
GHOST POINT DIFFUSION MAPS FOR SOLVING ELLIPTIC PDE'S ON MANIFOLDS WITH CLASSICAL BOUNDARY CONDITIONS

A PREPRINT

Shixiao Willing Jiang

Department of Mathematics
The Pennsylvania State University, University Park, PA 16802, USA
suj235@psu.edu

John Harlim

Department of Mathematics, Department of Meteorology and Atmospheric Science,
Institute for Computational and Data Sciences
The Pennsylvania State University, University Park, PA 16802, USA
jharlim@psu.edu

December 8, 2021

ABSTRACT

In this paper, we extend the class of kernel methods, the so-called diffusion maps (DM), and its local kernel variants, to approximate second-order differential operators defined on smooth manifolds with boundary that naturally arise in elliptic PDE models. To achieve this goal, we introduce the Ghost Point Diffusion Maps (GPDM) estimator on an extended manifold, identified by the set of point clouds on the unknown original manifold together with a set of ghost points specified along the tangential direction at the boundary of the original manifold. The resulting GPDM estimator restricts the standard DM matrix to a set of extrapolation equations that estimates the function values on the ghost points. This adjustment is analogous to the usual correction on the matrix components near the boundary in classical finite-difference methods. As opposed to the classical DM which diverges near the boundary, the proposed GPDM estimator converges pointwise even near the boundary, assuming that the extended manifold is smooth. Applying the consistent GPDM estimator to solve the well-posed elliptic PDEs with classical boundary conditions (Dirichlet, Neumann, and Robin), we establish the convergence of the approximate solution under appropriate smoothness assumptions. We validate the proposed mesh-free PDE solver with supporting numerical examples on various problems, defined on simple manifolds embedded in 2D to 5D Euclidean spaces as well as on an unknown manifold. Numerically, we also find that the GPDM is more accurate compared to DM in solving eigenvalue problems associated with the second-order differential operators on bounded smooth manifolds.

Keywords Ghost Point Diffusion Maps · Diffusion Maps · Elliptic PDEs on Manifolds · Mesh-free PDE solvers

1 Introduction

Elliptic Partial Differential Equations (PDEs) [1] arise naturally in the modeling of physical phenomena, including groundwater flow [2], heat conduction [3], neutron diffusion [4], and probability theory [5]. In the manifold setting, there are many methods have been developed, especially for two-dimensional surfaces. Most of these methods, however, require a parameterization of the surface which is used to approximate the tangential derivatives along the surface. For example, the finite element method (FEM) represents the surface [6, 7, 8] using triangular meshes. The approach in [9, 10] represents the surface using level sets. The closest point method [11] uses the closest point representation of the surface. The mesh-free radial basis function (RBF) method represents the surface using a

distance function such that the surface is a level set of the function [12]. See also the references in [12, 13] for a more comprehensive literature review on this topic.

Unlike the approaches mentioned above, we consider approximating the second-order elliptic differential operators directly on the point clouds that lie on the manifold. Our approach rests on the fact that away from the boundary, these differential operators can be approximated with integral operators defined with appropriate Gaussian kernels, which is the theoretical underpinning of the popular nonlinear manifold learning algorithm known as the Diffusion Maps [14] and its local kernel variants [15]. The key advantage of such approximation is that one can avoid the parameterization of the manifold of the methods discussed in the previous paragraph, which can be complicated if the low-dimensional manifold is embedded in a high-dimensional space. As of the authors' knowledge, the idea of using such kind of integral operator for solving PDEs was first numerically realized by the Point Integral Method (PIM) for solving Poisson problems [16] and isotropic elliptic equations [13]. In separate works, the same idea was realized with the Diffusion Maps (DM) algorithm [15] for solving elliptic PDEs with non-symmetric advection-diffusion (Kolmogorov) operators associated with Itô diffusion [17] and anisotropic diffusion [18]. We should point out that despite having the same vein, DM and PIM approaches are not exactly similar as pointed out in [17].

