The Closest Point Heat Method for Solving Eikonal Equations on Implicit Surfaces

Tony Wong* Shingyu Leung† Byungjoon Lee‡,¶

September 11, 2025

Abstract

We introduce the Closest Point Heat Method (CPHM), a novel approach for solving the surface Eikonal equation on general smooth surfaces. Building on the strengths of the classical heat method, such as simplicity of implementation and computational efficiency, CPHM integrates closest point techniques to reduce dependence on surface meshes. This embedding framework naturally extends the heat method to implicit surfaces while preserving both its efficiency and intrinsic geometric properties. Numerical experiments on benchmark geometries confirm the accuracy and convergence of the proposed method and demonstrate its effectiveness on complex shapes.

Keywords— Closest point method, Heat method, Surface Eikonal equations
AMS subject classifications— 58J05, 65M06, 65M20, 65N06, 65N40, 65D18

1 Introduction

The Eikonal equation is a fundamental nonlinear partial differential equation that arises in a variety of contexts involving wavefront propagation, most notably in geometric optics and computational geometry. On a smooth surface $\mathcal{S}$ embedded in $\mathbb{R}^3$, this equation takes the form of the surface Eikonal equation, which underpins the analysis of high-frequency surface wave propagation [1], and is given by:

$$\begin{aligned} \|\nabla_{\mathcal{S}}\phi(\boldsymbol{y})\| &= \frac{1}{F(\boldsymbol{y})}, \quad \boldsymbol{y} \in \mathcal{S} \setminus P, \\ \phi(\boldsymbol{y}_0) &= 0, \quad \boldsymbol{y}_0 \in P, \end{aligned} \tag{1.1}$$

where $\|\cdot\|$ is the Euclidean norm, P is the set of source points, $F(\boldsymbol{y})$ is the wave speed, and $\phi(\boldsymbol{y})$ denotes the shortest traveling time of the wave from the source set P to a point $\boldsymbol{y} \in \mathcal{S}$. The surface gradient $\nabla_{\mathcal{S}}\phi$ is defined by

$$\nabla_{\mathcal{S}}\phi(\boldsymbol{y}) = \nabla\phi - (\mathbf{N} \cdot \nabla\phi)\,\mathbf{N}$$

where $\mathbf{N}$ is the unit normal vector to the surface $\mathcal{S}$. Beyond wave propagation, the surface Eikonal equation is a fundamental tool in computer graphics, geometric processing, and machine learning

*Department of Mathematics, University of California, Los Angeles, USA
†Department of Mathematics, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong
‡Department of Mathematics, The Catholic University of Korea, Republic of Korea
¶Corresponding author. Email: blee@catholic.ac.kr

applications such as surface reparameterization [2], image segmentation on manifolds [3], and intrinsic distance-based feature extraction [4]. While the general Eikonal equation allows for spatially varying wave speed, the present work focuses on the special case when $F(\boldsymbol{y}) = 1$, corresponding to the unit-speed formulation $\|\nabla_{\mathcal{S}}\phi\| = 1$. This simplification is sufficient for computing geodesic distance functions and enables the development of efficient projection-based solvers.

Despite its wide applicability, computing numerical solutions to the surface Eikonal equation on general surfaces remains challenging due to the need to respect the underlying geometry and the nonlinearity of the PDE. Following the work of [5], approximation-based approaches such as variants of fast marching [6] and fast sweeping [7] methods have been widely adapted to surface domains. For instance, a triangulation-invariant method for anisotropic geodesic map computation improves robustness to mesh irregularities and anisotropy [8]. Fast sweeping techniques have also been extended to triangulated surfaces for geodesic computation [9], and even to implicit surfaces [10]. In addition, a number of methods based on parametric surface representations have been proposed to enhance computational efficiency: these include weighted distance map computation [11], efficient Eikonal solvers on parametric manifolds [12], and parallel algorithms for approximating distance maps [13]. An optimal control approach [14] has also been developed, employing the Hopf–Lax formula to recast the Eikonal equation [15, 16] as a variational problem solved via convex optimization techniques. Together, these methods form a diverse set of tools for surface Eikonal solvers, though many rely on surface discretization quality and parametrization, which can limit robustness in general geometric settings.

An alternative approach, known as the heat method, sidesteps these difficulties by exploiting the short-time behavior of the heat equation to approximate geodesic distances. First introduced by [17], the heat method solves a pair of linear problems, heat diffusion followed by the solution of a Poisson equation, making it efficient, robust to mesh quality, and naturally compatible with intrinsic surface geometry. However, the heat method typically relies on a mesh or parametrization of the surface, which can limit its applicability to more general implicit surfaces or point cloud data where such discretizations are unavailable or unreliable.

In this paper, we devote ourselves to extending the heat method through the use of closest point techniques, which allow for solving surface partial differential equations (PDEs) without requiring an explicit parametrization or triangulated mesh. The closest point method (CPM) embeds the surface problem into a higher-dimensional Cartesian space by extending functions off the surface via a closest point extension, where each point in a tubular neighborhood is mapped to its nearest point on the surface. This enables the use of standard Cartesian finite difference schemes to approximate differential operators, such as gradients and Laplacians, directly on the surface. By avoiding mesh generation and surface fitting, this approach naturally handles surfaces represented implicitly, such as level sets or point clouds, and improves robustness in complex geometric settings. Through this embedding framework, we generalize the heat method to implicit surfaces while preserving its efficiency and intrinsic geometric fidelity.

We acknowledge that similar ideas have been explored in prior work. In particular, the authors of [18] proposed a CPM-based variant of the heat method that incorporates internal boundary conditions (IBCs) to enforce Dirichlet constraints near the source point. Their formulation modifies the right-hand side of the heat equation using a smoothed Heaviside function, requiring careful treatment of boundary interpolation and stabilization. In contrast, our approach maintains the original two-step structure of the heat method, consisting of heat diffusion followed by Poisson recovery, while leveraging explicit closest point extensions for both scalar and vector fields. This avoids the complexity of IBC enforcement and results in a simpler, mesh-free framework that naturally accommodates implicit surface representations.

Our main contributions are summarized as follows:

- We introduce a fast and reliable method for solving the surface Eikonal equation, called the Closest Point Heat Method (CPHM), for computing geodesic distances.
- The proposed method eliminates the need for surface parameterizations or triangulated meshes

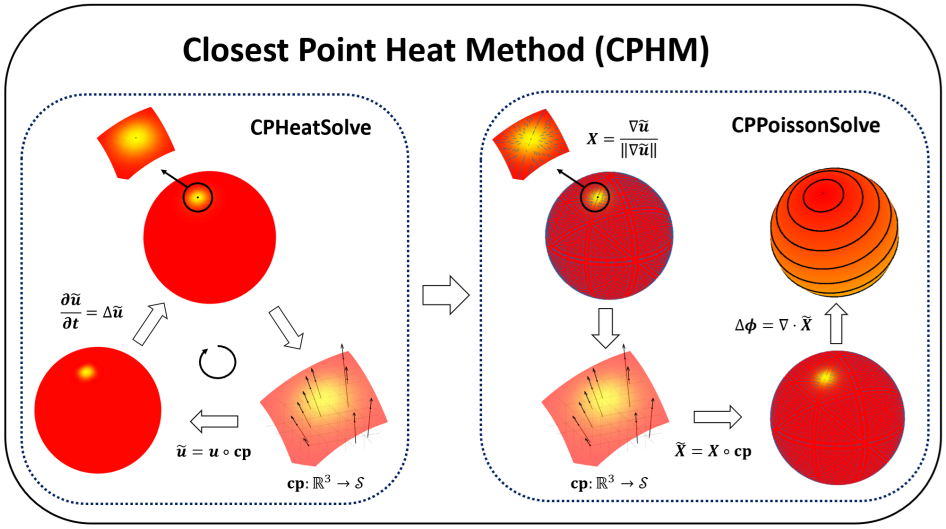

Figure 1: Overview of the proposed method. As in the original heat method [17], the approach consists of two steps: (CPHeatSolve, left) the heat equation is solved in the embedding space using the closest point extension $\tilde{u} = u \circ \mathrm{cp}$, allowing heat flow on the surface $\mathcal{S} \subset \mathbb{R}^3$ via $\partial \tilde{u}/\partial t = \Delta \tilde{u}$;(CPPoissonSolve, right) the normalized gradient $X = \nabla \tilde{u}/\|\nabla \tilde{u}\|$ is computed and extended, and the Poisson equation $\Delta \phi = \nabla \cdot \tilde{X}$ is then solved to recover approximate geodesic distances. All computations are carried out in $\mathbb{R}^3$ using the closest point map $\mathrm{cp} : \mathbb{R}^3 \to \mathcal{S}$.

by leveraging standard Cartesian finite difference schemes.

- By embedding the problem in the ambient space, the method extends the classical heat method to implicit surfaces while preserving intrinsic geometric properties.

The overall procedure of CPHM is illustrated in Figure 1.

This paper is organized as follows. Section 2 provides brief reviews of the CPM and the heat method. In Section 3, we present the proposed method for solving the surface Eikonal equation, the Closest Point Heat Method (CPHM). Section 4 presents various numerical results that validate the performance of the method, and the final section offers conclusions and discusses directions for future work.

2 Preliminaries

This section offers brief overviews of the two principal ingredients of this work: the CPM [19] and the heat method [17].

