| \Delta \text{sign}_w + \text{sign}_w \Delta \log|w|). \end{aligned}$$

We see that these derivatives are only well-defined for points x which satisfy the following two conditions — though later we will see that they are equivalent:

1. Though derivatives involving $|w|$ are only well-defined where $w \neq 0$, above we established by Equation A.3 that $|w| > 0$ always. Hence for any x away from Ω (more precisely, x which admit open neighborhoods with empty intersection with Ω), we may safely take the gradient of $|w(x)|$.
2. Second, we have that $\nabla \text{sign}_w(x) = 0$ and $\Delta \text{sign}_w(x)$ for $x \notin \partial w$, where ∂w denotes the locus of points where w changes sign.

For all x satisfying these two conditions,

$$\begin{aligned}
 \nabla u(x) &= -\sqrt{t} \operatorname{sign}_w(x) \nabla \log |w(x)| \\
 &= -\sqrt{t} \underbrace{\operatorname{sign}_w(x) w(x)}_{|w(x)|} \frac{\nabla w(x)}{|w(x)|^2} \\
 &= -\sqrt{t} \nabla w(x) / |w(x)| \\
 \Delta u(x) &= -\sqrt{t} \frac{|w(x)| \Delta w(x) - \nabla w(x) \cdot \frac{w(x) \nabla w(x)}{|w(x)|}}{|w(x)|^2} \\
 &= -\sqrt{t} \left(\frac{\Delta w(x)}{|w(x)|} - \frac{w(x) \|\nabla w(x)\|^2}{|w(x)|^3} \right) \\
 &\Rightarrow \sqrt{t} \Delta u(x) - \operatorname{sign}_w(x) (\|\nabla u(x)\|^2 - 1) = 0 \\
 &\Rightarrow -t \left(\frac{\Delta w(x)}{|w(x)|} - \frac{w(x) \|\nabla w(x)\|^2}{|w(x)|^3} \right) - \operatorname{sign}_w(x) \left(t \frac{\|\nabla w(x)\|^2}{|w(x)|^2} - 1 \right) = 0 \\
 &\Rightarrow -t \operatorname{sign}_w(x) \left(\frac{\Delta w(x)}{|w(x)|} - \frac{w(x) \|\nabla w(x)\|^2}{|w(x)|^3} \right) - \left(t \frac{\|\nabla w(x)\|^2}{|w(x)|^2} - 1 \right) = 0 \\
 &\Rightarrow -t \operatorname{sign}_w(x) \frac{\Delta w(x)}{|w(x)|} + t \underbrace{\operatorname{sign}_w(x) w(x)}_{|w(x)|} \frac{\|\nabla w(x)\|^2}{|w(x)|^3} - t \frac{\|\nabla w(x)\|^2}{|w(x)|^2} + 1 = 0 \\
 &\Rightarrow -t \operatorname{sign}_w(x) \frac{\Delta w(x)}{|w(x)|} + 1 = 0 \\
 &\Rightarrow -t \frac{\Delta w(x)}{w(x)} + 1 = 0 \\
 &\Rightarrow \Delta w(x) - \frac{1}{t} w(x) = 0.
 \end{aligned}$$

We arrive at the same screened Laplace equation for $w(x)$ as before, from which we conclude that the two conditions on x are equivalent, because the locus of points where w changes sign exactly corresponds to Ω : the sign cannot change anywhere else due to the properties of the screened Laplace operator. In summary, this screened Laplace equation is valid for all $x \notin \Omega$.

We already established the Dirichlet boundary conditions of $w(x)$, so all that remains is to establish the Neumann boundary conditions:

$$\begin{aligned}
 1 &= \nabla u(x) \cdot n = -\frac{\sqrt{t}}{|w(x)|} (\nabla w(x) \cdot n), \quad x \in \Omega \\
 \Rightarrow \frac{\partial w}{\partial n}(x) &= -\frac{1}{\sqrt{t}} \quad |w(x)| = 1
 \end{aligned}$$

which are the same Neumann boundary conditions we obtained in Equation A.6 when we assumed Ω was closed. In summary, we established the equivalence of the signed viscous

eikonal equation (Equation A.1) with the jump screened Laplace equation in Equation A.6, for both closed and open Ω .