On manifolds with boundary, however, the Neumann problem is the only natural boundary condition for the Gaussian kernel integral approximation as noted in [14]. Furthermore, as we shall see in this paper, even if the function satisfies the Neumann boundary condition, the diffusion maps integral approximation does not converge in the pointwise sense at interior points close to the boundary. For other types of boundary conditions, there are several approaches have been proposed. For example, the PIM approximates the Dirichlet problem with an artificial Robin boundary condition with a small first-order derivative term [16]. Another approach is to use a volume constraint [19], which is a simple version of the ghost point method that is proposed in the current paper, as they set the function value at the ghost points to be zero. In [20], they proposed an empirical approach for the Dirichlet problem by appending the discrete representation of the integral approximation at the interior points with a discrete representation of the Dirichlet boundary condition. All of these approaches, however, do not improve the integral approximation on the interior points near the boundary and it is unclear whether they can be extended to Robin boundary condition.

As an alternative to these approaches, a recent work [21] suggests that the diffusion maps asymptotic expansion is a consistent estimator of Laplacian in a weak sense and they devised a boundary integral estimator to specify the desired boundary conditions. In this paper, we introduce the Ghost Point Diffusion Maps (GPDM) as a consistent estimator in the sense of pointwise, complementary to the result in [21]. The GPDM modifies the classical DM, following the classical ghost points method to obtain a higher-order finite-difference approximation of Neumann problem (e.g. [22]), which is nontrivial for non-flat geometry. The core of our method is a combination of a kernel and finite-difference approximation for normal vectors at the boundary of 1D and 2D smooth manifolds, which, in turns, allows for the specification of the ghost points and the approximation of the directional derivatives on the normal component at the boundary. We show that the proposed approach converges pointwise even for points close to the boundary when the function values at the ghost points are extrapolated with a set of discrete constraints that resemble matching the second-order derivatives. Subsequently, we apply the GPDM to solve elliptic PDEs with Dirichlet, Neumann, and Robin boundary conditions. Through theoretical analysis and numerical studies, we show that the proposed solver is a uniform convergent scheme. We also numerically show that GPDM is more accurate compared to DM in terms of solving eigenvalue problems.

The paper will be organized as follows. In Section 2, we provide a short review of diffusion maps and its local kernel variants to approximate various types of linear second-order elliptic differential operators defined on smooth manifolds embedded in \mathbb{R}^n . We end the section illustrating the problem of DM near the boundary. In Section 3, we present the GPDM method which overcomes the issue near the boundary. We close this section with numerical examples to support the theoretical results. In Section 4, we discuss the application of GPDM for solving elliptic PDEs with various boundaries. In Section 5, we discuss the application of GPDM for solving eigenvalue problems corresponding to the elliptic PDEs. We close the paper with a summary and a list of open problems in Section 6. To improve the readability, we report the detailed proofs in four appendices.

2 Diffusion maps and its extension with local kernels

In this section, we provide a short review of the diffusion maps algorithm [14] as a method to approximate the Laplacian, a class of second-order, self-adjoint, positive-definite, differential operators that acts on functions defined on smooth compact Riemannian manifolds. In addition, we also review the variant of diffusion maps to approximate the second-order elliptic diffusion operator with a given diffusion coefficient [18] and the non-symmetric drifted diffusions via the local kernels [15].

Let M be a d -dimensional compact Riemannian manifold embedded in \mathbb{R}^n , possibly with boundary ∂M . Let $u \in C^3(M)$ and $h : [0, \infty) \rightarrow (0, \infty)$ be defined as $h(s) = \exp(-s)$. Let $\epsilon > 0$, for all $x \in M$ whose distance from the boundary is larger than ϵ^r , where $0 < r < 1/2$, the integral operator,

$$G_\epsilon u(x) := \epsilon^{-d/2} \int_M h\left(\frac{|x-y|^2}{4\epsilon}\right) u(y) dV(y) = \epsilon^{-d/2} \int_{M_{\epsilon,x}} h\left(\frac{|x-y|^2}{4\epsilon}\right) u(y) dV(y) + \mathcal{O}(\epsilon^2) \quad (1)$$

is effectively a local integral operator over the ϵ^r -ball around x , $M_{\epsilon,x} := \{y \in M, |y-x| < \epsilon^r\}$. In (1). The notation $|\cdot|$ denotes the standard Euclidean norm for vectors in \mathbb{R}^n . The key idea of the diffusion maps algorithm lies on the following asymptotic expansion. For any points $x \in M$ whose distance from the boundary is larger than ϵ^r , where $0 < r < 1/2$,