2.1 The Closest Point Method

The *closest point method* (CPM) is a numerical technique for solving partial differential equations (PDEs) defined on surfaces embedded in higher-dimensional spaces. The main idea of CPM is to extend the surface PDE to a neighborhood in the embedding space, enabling the use of standard Cartesian finite difference or finite element methods. We shall first introduce some key terminologies. Give a smooth surface $\mathcal{S} \subset \mathbb{R}^n$, there is a tubular neighborhood $B(\mathcal{S})$ such that each point $\boldsymbol{x} \in B(\mathcal{S})$ has a unique nearest point on $\mathcal{S}$ (in Euclidean distance). As such, we introduce the *closest point*

function:

$$\mathrm{cp}(\boldsymbol{x}) = \underset{\boldsymbol{y}\in\mathcal{S}}{\arg\min} \|\boldsymbol{x}-\boldsymbol{y}\|\,, \qquad \boldsymbol{x}\in B(\mathcal{S})\,, \tag{2.1}$$

Using this function, a surface function $u : S \to \mathbb{R}$ can be extended to $B(\mathcal{S})$ via the *closest point extension*,

$$Eu(\boldsymbol{x}) := u\left(\mathrm{cp}(\boldsymbol{x})\right)\,, \qquad \boldsymbol{x}\in B(\mathcal{S}) \tag{2.2}$$

We emphasize that Eu is a function that is defined in the tubular neighborhood $B(\mathcal{S})$. Such extension propagates the surface function constantly along the normal direction of $\mathcal{S}$, and therefore allows standard differential operators (e.g., gradient, Laplacian) to act on Eu in the embedding space while still capturing the surface behavior. We shall state the key principles [19] that allow us to extend a surface PDE into the embedding space:

Principle 1 (Gradient principle). Let $\boldsymbol{y}\in\mathcal{S}$. Then

$$\nabla_{\mathcal{S}} u\left(\boldsymbol{y}\right) = \nabla\left[Eu\left(\boldsymbol{y}\right)\right]\,. \tag{2.3}$$

Principle 2 (Divergence principle). Denote the surface divergence operator by $\nabla_{\mathcal{S}}\cdot$. Let $\mathcal{S}_\delta$ be the surface that is constantly displaced by a displacement of δ from $\mathcal{S}$. Suppose $\boldsymbol{v}$ is a vector field on $\mathbb{R}^3$ that is tangent at $\mathcal{S}$ and all $\mathcal{S}_\delta \subset B(\mathcal{S})$. Then for $\boldsymbol{y}\in\mathcal{S}$, we have

$$\nabla_{\mathcal{S}}\cdot\boldsymbol{v}(\boldsymbol{y}) = \nabla\cdot\boldsymbol{v}(\boldsymbol{y})\,. \tag{2.4}$$

By applying divergence principle on $\boldsymbol{v} = \nabla_{\mathcal{S}} u = \nabla u|_{\mathcal{S}}$, we derive the Laplacian principle,

$$\Delta_{\mathcal{S}} u(\boldsymbol{y}) = \Delta\left[Eu\right](\boldsymbol{y})\,, \qquad \boldsymbol{y}\in\mathcal{S}\,. \tag{2.5}$$

This allows us to approximate the surface Laplace–Beltrami operator $\Delta_S u$ by the standard Laplacian to Eu near the surface. This approach eliminates the need for surface parameterizations or triangulations and is particularly effective when combined with implicit surface representations [20]. For further details on CPM, see [19, 21].

2.2 The Heat Method

To compute geodesic distances on a surface, the heat method leverages the connection between heat diffusion and shortest paths, formalized by Varadhan's asymptotic formula [22]:

$$\phi(\boldsymbol{x},\boldsymbol{y}) = \lim_{t\to 0}\sqrt{-4t\log k_{t,\boldsymbol{x}}\left(\boldsymbol{y}\right)}, \tag{2.6}$$

where $\phi(\boldsymbol{x},\boldsymbol{y})$ denotes the geodesic distance between points $\boldsymbol{x}$ and $\boldsymbol{y}$, and $k_{t,\boldsymbol{x}}(\boldsymbol{y})$ is the heat kernel, which is the fundamental solution of the heat equation initiated at $\boldsymbol{x}$.

Rather than evaluating the limit directly, the heat method approximates the distance by simulating heat flow. It begins by solving the heat equation with an initial delta impulse at $\boldsymbol{x}$:

$$\partial_t u = \Delta_{\mathcal{S}} u, \quad u\left(\boldsymbol{x},0\right) = \delta_{\boldsymbol{x}_0}\left(\boldsymbol{x}\right),$$

where $\Delta_{\mathcal{S}}$ is the Laplace–Beltrami operator. The solution $u\left(\boldsymbol{y},t\right)$ approximates the heat kernel $k_{t,\boldsymbol{x}}\left(\boldsymbol{y}\right)$ for small t, providing a smooth function from which distance information can be inferred.

From the asymptotic form of the heat kernel in (2.6), we have

$$u(\boldsymbol{y},t) \approx k_{t,\boldsymbol{x}}\left(\boldsymbol{y}\right) \sim \frac{1}{(4\pi t)^{n/2}} e^{-\phi(\boldsymbol{x},\boldsymbol{y})^2/4t}\cdot a(\boldsymbol{x},\boldsymbol{y}),$$

where $a\left(\boldsymbol{x},\boldsymbol{y}\right)$ is a smooth, positive function. Neglecting the lower-order variation introduced by $a\left(\boldsymbol{x},\boldsymbol{y}\right)$, we take the logarithm and differentiate with respect to $\boldsymbol{y}$ to obtain

$$\nabla_{\mathcal{S}} u \approx u\left(\boldsymbol{y},t\right)\cdot\nabla_{\mathcal{S}}\log k_{t,\boldsymbol{x}}\left(\boldsymbol{y}\right) \sim -\frac{\phi\left(\boldsymbol{x},\boldsymbol{y}\right)}{2t}\nabla_{\mathcal{S}}\phi\cdot u\left(\boldsymbol{y},t\right),$$

which implies $\nabla_{\mathcal{S}} u \propto -\nabla_{\mathcal{S}} \phi$. In other words, the direction of steepest decrease in heat closely approximates the direction of shortest paths, that is, the geodesics on the surface.

Finally, to recover an approximation of the geodesic distance, the method solves a Poisson equation whose divergence matches that of the normalized vector field $\boldsymbol{X} = -\nabla_{\mathcal{S}} u / \|\nabla_{\mathcal{S}} u\|$:

$$\Delta_{\mathcal{S}} \phi = \nabla_{\mathcal{S}} \cdot \boldsymbol{X}.$$

For more information on the heat method, we refer the reader to [17].

3 Closest Point Heat Method

This section introduces a new method for solving the surface Eikonal equation, the Closest Point Heat Method (CPHM). It builds upon the classical heat method introduced by [17], adapting it to the CPM framework so that all computations are carried out in the ambient Euclidean space. This formulation removes the need for surface parameterizations or triangulated meshes and instead leverages standard Cartesian finite difference schemes. In this work, we restrict ourselves to surfaces embedded in $\mathbb{R}^3$, where the closest point extension and finite difference discretization can be eifficiently implemented on regular grids.

The method begins by solving the surface heat equation using CPM for a prescribed time T. As suggested in [17], we set $T = (\Delta x)^2$ throughout the paper. Since the geodesic distance on the surface, the solution to the surface Eikonal equation, can be approximated by solving a related Poisson equation, the second step recovers the distance function by solving this equation within the CPM framework, with careful attention to the extension and projection steps. Algorithm 1 provides a summary of the CPHM procedure for a single source point. The algorithm can be easily generalized to multiple source points, which will be discussed in Sec. 3.2.

Algorithm 1 Closest Point Heat Method for the Surface Eikonal Equation

1 **Input:** Grid points $\boldsymbol{x} \in \mathbb{R}^3$, surface $\mathcal{S}$, closest point map cp, time step Δt, penalty parameter γ, interpolation orders p, q

2 **Initialize:** $\mathsf{u}_0 \leftarrow \delta_{\mathrm{cp}(\boldsymbol{x}_0)}$ for a source point $\boldsymbol{x}_0 \in \mathcal{S}$

3 Compute extension matrices E_p, E_q, discrete Laplacian matrix L, and finite difference derivative matrices D_{x_1}, D_{x_2}, D_{x_3} that approximate ∂_{x_1}, ∂_{x_2}, ∂_{x_3}, respectively.

4 Initialize identity matrix I

5 $\mathsf{u}_1 \leftarrow$ `CPHeatSolve`(u_0, Δt, γ, I, E_p, E_q, L) ▷ Algorithm 2

6 $\phi \leftarrow$ `CPPoissonSolve`(u_1, γ, I, E_p, E_q, L, D_{x_1}, D_{x_2}, D_{x_3}) ▷ Algorithm 3

7 **Output:** ϕ restricted to S

3.1 Closest Point Treatment of Surface Equations

To solve PDEs on a surface $\mathcal{S}$ without requiring explicit surface parameterizations, we use the CPM. This approach extends a surface PDE on the embedded surface $\mathcal{S} \in \mathbb{R}^3$ into a narrow tubular neighborhood $B(\mathcal{S})$ where the closest point function (2.1) is well-defined. Here, we give a brief review on the CPM for a screened Poisson equation,

$$\Delta_{\mathcal{S}} u(\boldsymbol{y}) - c u(\boldsymbol{y}) = f(\boldsymbol{y}), \qquad \boldsymbol{y} \in \mathcal{S}, \tag{3.1}$$

where $c > 0$ and $\Delta_{\mathcal{S}}$ denotes the Laplace–Beltrami operator on $\mathcal{S}$. Firstly, we denote the closest point extension of the surface solution as $\tilde{u}(\boldsymbol{x}) = Eu(\boldsymbol{x})$ for $\boldsymbol{x} \in B(\mathcal{S})$. Similarly, $\tilde{f} = Ef$ in $B(\mathcal{S})$. Since $u \equiv \tilde{u}$ and $f \equiv \tilde{f}$ on $\mathcal{S}$, we can replace u by $\tilde{u}$ and f by $\tilde{f}$ in (3.1):

$$\Delta_{\mathcal{S}} \tilde{u}(\boldsymbol{y}) - c \tilde{u}(\boldsymbol{y}) = \tilde{f}(\boldsymbol{y}), \qquad \boldsymbol{y} \in \mathcal{S}. \tag{3.2}$$