Remarks:

- If we did not assume $\text{sign}_\Omega = \text{sign}_w$, then we would not have been able to replace $\text{sign}_w(x)w(x)$ with $|w(x)|$ in the inverse signed Hopf-Cole transformation. It is this assumption that (1) guarantees that the signed Hopf-Cole transformation can be inverted, since it guarantees that the argument of $\log(\text{sign}_\Omega(x)w(x)) \rightarrow \log |w(x)|$ is positive; and (2) allows us to avoid the possibility of Dirac deltas when differentiating u or w .
- We have that $\text{sign}_w = \text{sign}_u$, which can be seen either by inspecting the signed Hopf-Cole transformation (Equation A.3), or by following the reasoning in the above bulletpoint and observing that sign_Ω determines the sign of the solution $u(x)$ to the viscous signed eikonal equation in Equation A.1. Hence we can also express the updated signed Hopf-Cole transformation (Equation A.7) as

$$w(x) = \text{sign}_u(x) \exp(-|u(x)|/\sqrt{t}).$$

We also see that since $\text{sign}_w = \text{sign}_u$ is determined by a screened Poisson equation, that sign_u is different than, for example, the sign given by winding numbers (for $t < \infty$).

A.2 Analysis of the screened Laplace equation as a boundary-layer problem

This time we start from a screened Laplace equation with spatially-varying screening coefficient,

$$\varepsilon^2 \Delta w(x) - \alpha^2(x)w(x) = 0.$$

Equations of this form have been studied as time-independent Schrödinger equations, and we paraphrase here the derivation given in an example in [Bender and Orszag \[1999, §10\]](#).

Dissipative and dispersive phenomena are both characterized by exponential behavior (the exponent is real for dissipation/diffusion, and imaginary for dispersion.) Thus for differential equations exhibiting dissipative and/or dispersive phenomena, we can try seeking an approximate solution of the form

$$w(x) \sim A(x) \exp(S(x)/\delta), \quad \delta \rightarrow 0^+$$

which is called the *WKB approximation*. (When the phase $S(x)$ is real, there is a boundary layer of thickness δ ; when $S(x)$ is imaginary, there is a region of oscillation characterized by waves of wavelength of order δ .) Making more explicit the dependencies of the amplitude and phase on δ , and using series expansions of the amplitude $A(x)$ and phase $S(x)$ in powers of δ , yields the more useful power series

$$w(x) \sim \exp\left(\frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n S_n(x)\right), \quad \delta \rightarrow 0^+.$$

Substituting this WKB approximation, whose derivatives are

$$\begin{aligned}\nabla w(x) &\sim \left(\frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n \nabla S_n(x) \right) \exp \left(\frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n S_n(x) \right), & \delta \rightarrow 0 \\ \Delta w(x) &\sim \left[\left(\frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n \nabla S_n(x) \right)^2 + \frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n \Delta S_n(x) \right] \exp \left(\frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n S_n(x) \right), & \delta \rightarrow 0\end{aligned}$$

we obtain

$$\begin{aligned}&\varepsilon^2 \left[\left(\frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n \nabla S_n(x) \right)^2 + \frac{1}{\delta} \sum_{n=0}^{\infty} \delta^n \Delta S_n(x) \right] - \alpha^2(x) = 0 \\ \Rightarrow &\frac{\varepsilon^2}{\delta^2} \left(\sum_{n=0}^{\infty} \delta^n \nabla S_n(x) \right)^2 + \frac{\varepsilon^2}{\delta} \sum_{n=0}^{\infty} \delta^n \Delta S_n(x) = \alpha^2(x).\end{aligned}$$