Note that the closest point extension is idempotent, we have $\tilde{u} = E\tilde{u}$ in $B(\mathcal{S})$. Therefore, we have

$$\Delta_{\mathcal{S}}\tilde{u} = \Delta_{\mathcal{S}}\left(E\tilde{u}\right) = \Delta\tilde{u}\mid_{\mathcal{S}} \quad \text{on } \mathcal{S}\,, \tag{3.3}$$

where the second equality is due to the Laplacian principle (2.5). With (3.3), the surface equation (3.2) can be expressed in terms of the Cartesian Laplacian,

$$\Delta\tilde{u}\left(\boldsymbol{y}\right) - cu\left(\boldsymbol{y}\right) = \tilde{f}\left(\boldsymbol{y}\right), \qquad \boldsymbol{y} \in \mathcal{S}\,. \tag{3.4}$$

Next, we apply the closest point extension to (3.4). In this way, we obtain an embedding equation for $\tilde{u}$ in $B(\mathcal{S})$,

$$E\Delta\tilde{u}\left(\boldsymbol{x}\right) - c\tilde{u}\left(\boldsymbol{x}\right) = \tilde{f}\left(\boldsymbol{x}\right), \qquad \boldsymbol{x} \in B(\mathcal{S}) \tag{3.5a}$$

$$\text{subject to } \tilde{u}\left(\boldsymbol{x}\right) = E\tilde{u}\left(\boldsymbol{x}\right), \qquad \boldsymbol{x} \in B(\mathcal{S})\,. \tag{3.5b}$$

The constraint (3.5b) effectively enforces that the extended solution $\tilde{u}$ being a closest point extension of the surface solution u, and hence maintains constancy of $\tilde{u}$ along the normal direction of $\mathcal{S}$. Following [23], we enforce the constraint (3.5b) through a penalty formulation that discourages deviations from the extension condition,

$$E\Delta u - cu - \gamma\left(u - Eu\right) = f \ \text{ in } \ B(\mathcal{S})\,, \tag{3.6}$$

which is referred as the *embedding equation* of the surface equation (3.1). We omit the tilde on $\tilde{u}$ and $\tilde{f}$ for notational convenience. In (3.6), $\gamma > 0$ is a parameter that represents the penalty strength. Let $n \in \mathbb{N}$ be the dimension of the embedding space. Following [24, 23], we choose $\gamma = 2n/(\Delta x)^2$ to balance the accuracy and effectiveness to enforce the extension condition (3.5b).

Now, we review a matrix formulation (Sec. 2.2 in [21]) of a finite difference scheme for the embedding equation (3.6). Let $\Omega_{\Delta x}$ be the collection of rectangular grid points inside the narrow band $B(\mathcal{S})$. Assume $|\Omega_{\Delta x}| = N$. We introduce the vector $\mathsf{u} \in \mathbb{R}^N$ with entries $\mathsf{u}_i \approx u(\boldsymbol{x}_i)$ for each grid point $\boldsymbol{x}_i \in \Omega_{\Delta x}$. We approximate the Cartesian Laplacian by $\Delta u \approx \mathsf{L}\mathsf{u}$, where $\mathsf{L} \in \mathbb{R}^{N\times N}$ is a Laplacian matrix that contains the weights of the standard second-order central difference scheme. When handling the closest point extension E of a surface quantity, we need to assign the value at each grid point $\boldsymbol{x}_i$ to be the value at the corresponding closest point $\mathrm{cp}\left(\boldsymbol{x}_i\right)$. Since $\mathrm{cp}\left(\boldsymbol{x}_i\right)$ generally does not coincide with the grid points, we obtain the closest point value through polynomial interpolation of the surrounding grid values. Suppose the p-th order interpolation is used to approximate the closest point values. We may collect the interpolation weights at the stencils and incorporate them into the extension matrix $\mathsf{E}_p \in \mathbb{R}^{N\times N}$. In this way, the closest point extension Eu is numerically performed by $\mathsf{E}_p\mathsf{u}$. To balance accuracy and efficiency, we use a lower-order approximation when applying the Laplacian and higher-order interpolation for the penalty term. For example, a typical discretization of (3.1) takes the form

$$\left[\mathsf{E}_p\mathsf{L} - c\mathsf{I} - \gamma\left(\mathsf{I} - \mathsf{E}_q\right)\right]\mathsf{u} = \mathsf{f}\,. \tag{3.7}$$

where I is the N-by-N identity matrix. $\mathsf{f} \in \mathbb{R}^N$ is the discretization of f in Ω_h. E_p and E_q denote the interpolation matrices of degree-p and degree-q, respectively. A common choice is $p = 1$, $q = 3$. This strategy ensures that off-surface computations remain consistent with the surface geometry while maintaining the stability and robustness of the scheme. This section leads to a unified framework applicable to both time-dependent and steady-state surface equations in CPHM, as outlined below.

3.1.1 Time-Dependent Case: Surface Heat Equation

In the heat method, we have to compute a short-time solution to the surface heat equation

$$\begin{aligned} \partial_t u\left(\boldsymbol{y}, t\right) &= \Delta_{\mathcal{S}} u\left(\boldsymbol{y}, t\right), \quad \boldsymbol{y} \in \mathcal{S}, \ \ t > 0\,, \\ u\left(\boldsymbol{y}, 0\right) &= u_0\left(\boldsymbol{y}\right), \quad \boldsymbol{y} \in \mathcal{S}\,. \end{aligned} \tag{3.8}$$

The initial condition $u_0(\boldsymbol{y})$ is taken to be the Dirac Delta distribution at the source point $\boldsymbol{y}_0$. In CPHM, we adopt a regularized Dirac Delta initial condition. See the details in Sec. 3.2. Let $u_1(\boldsymbol{y}) \approx u(\boldsymbol{y}, \Delta t)$ be the approximated short-time heat solution. Following [17], we obtain u_1 by applying the Backward Euler method to (3.8) for one time step Δt. This leads to a screened Poisson equation for u_1,

$$\Delta_{\mathcal{S}} u_1(\boldsymbol{y}) - \frac{1}{\Delta t} u_1(\boldsymbol{y}) = -\frac{1}{\Delta t} u_0(\boldsymbol{y}), \quad \boldsymbol{y} \in \mathcal{S}. \tag{3.9}$$

Define the closest point extensions $\tilde{u}_0 = Eu_0$ and $\tilde{u}_1 = Eu_1$. The corresponding embedding equation of (3.9) for $\tilde{u}_1$ is

$$E\Delta\tilde{u}_1 - \frac{1}{\Delta t}\tilde{u}_1 - \gamma(\tilde{u}_1 - E\tilde{u}_1) = -\frac{1}{\Delta t}\tilde{u}_0 \text{ in } B(\mathcal{S}), \tag{3.10}$$

which we numerically solve with the CPM. A detailed description of the numerical procedure is summarized in Algorithm 2.

Algorithm 2 `CPHeatSolve`: Closest Point Method for Surface Heat Equation

1 **function** `CPHeatSolve`(u_0, Δt, γ, I, E_p, E_q, L)

2 Let $\mathsf{u}_1 \in \mathbb{R}^N$ with entries $[\mathsf{u}_1]_i \approx \tilde{u}_1(\boldsymbol{x}_i)$ in (3.10). Compute u_1 by solving the linear system:

$$\left[\mathsf{E}_p\mathsf{L} - \frac{1}{\Delta t}\mathsf{I} - \gamma(\mathsf{I} - \mathsf{E}_q)\right]\mathsf{u}_1 = -\frac{1}{\Delta t}\mathsf{u}_0 \tag{3.11}$$

3 **return** u_1

3.1.2 Steady-State Case: Surface Poisson Equation

To obtain the Eikonal solution, we have to solve the surface Poisson equation,

$$\Delta_{\mathcal{S}}\phi = \nabla_{\mathcal{S}} \cdot \boldsymbol{X} \text{ on } \mathcal{S}, \qquad \text{where } \boldsymbol{X} = -\nabla_{\mathcal{S}} u_1 / \|\nabla_{\mathcal{S}} u_1\|, \tag{3.12}$$

which corresponds to (3.1) with $c = 0$ and $f = \nabla_{\mathcal{S}} \cdot \boldsymbol{X}$. In order to derive the embedding equation for (3.12), we provide some details on how to perform the closest point extension of its right-hand side function f. By the gradient principle (2.3), we can replace the surface gradient of u_1 by the Cartesian gradient of $\tilde{u}_1 = Eu_1$. Therefore, we have

$$\boldsymbol{X} = -\nabla\tilde{u}_1 / \|\nabla\tilde{u}_1\| \text{ on } \mathcal{S}. \tag{3.13}$$

Since $\boldsymbol{X}$ is parallel to $\nabla_{\mathcal{S}} u_1$, hence it is tangent to $\mathcal{S}$. Define $\tilde{\boldsymbol{X}}$ as the closest point extension of the vector field $\boldsymbol{X}$, i.e., $\tilde{\boldsymbol{X}}(\boldsymbol{x}) = \boldsymbol{X}(\mathrm{cp}(\boldsymbol{x}))$. By the Divergence principle, we can replace the surface divergence of $\boldsymbol{X}$ by the Cartesian divergence of $\tilde{\boldsymbol{X}}$ when computing

$$f = \nabla_{\mathcal{S}} \cdot \boldsymbol{X} = \nabla \cdot \tilde{\boldsymbol{X}} \text{ on } \mathcal{S}. \tag{3.14}$$

Following the derivation in the previous sections, the embedding equation of $\tilde{\phi} = E\phi$ for (3.12) is

$$E\Delta\phi - \gamma(\phi - E\phi) = \tilde{f} \text{ in } B(\mathcal{S}), \quad \text{where } \tilde{f}(\boldsymbol{x}) = [\nabla \cdot \tilde{X}](\mathrm{cp}(\boldsymbol{x})). \tag{3.15}$$

The corresponding solution method is outlined in Algorithm 3.

By using the closest point extension operator and a penalty-based embedding strategy, both parabolic and elliptic surface PDEs can be treated in a consistent and efficient framework. This approach allows the use of standard Cartesian discretizations within a narrow band around the surface, simplifying implementation while preserving the intrinsic geometry of the problem.

Algorithm 3 `CPPoissonSolve`: Recovery of Distance via Closest Point Poisson Solve

function `CPPoissonSolve`(u_1, γ, I, E_p, E_q, L, D_{x_1}, D_{x_2}, D_{x_3})

Compute $\nabla \tilde{u}_1 = [\partial_{x_1}\tilde{u}_1, \partial_{x_2}\tilde{u}_1, \partial_{x_3}\tilde{u}_1]^T$ at each grid point $\boldsymbol{x}_i$ component-by-component:

$$\partial_{x_j}\tilde{u}\left(\boldsymbol{x}_i\right) \leftarrow \left[\mathsf{D}_{x_j}\mathsf{u}_1\right]_i, \qquad \text{for } j = 1, 2, 3\,.$$

Compute normalized heat gradient $\boldsymbol{X} = [X_1, X_2, X_3]^T$ at each grid point $\boldsymbol{x}_i$:

$$\boldsymbol{X}(\boldsymbol{x}_i) = -\nabla\tilde{u}_1(\boldsymbol{x}_i)/\|\nabla\tilde{u}_1(\boldsymbol{x}_i)\|\,.$$

Compute $\partial_{x_j}X_j$ for each grid point $\boldsymbol{x}_i$, for $j = 1, 2, 3$:

$$\partial_{x_j}X_j\left(\boldsymbol{x}_i\right) \leftarrow \left[\mathsf{D}_{x_j}\left[X_j(\boldsymbol{x}_1), X_j(\boldsymbol{x}_2), \cdots, X_j(\boldsymbol{x}_N)\right]^T\right]_i$$

Compute $f = \nabla\cdot\boldsymbol{X} = \partial_{x_1}X_1 + \partial_{x_2}X_2 + \partial_{x_3}X_3$ at each grid point $\boldsymbol{x}_i$, ▷ Cartesian divergence

$$f(\boldsymbol{x}_i) \leftarrow \partial_{x_j}X_1\left(\boldsymbol{x}_i\right) + \partial_{x_j}X_2\left(\boldsymbol{x}_i\right) + \partial_{x_j}X_3\left(\boldsymbol{x}_i\right)$$

Let $\mathsf{f} \in \mathbb{R}^N$ with entries $\mathsf{f}_i = f(\boldsymbol{x}_i)$. Solve for ϕ from the embedding equation:

$$\left[\mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_q)\right]\phi = \mathsf{E}_q\mathsf{f}. \tag{3.16}$$

return ϕ

3.2 Approximation of the Dirac Delta Function on a Surface

In the CPHM, the accuracy of the surface delta function approximation plays a crucial role in the overall quality of the solution. Since this approximation represents a concentrated source term, any error in its support width, placement, or the underlying geodesic distance can propagate through the simulation and degrade the fidelity of results such as heat flow or distance maps.

Let $\phi_{\mathcal{S}}(\boldsymbol{x})$ be the geodesic distance between the source point $\boldsymbol{x}_0$ and $\boldsymbol{x}$ on the surface $\mathcal{S}$. When solving the heat equation (3.8), we approximate the surface Dirac Delta initial condition by

$$\delta_{\boldsymbol{x}_0,H}(\boldsymbol{x}) = \begin{cases} \dfrac{2\pi}{(\pi^2-4)H^2}\left[1 + \cos\left(\dfrac{\pi\,\phi_{\mathcal{S}}(\boldsymbol{x})}{H}\right)\right], & \text{if} \quad \phi_{\mathcal{S}}(\boldsymbol{x}) \le H, \\ 0, & \text{if} \quad \phi_{\mathcal{S}}(\boldsymbol{x}) > H, \end{cases} \tag{3.17}$$

which is obtained by generalizing a radially symmetric approximation of the Dirac delta distribution on $\mathbb{R}^2$ with compact support of radius H [25], replacing the Euclidean distance with the geodesic distance $d_{\mathcal{S}}$.

To assign such an initial condition in a narrow band of the surface $\mathcal{S}$, we first search for the grid points whose closest points lie within a Euclidean ball of radius H centered at x_0. Then, for each of these closest points, we approximate their geodesic distance from x_0 by fitting a bivariate polynomial surface $\tilde{\mathcal{S}}$ near x_0. The geodesic distance on the original surface is then approximated by that on the fitting surface $\tilde{\mathcal{S}}$, which can be computed by solving a geodesic equation [26].

The local reconstructed surface $\tilde{\mathcal{S}}$ is defined in a local coordinate system centered at $\boldsymbol{x}_0$, where the coordinate axes are aligned with the principal directions at that point: the surface normal $\boldsymbol{N}(\boldsymbol{x}_0)$ and two orthonormal tangent vectors $\boldsymbol{t}_1(\boldsymbol{x}_0)$ and $\boldsymbol{t}_2(\boldsymbol{x}_0)$ spanning the tangent plane, as described in [27]. In this coordinate frame, any point near $\boldsymbol{x}_0$ can be expressed as $(\xi, \eta, \tilde{z})$, where (ξ, η) are coordinates in the tangent plane and $\tilde{z}$ is the height along the normal direction.

We fit a second-order polynomial surface of the form

$$\tilde{z}(\xi, \eta) = a_0 + a_1\xi + a_2\eta + a_3\xi^2 + a_4\xi\eta + a_5\eta^2,$$

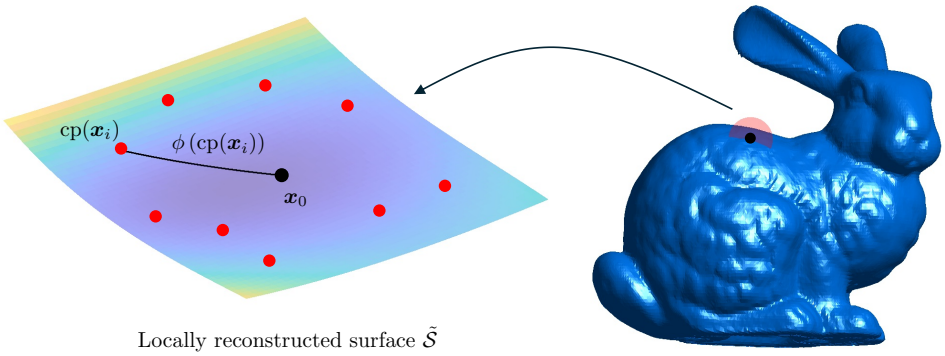

Figure 2: Here, we illustrate the local reconstruction of the surface $\mathcal{S}$ used to approximate the geodesic distance near the source point $\boldsymbol{x}_0$. The intersection of a Euclidean ball centered at $\boldsymbol{x}_0$ and the surface $\mathcal{S}$ is approximated by a bivariate polynomial surface $\tilde{\mathcal{S}}$ defined in a local coordinate system at $\boldsymbol{x}_0$. The geodesic distance between $\boldsymbol{x}_0$ and a closest point $\mathrm{cp}(\boldsymbol{x}_{i,j})$ on $\mathcal{S}$ is then approximated by the geodesic distance between the corresponding points on $\tilde{\mathcal{S}}$ within this local frame.

using least squares fitting from the local closest point data transformed into this coordinate system. This reconstructed surface $\tilde{\mathcal{S}}$ captures the local curvature of $\mathcal{S}$ near x_0 and enables accurate approximation of intrinsic quantities such as geodesic distance.

Empirically, we have found that $H = 2\Delta x$ gives the optimal accuracy. Note that this surface local reconstruction procedure can be easily generalized to multiple source points, provided that their support do not overlap.

3.3 Extension to Open Surfaces

Here, we consider an embedded surface $\mathcal{S} \subset \mathbb{R}^3$ with a smooth and co-dimension one boundary $\partial\mathcal{S}$. Such open surfaces pose two challenges to CPHM. Firstly, we must impose appropriate boundary conditions (BCs) for both the surface heat and Poisson equations on $\partial\mathcal{S}$. Secondly, we must adapt the closest point extension to the surface PDEs with the presence of BCs. Throughout this section, we let $\boldsymbol{n} = \boldsymbol{n}(\boldsymbol{y})$ be the unit outward normal at a point $\boldsymbol{y} \in \partial\mathcal{S}$.