The largest term on the l.h.s. is the first term $\frac{\varepsilon^2}{\delta^2} \|\nabla S_0(x)\|^2$ of the leftmost sum (all further terms are multiplied by more powers of the small scale parameter δ .) Assuming $\alpha(x) \neq 0$, by dominant balance this term must have the same order of magnitude as $\alpha^2(x)$ on the right side; hence we determine that $\delta(\varepsilon)$ is proportional to ε . Taking $\delta = \varepsilon$ yields

$$\left(\sum_{n=0}^{\infty} \varepsilon^n \nabla S_n(x) \right)^2 + \sum_{n=0}^{\infty} \varepsilon^{n+1} \Delta S_n(x) = \alpha^2(x)$$

which yields a recursive formula for the S_n :

$$\begin{cases} \|\nabla S_0(x)\|^2 = \alpha^2(x) \\ \Delta S_{n-1}(x) + \sum_{j=0}^n \nabla S_j(x) \cdot \nabla S_{n-j}(x) = 0, \quad n \geq 1 \end{cases}$$

which is obtained by equating powers of ε . (It is possible that $\delta \sim c\varepsilon$ for some constant of proportionality c , but in general in asymptotic analysis one is only interested in the order of magnitude of the boundary-layer thickness.)

We observe that the first equation is an eikonal equation with spatially-varying wave velocity (whose boundary conditions are determined by the boundary conditions of the original screened Poisson equation). In fact, our analysis tells us that the leading term of $w(x)$ in the small- ε regime is the solution to an eikonal equation — equivalent to what is implied by Hopf-Cole transformation and Varadhan's formula, which reflect the special case of $\alpha = \pm 1$.

The first few equations in the sequence are

$$\begin{aligned}\|\nabla S_0(x)\|^2 &= \alpha^2(x) \\ 2\nabla S_1(x) \cdot \nabla S_0(x) + \Delta S_0(x) &= 0 \\ \|\nabla S_1(x)\|^2 + 2\nabla S_2(x) \cdot \nabla S_0(x) + \Delta S_1(x) &= 0 \\ &\dots\end{aligned}$$

If $\alpha = 1$ is constant, then in 1D, at least, all terms beyond the leading one are zero (that is, $S_n = 0$ for $n \geq 1$), meaning the phenomena is accurately described by the leading eikonal term. Spatially-varying $\alpha(x)$ corresponds to distance induced by a metric derived from the coefficients [Varadhan 1967].

Making these types of approximations is common in optics; for example, using only the leading term corresponds to making a geometric or ray optics assumption, from which the eikonal equation is derived in what is called the *high-frequency limit*. For more theory connecting viscosity solutions, Hamilton-Jacobi equations, and more general reaction-diffusion equations, see [Freidlin 1986; Fleming and Souganidis 1986; Fedotov 1999].

Phase fields. The boundary-layer perspective also yields a connection between viscous eikonal equations and phase fields, also used by Lipman [2021]. In particular, consider the optimization

$$\begin{aligned} \min_{v: M \rightarrow \mathbb{R}} \quad & \int_M \frac{1}{\lambda^2} \|\nabla v(x)\|^2 + \frac{1}{2} (|v(x)| - 1)^2 \\ \text{s.t.} \quad & v(x) = 0 \quad x \in \partial M \end{aligned}$$

whose objective is a variant of the *Modica-Mortola functional* often used in image processing. Expanding the first term of the objective (and using angle brackets to denote the L^2 inner product) yields

$$\begin{aligned} \int_M \frac{1}{\lambda^2} \|\nabla v(x)\|^2 &\rightarrow \frac{1}{\lambda^2} \langle \Delta v, v \rangle_M \\ &= \frac{1}{\lambda^2} (-\langle \Delta v, v \rangle_M + \langle n \cdot \nabla v, v \rangle_{\partial M}) \quad (\text{Stokes' theorem}) \end{aligned}$$

where n denotes the unit normal to ∂M . To derive the necessary conditions for optimality, we group the interior and boundary terms of the objective, and differentiate the expressions w.r.t. v and set to 0, to obtain

$$\begin{aligned} \frac{1}{\lambda^2} \Delta v(x) - v(x) &= -\text{sign}_v(x) & x \in M \\ v(x) &= 0 & x \in \partial M \\ \frac{\partial v}{\partial n}(x) &= \mu & x \in \partial M \end{aligned}$$

where $\mu \in \mathbb{R}$ is a Lagrange multiplier. If the domain M is a closed region where $\text{sign}_v(x) = 1$, then applying the change of variable $w(x) = 1 - v(x)$ yields the jump screened Laplace equation in Equation A.6 where $\text{sign}_w(x) = 1$. Where $\text{sign}_v(x) = -1$, applying the change of variable $w(x) = -1 - v(x)$ yields Equation A.6, with the same Neumann boundary conditions.