3.3.1 Correct boundary conditions

Our objective is to compute the geodesic distance field from the source points without boundary effects; a crucial step is to compute a boundary-free heat gradient. To begin the discussion, suppose we solve the heat equation (3.8) with homogeneous Neumann (no-flux) BC,

$$\partial_n u = \langle \boldsymbol{n}, \nabla_{\mathcal{S}} u \rangle = 0 \ \text{ on } \ \partial\mathcal{S}\,.$$

Under this BC, the resulting heat gradient is orthogonal to the unit outward normal of $\partial\mathcal{S}$. Hence, we can see that the boundary geometry alters the heat gradient, which leads to a distorted Eikonal solution that differs from the geodesic distance field. To improve this, [17] proposes averaging the (homogeneous) Neumann heat solution with the Dirichlet heat solution. Such heuristic modification appears to mitigate the boundary effect. Alternatively, we consider a modified Neumann BC for (3.8)

$$\partial_n u = g\,, \ \text{ where } \ g = \langle \boldsymbol{n}, \nabla_{\mathcal{S}} u \rangle\,. \tag{3.18a}$$

In this formalism, the heat gradient $\nabla_{\mathcal{S}} u$ is completely determined by the heat source. At the boundary, the normal derivative of the heat solution is defined by the inner product of the heat gradient and the

unit outward normal of $\partial\mathcal{S}$. This self-consistent BC does not interfere with the heat propagation from the source points. Consequently, we obtain the heat gradient that has the boundary effects eliminated. As in Sec. 3.1.2, we impose the Eikonal gradient to be the normalized heat gradient $\boldsymbol{X} = -\nabla_{\mathcal{S}} u / \|\nabla_{\mathcal{S}} u\|$ on $\mathcal{S} \cup \partial\mathcal{S}$. This constraint leads to the minimization problem $\min_{\phi} \|\nabla_{\mathcal{S}}\phi - \boldsymbol{X}\|_2$. The Euler–Lagrange equation of this problem is the Poisson equation (3.12) with the Neumann BC

$$\partial_n \phi = \langle \boldsymbol{n}, \boldsymbol{X} \rangle \text{ on } \partial\mathcal{S}\,. \tag{3.18b}$$

In conclusion, we supplement the surface heat and Poisson equations with the correct BCs (3.18a) and (3.18b), respectively.

3.3.2 Modified CPHM for boundary condition

To handle an open surface, we adopt a modified closest point function [28] for the grid points near the boundary. Moreover, the discretization of the inhomogeneous Neumann BC requires special care such that the second-order accuracy of the Poisson solver can be retained. We leave the details in the Appendix A. Firstly, the discretization (3.11) of the heat solution is changed to

$$\left(\bar{\mathsf{E}}_p \mathsf{L} - \frac{1}{\Delta t}\mathsf{I}\right) \mathsf{u}_1 - \gamma(\mathsf{u}_1 - \bar{\mathsf{E}}_q \mathsf{u}_1 - \mathsf{g}) = -\frac{1}{\Delta t}\mathsf{u}_0 \tag{3.19}$$

where $\bar{\mathsf{E}}_p$ and $\bar{\mathsf{E}}_q$ are the interpolation matrices of order p and q at the modified closest points, respectively. The vector $\mathsf{g} \in \mathbb{R}^N$ encodes a second-order accurate discretization of the Neumann BC (3.18a). Since the right-hand side function g of (3.18a) is a linear operator of the heat solution u, its corresponding discretization g is also a linear with respects to the discretized heat solution u_1. Hence, we can explicitly write

$$\mathsf{g} = \bar{\mathsf{E}}_g \mathsf{u}_1\,, \text{ for some matrix } \bar{\mathsf{E}}_g \in \mathbb{R}^{N\times N}\,. \tag{3.20}$$

Upon substituting (3.20) into (3.19), u_1 now satisfies the linear system:

$$\left[\bar{\mathsf{E}}_p \mathsf{L} - \frac{1}{\Delta t}\mathsf{I} - \gamma(\mathsf{I} - \bar{\mathsf{E}}_q - \bar{\mathsf{E}}_g)\right] \mathsf{u}_1 = -\frac{1}{\Delta t}\mathsf{u}_0 \tag{3.21}$$

Similarly, for the discrete Poisson solution ϕ in (3.16), a second-order discretization of the Neumann BC (3.18b) modifies the extension constraint to $\phi = \mathsf{E}_q\phi + \mathsf{g}_2$, where $\mathsf{g}_2 \in \mathbb{R}^N$ encodes a second-order accurate discretization of the Neumann BC (3.18b). Upon substituting the modified constraint into (3.16), we have

$$\left[\bar{\mathsf{E}}_p \mathsf{L} - \gamma(\mathsf{I} - \bar{\mathsf{E}}_q)\right] \phi = \bar{\mathsf{E}}_q \mathsf{f} + \gamma \mathsf{g}_2\,. \tag{3.22}$$

These modifications allow CPHM to incorporate second-order accurate treatments of Neumann boundary conditions on open surfaces while maintaining the overall structure of the original algorithm.

Remark (On the Accuracy of CPHM). Although CPHM incorporates higher-order accurate discretizations for interpolation and the Laplacian within the closest point framework, the overall accuracy of the method remains first-order, consistent with the original heat method [17]. This limitation primarily stems from the design of `CPHeatSolve` (Algorithm 2), which adopts the same time discretization strategy, using a time step of $\Delta t = (\Delta x)^2$ to ensure stability. The resulting distance function inherits the $\mathcal{O}(\Delta x)$ error introduced by the short-time heat flow approximation, which only asymptotically recovers the gradient direction of the true geodesic distance. We also observe a potential further reduction in accuracy for certain open surface problems, primarily due to errors in the boundary condition approximation. A more detailed investigation into the accuracy of CPHM will be provided in Section 4.1.

4 Numerical results

In this section, we present numerical results to validate the performance of the proposed method, CPHM. The first subsection is dedicated to a convergence study, which includes two parts: numerical tests on the unit sphere, a closed surface, with both single and multiple point sources; and experiments on an open surface to evaluate the method's behavior near boundaries. A series of examples on complex surfaces follows, demonstrating the versatility of CPHM in accurately capturing geodesic distances on geometries with intricate features and topology. The penalty parameter γ is chosen to be $2n/\left(\Delta x\right)^2$, where n is the dimension of the data, and this value is used consistently across all examples. All computations were implemented in MATLAB using code based on the GitHub repository at `https://github.com/cbm755/cp_matrices`, and executed on a personal laptop (Apple M1 Pro, 3.2 GHz processor, 16 GB memory). Every linear system arising in the CPHM algorithm (Algorithm 1) is solved using MATLAB's `backslash` operator. We note that one could also adopt a multigrid approach, as in [23], to enhance the efficiency of the solver.

4.1 Convergence Study

4.1.1 Closed Surface: Unit Sphere

We begin our convergence study of CPHM with a simple benchmark problem. Let $\mathcal{S}$ be the surface of the unit sphere $\|\boldsymbol{x}\| = 1$ embedded in $\mathbb{R}^3$. We adopt the spherical coordinates (φ, θ), where $\varphi \in [0, 2\pi)$ represents the azimuth angle and $\theta \in [0, \pi/2]$ denotes the co-latitudinal angle measured from the positive z-axis. In the first example, we set a single source point at $(\varphi, \theta) = (\pi/4, \pi/3)$. The Eikonal solution shown in Fig. 3(a,b) exhibits a smooth profile and equispaced contour lines that are consistent with the exact geodesic distance on the unit sphere. The plots of relative error in Fig. 3(c) confirm that the proposed method achieves almost first-order convergence, in agreement with the original heat method [17]. We give the number of grid points inside the computational band (Length(ϕ^h)) and computational times for the main steps of CPHM, including the total time for local reconstruction in approximating Dirac Delta initial condition, the heat solver, and the Poisson solver. It is noteworthy that the computational time for local reconstruction is largely independent of the mesh size Δx. It is because the support H of the smoothened Dirac Delta initial condition scales with Δx (we choose $H = 2\Delta x$) In the second example, we consider five source points at $(\varphi, \theta) = (0, 0)$, $(\pm\,\pi/3, \pm\,\pi/3)$. Unlike the case of a single source point, the CPHM Eikonal solution in Fig. 4(a-c) is able to accurately capture the kinks at the intersection of characteristic curves of the Eikonal equation, supported with a first-order convergence in Fig. 4(d).

Table 1: Computational complexity and times for Fig. 3.

Δx	Length(ϕ^h)	Time (Local reconstruction)	Time (Heat solver)	Time (Poisson solver)
0.1	10,906	0.7072 (s)	0.2387 (s)	0.2193 (s)
0.05	41,870	0.5982 (s)	1.6003 (s)	1.6593 (s)
0.025	166,390	0.4262 (s)	15.7687 (s)	16.1150 (s)
0.0125	663,454	0.4366 (s)	306.1670 (s)	191.4646 (s)

4.1.2 Open Surfaces

First, to validate the proposed boundary condition in Sec.3.3, we apply the modified CPHM to the (planar) unit disk embedded in $\mathbb{R}^2$. We observe that the Eikonal solution in Fig. 5 propagates correctly near the boundary. The relative error in L_∞-norm confirms the first-order convergence of the modified CPHM solution (Fig.5(b)).

Next, we consider the upper hemisphere $\{(x, y, z) \in \mathbb{R}^3 : x^2 + y^2 + z^2 = 1, z > 0\}$. As shown in Fig. 6(a-b), the CPHM Eikonal solution is qualitatively correct, particularly near the boundary.

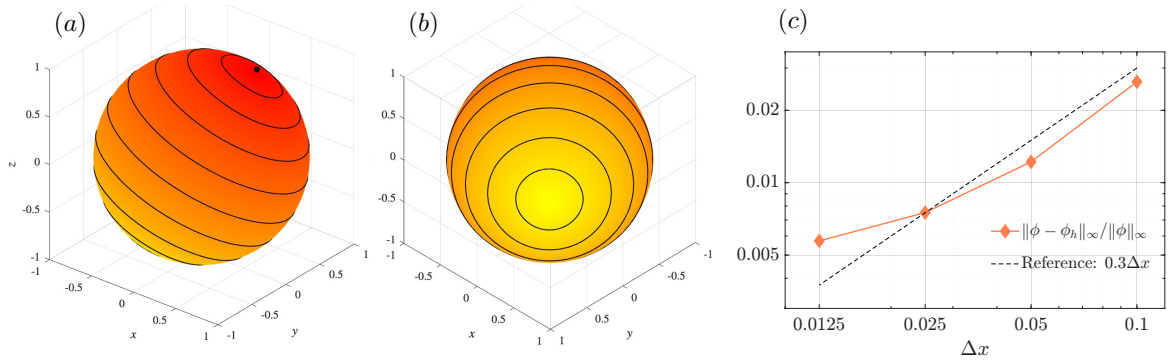

Figure 3: (a) CPHM Eikonal solution obtained with $\Delta x = 0.05$ (number of grid points is 41,870, see Table 1). The source point is located by the black marker. In (b), we show the eikonal solution from the bottom view. (c) We show the first-order convergence in the relative error l_∞ norm.

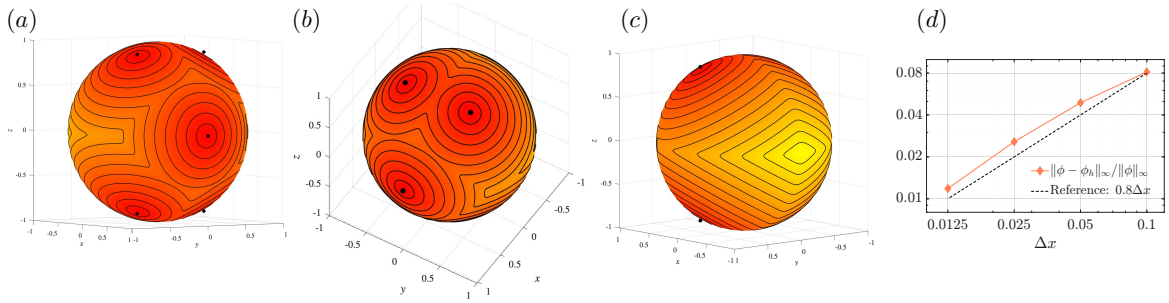

Figure 4: (a), (b), (c): CPHM Eikonal solution with 5 source points, and $\Delta x = 0.025$ (number of grid points is 166,390) in different viewing angles. Each source point is located by a black marker. In (d), we show the first-order convergence in the relative errors in l_∞ norm.

However, the relative error converges slower than first-order (see Fig. 6(c)). We attribute the slow convergence to error propagation from the heat solver into the Poisson solver. More precisely, the right-hand side of (3.12) is computed by taking the numerical divergence of the normalized gradient of the discrete heat solution. Since the heat equation (3.8) is solved with a standard second-order Laplacian discretization, the resulting (normalized) gradient is only first-order accurate. Thus the Poisson right-hand side carries an $O(\Delta x)$ truncation error that limits the overall convergence rate. This accuracy bottleneck is inherent to the heat-method pipeline, not to the proposed boundary condition or the underlying closest point framework: as shown for the unit disk (Fig. 5(b)), we do obtain first-order convergence. The boundary condition is correct at the continuous level, but a first-order convergence may require higher-accuracy discretizations in both the heat and Poisson solvers. Open surfaces therefore remain an important challenge for further improving the heat method.

4.2 Miscellaneous Examples

We present the Eikonal solution with a single source on various surfaces in Fig. 7: Bunny, Bumpy Sphere, Pig, Hippo, and Sappho's Head. The corresponding number of grid points (inside the computational band) is provided in Table 2. Each row shows a different surface geometry, with the left column displaying the surface mesh, and the middle and right columns illustrating the computed geodesic distance via isolines from different viewpoints. The smoothness and density of the contours demonstrate

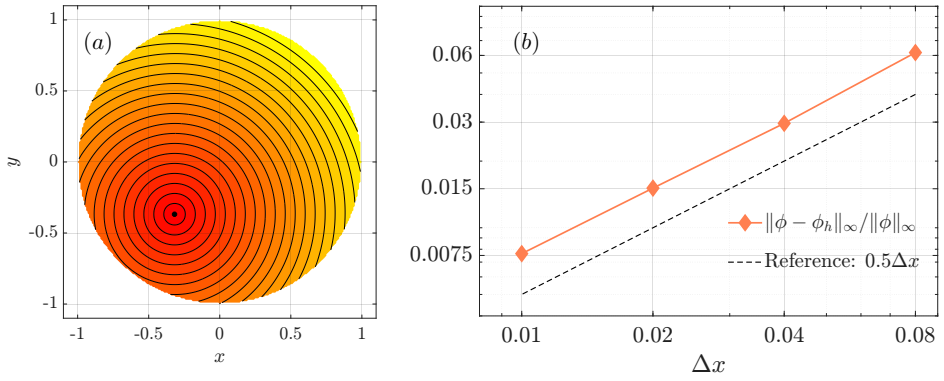

Figure 5: (a): CPHM Eikonal solution of the planar unit disk with source point $\boldsymbol{x}_0 = -(\pi^{-1}, e^{-1})$ obtained with $\Delta x = 0.01$ (number of grid points = 33,745). (b): First-order convergence of relative errors in L_∞-norm.

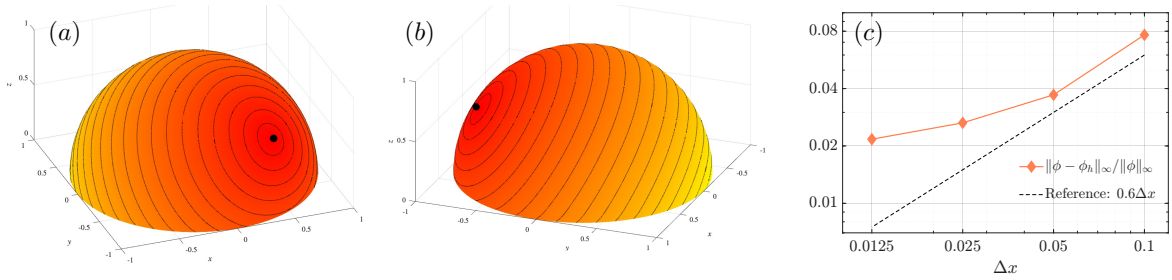

Figure 6: (a), (b): CPHM Eikonal solution on the upper hemisphere with source point (shown as black marker) at $(\varphi, \theta) = (5\pi/3,\, 3\pi/10)$ obtained with $\Delta x = 0.025$ (number of grid points = 89,989). (b): The relative errors in L_∞-norm.

the effectiveness of the method in capturing intrinsic distances across a variety of topologies and surface complexities.

5 Concluding Remarks

In this paper, we presented the Closest Point Heat Method (CPHM), a novel approach for solving the surface Eikonal equation on general smooth surfaces. By extending the heat method to an embedding framework using the closest point methodology, CPHM overcomes several limitations of traditional techniques that rely on surface meshes or parametrizations. Our method enables intrinsic geodesic distance computations without requiring explicit surface discretizations, making it particularly effective for implicit surfaces or data represented as level sets or point clouds.

The key strengths of CPHM lie in its simplicity, mesh-free nature, and compatibility with standard finite difference tools on Cartesian grids. Through numerical experiments, we demonstrated the accuracy and convergence of the method on benchmark geometries and illustrated its applicability to complex shapes. The method maintains robustness even in the presence of geometric irregularities, while preserving the desirable properties of the original heat method, such as efficiency and stability.

Several promising directions remain for future research. A primary avenue is the enhancement of numerical accuracy and efficiency through adaptive or higher-order discretizations, as well as through the use of CPHM to generate high-quality initializations, for example, for learning-based methods

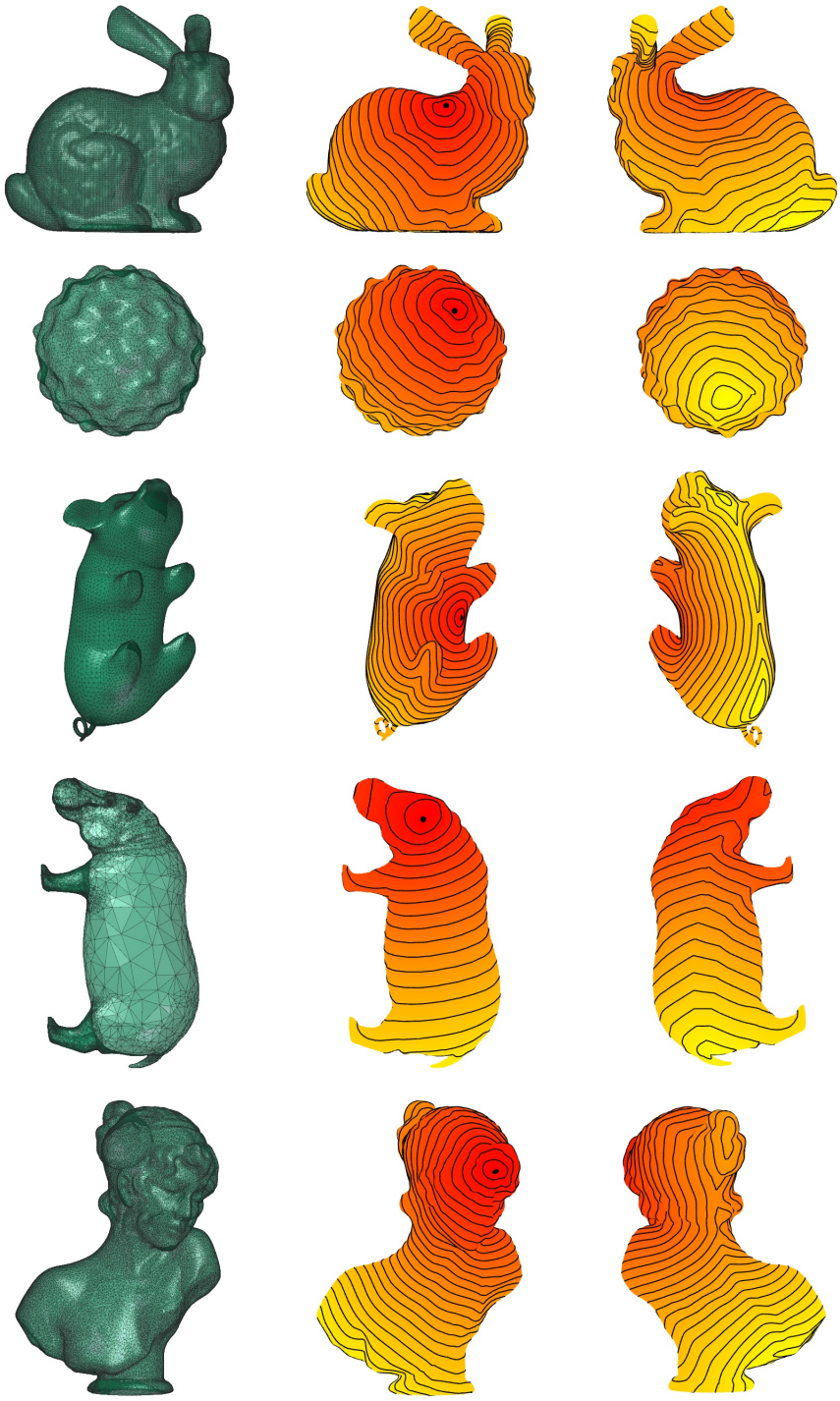

Figure 7: CPHM solution of the Eikonal equation $\|\nabla_{\mathcal{S}}\phi\| = 1$ on various surfaces (in vertical order): Bunny, Bumpy Sphere, Pig, Hippo, and Sappho's Head. The numerical resolution for each case can be found in Table 2.

[30, 31]. These improvements can be particularly beneficial in geometrically complex regions or learning

Example	Length(ϕ_h)
Bunny	108,643
Bumpy sphere	128,591
Pig	278,573
Hippo	396,311
Sappho's head [29]	316,553

Table 2: Summary of grid resolutions used in Fig. 7.

frameworks where precise intrinsic information is crucial. Another central challenge is the extension of CPHM to anisotropic settings, where the speed function depends not only on position but also on direction. The anisotropic Eikonal equation arises in applications such as image processing, medical imaging, and fiber tractography, where wavefront propagation must align with directional features of the medium. Formally, it takes the form $\|A(\boldsymbol{y})\nabla_{\mathcal{S}}\phi(\boldsymbol{y})\| = 1$, where $A(\boldsymbol{y})$ is a position-dependent tensor encoding local anisotropy. At present, our method is limited to the isotropic case with constant unit speed ($F(\boldsymbol{y}) = 1$), and even the extension to general spatially varying $F(\boldsymbol{y})$ remains an open challenge. Addressing the anisotropic case would require fundamental modifications to the heat flow approximation and projection operations, making it a natural but ambitious direction for future work.

A Implementation of inhomogeneous Neumann boundary condition

When applying the CPHM to an open surface $\mathcal{S}$ in Sec. 3.3, we impose inhomogeneous Neumann BCs for the surface heat and Poisson equations. To the best of our knowledge, extending the CPM to handle general boundary conditions (besides homogeneous Neumann and Dirichlet BCs) remains an open challenge. Here, we generalize the modified closest point function (for open surfaces) proposed in [28] to treat an inhomogeneous Neumann BC. First, we first introduce the terminologies that lead to the modified closest point function.

Let's consider a tubular neighbourhood $\mathcal{T}$ of an open surface $\mathcal{S}$ such that all grid points inside $\mathcal{T}$ have a unique closest point on $\mathcal{S}$. Let $v := u \circ \mathrm{cp}$ be the closest point extension of a surface function u defined on $\mathcal{S}$. For any grid point $\boldsymbol{x}_g \in \mathcal{T}$, we have $v(\boldsymbol{x}_g) = v\left(\mathrm{cp}(\boldsymbol{x}_g)\right)$. Therefore, the closest point extension propagates the boundary values into the $\mathcal{T}$ along the normal directions to the boundary. In other words, when applied to an open surface, the closest point extension effectively imposes the homogeneous Neumann BC $\partial_n u = 0$ on ∂S. See Fig. 8.

Now, let's consider the modified closest point function from [28]

$$\bar{\mathrm{cp}}\left(\boldsymbol{x}_g\right) := \mathrm{cp}\left(\boldsymbol{x}_g + \mathrm{cp}(\boldsymbol{x}_g) - \boldsymbol{x}_g\right) = \mathrm{cp}\left(2\mathrm{cp}(\boldsymbol{x}_g) - \boldsymbol{x}_g\right) .$$

In here, $2\mathrm{cp}(\boldsymbol{x}_g) - \boldsymbol{x}_g$ is the "mirror point" of $\boldsymbol{x}_g$ in the direction $\boldsymbol{x}_g - \mathrm{cp}(\boldsymbol{x}_g)$ across $\mathcal{S}$. Suppose that $\boldsymbol{x}_g - \mathrm{cp}(\boldsymbol{x}_g)$ is orthogonal to $\mathcal{S}$. Then both $\boldsymbol{x}_g$ and its mirror point $2\mathrm{cp}(\boldsymbol{x}_g) - \boldsymbol{x}_g$ share the same orthogonal projection, $\mathrm{cp}(\boldsymbol{x}_g)$, on $\mathcal{S}$. That is, $\mathrm{cp}(\boldsymbol{x}_g) = \bar{\mathrm{cp}}(\boldsymbol{x}_g)$, which is true when $\boldsymbol{x}_g$ is away from the boundary ∂S. This naturally leads to a classification of boundary points in the embedding space. Specifically, we say that $\boldsymbol{x}_g$ is a boundary point if $\mathrm{cp}(\boldsymbol{x}_g) \in \partial\mathcal{S}$. Consequently, the vector $\boldsymbol{x}_g - \mathrm{cp}(\boldsymbol{x}_g)$ is not orthogonal to $\mathcal{S}$ and $\mathrm{cp}(\boldsymbol{x}_g) \neq \bar{\mathrm{cp}}(\boldsymbol{x}_g)$. See Fig. 8. In this case, we identify $\boldsymbol{x}_g$ as a ghost point, whose value is treated as a boundary value. By replacing all instances of $\mathrm{cp}(\boldsymbol{x}_g)$ with $\bar{\mathrm{cp}}(\boldsymbol{x}_g)$ in the discretization stencil, the modified closest point extension yields a second-order accurate discretization of the homogeneous Neumann BC. We let the modified interpolation matrix of order q be $\bar{\mathsf{E}}_q$. Numerically, the modified closest point extension is enforced by the discrete constraint,

$$\mathsf{u} = \bar{\mathsf{E}}_q \mathsf{u} . \tag{A.1}$$

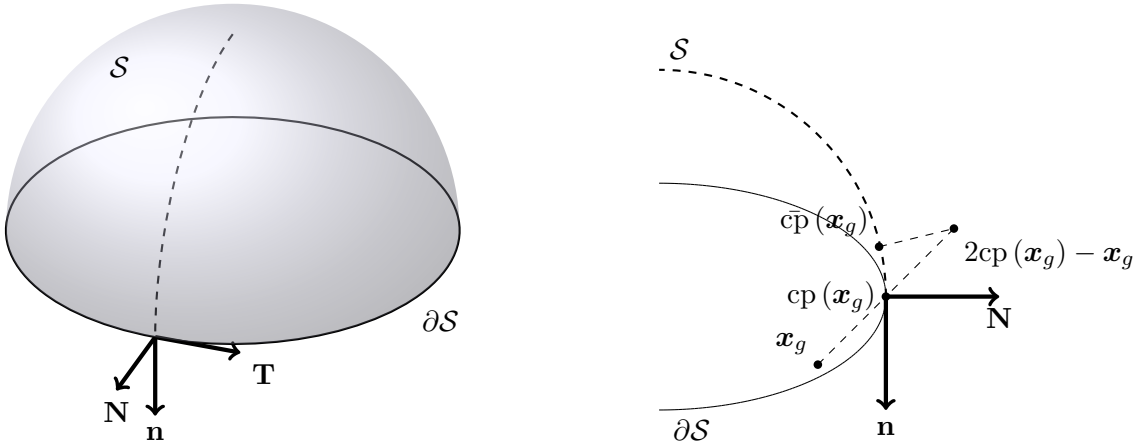

Figure 8: Illustration of mirror point construction near the boundary $\partial\mathcal{S}$ of the hemisphere $\mathcal{S}$. (Left) The surface normal $\boldsymbol{n}$, boundary normal $\boldsymbol{N}$, and tangential direction $\boldsymbol{T}$ at a boundary point. (Right) A grid point $\boldsymbol{x}_g$, its closest point $\text{cp}\,(\boldsymbol{x}_g)$, and the corresponding mirror point $2\text{cp}\,(\boldsymbol{x}_g) - \boldsymbol{x}_g$.

Now, we utilize the modified closest point extension to handle the inhomogeneous Neumann BC $\partial_n u = g$ on ∂S. At a boundary point $\boldsymbol{x}_g$, let $\boldsymbol{T}$ and $\boldsymbol{N}$ be a unit tangent and normal vector of $\mathcal{S}$ such that $\boldsymbol{n} := \boldsymbol{T} \times \boldsymbol{N}$ is the unit outward normal of ∂S. Note that $\{\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{n}\}$ is an orthonormal basis of $\mathbb{R}^3$. Let $\boldsymbol{w} = \frac{\boldsymbol{x}_g - \text{cp}(\boldsymbol{x}_g)}{||\boldsymbol{x}_g - \text{cp}(\boldsymbol{x}_g)||}$. Since $\boldsymbol{w}$ is orthogonal to $\boldsymbol{T}$, we have

$$\boldsymbol{w} = \langle \boldsymbol{w}, \boldsymbol{n} \rangle \, \boldsymbol{n} + \langle \boldsymbol{w}, \boldsymbol{N} \rangle \boldsymbol{N} \, . \tag{A.2}$$

From (A.2), we may decompose the directional derivative in the $\boldsymbol{w}$-direction as

$$\partial_w = \langle \boldsymbol{w}, \boldsymbol{n} \rangle \, \partial_n + \langle \boldsymbol{w}, \boldsymbol{N} \rangle \, \partial_N \, .$$

Apply this to $u\big(\text{cp}(\boldsymbol{x}_g)\big)$, we have

$$\partial_w \, u\big(\text{cp}(\boldsymbol{x}_g)\big) = \langle \boldsymbol{w}, \boldsymbol{n} \rangle \, g\big(\text{cp}(\boldsymbol{x}_g)\big) \, . \tag{A.3}$$

where we use the given BC $\partial_n u = g$ at $\boldsymbol{x} = \text{cp}(\boldsymbol{x}_g)$, and the fact that $u\big(\text{cp}(\boldsymbol{x})\big)$ is constant along the normal direction of S, hence $\partial_N u\big(\text{cp}(\boldsymbol{x}_g)\big) = 0$. Next, we approximate the LHS of (A.3) by the central difference,

$$\partial_w u\big(\text{cp}(\boldsymbol{x}_g)\big) \approx \frac{u(\boldsymbol{x}_g) - u\big(\bar{\text{cp}}(\boldsymbol{x}_g)\big)}{2||\boldsymbol{x}_g - \text{cp}(\boldsymbol{x}_g)||} \, . \tag{A.4}$$

This yields

$$\frac{u(\boldsymbol{x}_g) - u(\bar{\text{cp}}(\boldsymbol{x}_g))}{2\|\boldsymbol{x}_g - \bar{\text{cp}}(\boldsymbol{x}_g)\|} \approx \langle \boldsymbol{w}, \boldsymbol{n} \rangle \, g(\text{cp}(\boldsymbol{x}_g)),$$

which eventually leads to a second-order accurate extrapolation formula:

$$u(\boldsymbol{x}_g) = u\,(\bar{\text{cp}}(\boldsymbol{x}_g)) + \langle \boldsymbol{x} - \text{cp}(\boldsymbol{x}), \, \boldsymbol{n} \rangle \, g(\text{cp}(\boldsymbol{x})). \tag{A.5}$$

This modifies the constraint (A.1) to

$$\mathsf{u} = \bar{\mathsf{E}}_q \mathsf{u} + \mathsf{g} \, , \tag{A.6}$$

where g is the vector that encodes the inhomogeneous Neumann BC (the second term of the RHS in (A.5)) at the boundary grid points. To demonstrate the second-order convergence with the mirror

point treatment of the inhomogeneous Neumann BC, we consider a shifted Poisson equation (3.1) on the upper hemisphere $\mathcal{S}$,

$$\Delta_{\mathcal{S}} u - u = f \text{ in } \mathcal{S}, \qquad \partial_n u = g \text{ on } \partial\mathcal{S},$$

In spherical coordinate $(\varphi, \theta) \in [0, 2\pi) \times (0, \pi)$, the exact solution is chosen to be $u(\varphi, \theta) = \sin\theta \cos\theta \cos\varphi$. Then we have $f = -7u$ and $g = -\cos\varphi$. We present the relative error and the convergence order in Table 3.

Δx	$\frac{\|u - u_h\|_\infty}{\|u\|_\infty}$	Order
0.1	6.6396E-3	—
0.05	1.8217E-3	1.8658
0.025	4.7954E-4	1.9256
0.0125	1.2362E-4	1.9557

Table 3: Second-convergence of the inhomogeneous Neumann problem

Acknowledgement

The research of Byungjoon Lee was supported by the Catholic University of Korea, Research Fund. The authors are grateful to anonymous reviewers for their careful reading and valuable comments.

References

[1] Grimshaw R. Propagation of surface waves at high frequencies. IMA Journal of Applied Mathematics. 1968;4(2):174-93.

[2] Mémoli F, Sapiro G. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. Journal of computational Physics. 2001;173(2):730-64.

[3] Liu J, Leung S. A splitting algorithm for image segmentation on manifolds represented by the grid based particle method. Journal of Scientific Computing. 2013;56:243-66.

[4] Bronstein MM, Bruna J, LeCun Y, Szlam A, Vandergheynst P. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine. 2017;34(4):18-42.

[5] Kimmel R, Sethian JA. Computing geodesic paths on manifolds. Proceedings of the national academy of Sciences. 1998;95(15):8431-5.

[6] Tsitsiklis JN. Efficient algorithms for globally optimal trajectories. IEEE transactions on Automatic Control. 2002;40(9):1528-38.

[7] Zhao H. A fast sweeping method for eikonal equations. Mathematics of computation. 2005;74(250):603-27.

[8] Yoo SW, Seong JK, Sung MH, Shin SY, Cohen E. A triangulation-invariant method for anisotropic geodesic map computation on surface meshes. IEEE Transactions on Visualization and Computer Graphics. 2012;18(10):1664-77.

[9] Xu SG, Zhang YX, Yong JH. A fast sweeping method for computing geodesics on triangular manifolds. IEEE transactions on pattern analysis and machine intelligence. 2008;32(2):231-41.

[10] Wong T, Leung S. A fast sweeping method for eikonal equations on implicit surfaces. Journal of Scientific Computing. 2016;67:837-59.

[11] Bronstein AM, Bronstein MM, Kimmel R. Weighted distance maps computation on parametric three-dimensional manifolds. Journal of Computational Physics. 2007;225(1):771-84.

[12] Spira A, Kimmel R. An efficient solution to the eikonal equation on parametric manifolds. Interfaces and Free Boundaries. 2004;6(3):315-27.

[13] Weber O, Devir YS, Bronstein AM, Bronstein MM, Kimmel R. Parallel algorithms for approximation of distance maps on parametric surfaces. ACM Transactions on Graphics (TOG). 2008;27(4):1-16.

[14] Huynh E, Parkinson C. A Scalable Method for Optimal Path Planning on Manifolds via a Hopf-Lax Type Formula. arXiv preprint arXiv:241213346. 2024.

[15] Chow YT, Darbon J, Osher S, Yin W. Algorithm for overcoming the curse of dimensionality for state-dependent Hamilton-Jacobi equations. Journal of Computational Physics. 2019;387:376-409.

[16] Lee B, Darbon J, Osher S, Kang M. Revisiting the redistancing problem using the Hopf–Lax formula. Journal of Computational Physics. 2017;330:268-81.

[17] Crane K, Weischedel C, Wardetzky M. Geodesics in heat: A new approach to computing distance based on heat flow. ACM Transactions on Graphics (TOG). 2013;32(5):1-11.

[18] King N, Su H, Aanjaneya M, Ruuth S, Batty C. A Closest Point Method for PDEs on manifolds with interior boundary conditions for geometry processing. ACM Transactions on Graphics. 2024;43(5):1-26.

[19] Ruuth SJ, Merriman B. A simple embedding method for solving partial differential equations on surfaces. Journal of Computational Physics. 2008;227(3):1943-61.

[20] Osher S, Sethian JA. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. Journal of computational physics. 1988;79(1):12-49.

[21] Macdonald CB, Ruuth SJ. The implicit closest point method for the numerical solution of partial differential equations on surfaces. SIAM Journal on Scientific Computing. 2010;31(6):4330-50.

[22] Varadhan SRS. On the behavior of the fundamental solution of the heat equation with variable coefficients. Communications on Pure and Applied Mathematics. 1967;20(2):431-55.

[23] Chen Y, Macdonald CB. The closest point method and multigrid solvers for elliptic equations on surfaces. SIAM Journal on Scientific Computing. 2015;37(1):A134-55.

[24] von Glehn I, März T, Macdonald CB. An embedded method-of-lines approach to solving partial differential equations on surfaces. arXiv preprint arXiv:13075657. 2013.

[25] Hosseini B, Nigam N, Stockie JM. On regularizations of the Dirac delta distribution. Journal of Computational Physics. 2016;305:423-47.

[26] Kasap E, Yapici M, Akyildiz FT. A numerical study for computation of geodesic curves. Applied Mathematics and Computation. 2005;171(2):1206-13.

[27] Leung S, Zhao H. A grid based particle method for moving interface problems. Journal of Computational Physics. 2009;228(8):2993-3024.

[28] Macdonald CB, Brandman J, Ruuth SJ. Solving eigenvalue problems on curved surfaces using the closest point method. Journal of Computational Physics. 2011;230(22):7944-56.

[29] Zhou Q, Jacobson A. Thingi10K: A Dataset of 10,000 3D-Printing Models. arXiv preprint arXiv:160504797. 2016.

[30] Smith JD, Azizzadenesheli K, Ross ZE. Eikonet: Solving the eikonal equation with deep neural networks. IEEE Transactions on Geoscience and Remote Sensing. 2020;59(12):10685-96.

[31] bin Waheed U, Haghighat E, Alkhalifah T, Song C, Hao Q. PINNeik: Eikonal solution using physics-informed neural networks. Computers & Geosciences. 2021;155:104833.