$$G_\epsilon u(x) = m_0 u(x) + \epsilon m_2 (\omega(x) u(x) + \Delta_g u(x)) + \mathcal{O}(\epsilon^2), \quad (2)$$

where m_0 and m_2 are constants that depend on h , ω depends also on the geometry of M , and Δ_g denotes the negative definite Laplace-Beltrami operator defined with respect to the Riemannian metric g inherited by M from \mathbb{R}^n . Based on this asymptotic expansion, one can approximate the Laplace-Beltrami operator as,

$$L_{1,\epsilon} u(x) := \frac{(G_\epsilon 1(x))^{-1} G_\epsilon u(x) - u(x)}{\epsilon} = \Delta_g u(x) + \mathcal{O}(\epsilon) := \mathcal{L}_1 u(x) + \mathcal{O}(\epsilon), \quad (3)$$

for all $x \in M$ whose distance from the boundary is larger than ϵ^r , where $0 < r < 1/2$. If one is given a strictly positive, smooth, diffusion coefficient $\kappa : M \rightarrow (0, \infty)$, one can also approximate the anisotropic diffusion operator,

$$L_{2,\epsilon} u(x) := \kappa(x) \frac{(G_\epsilon \sqrt{\kappa(x)})^{-1} G_\epsilon (\sqrt{\kappa(x)} u(x)) - u(x)}{\epsilon} = \text{div}_g(\kappa(x) \nabla_g u(x)) + \mathcal{O}(\epsilon) := \mathcal{L}_2 u(x) + \mathcal{O}(\epsilon), \quad (4)$$

where we have used the notations div_g and ∇_g for the divergence and gradient operators, respectively, defined with respect to the Riemannian metric g . One can also apply the equivalent diffusion operator using the symmetric version as reported in [18].

Beyond these two self-adjoint operators, one can also approximate the backward Kolmogorov operator,

$$\mathcal{L}_3 u := b \cdot \nabla_g u + \frac{1}{2} c^{ij} \nabla_i \nabla_j u, \quad (5)$$

where ∇_i is the covariant derivative in the i th direction, and $\nabla_i \nabla_j$ is the components of the Hessian operator. Here, the differential operators and the dot product are defined with respect to the Riemannian metric inherited by M from \mathbb{R}^n . The vector field $b : M \rightarrow \mathbb{R}^d$ is the drift and the symmetric positive definite diffusion tensor $c : M \rightarrow \mathbb{R}^d \times \mathbb{R}^d$ is a $d \times d$ diffusion matrix, where d is the dimension of manifold M .

The operator in (5) can be accessed by employing the integral operator in (1) with the following prototypical kernel [15]:

$$K(\epsilon, x, y) := h\left(\frac{|x + \epsilon B(x) - y|_{C(x)^{-1}}^2}{2\epsilon}\right) = \exp\left(-\frac{(x + \epsilon B(x) - y)^\top C(x)^{-1} (x + \epsilon B(x) - y)}{2\epsilon}\right) \quad (6)$$

where $B : M \rightarrow \mathbb{R}^n$ and $C : M \rightarrow \mathbb{R}^n \times \mathbb{R}^n$ are related to b and c , respectively, through an embedding function $\iota : M \rightarrow \mathbb{R}^n$ as follows:

$$B(x) = D\iota(x) b(x), \quad C(x)^{-1} = (D\iota(x) c(x) D\iota(x)^\top)^\dagger. \quad (7)$$

Here, the notation † denotes the pseudo-inverse and the Jacobian $D\iota(x)$ is a map that takes vectors in the tangent space $T_x M \cong \mathbb{R}^d$ (written in the basis $\{\partial_i\}$) to the ambient space $T\mathbb{R}^n \cong \mathbb{R}^n$ so that $D\iota(x)$ is an $n \times d$ matrix. Applying the integral operator in (1) with the prototypical kernel $K(\epsilon, x, y)$ on manifold without boundary, we obtain,

$$G_{K,\epsilon} u(x) := \epsilon^{-d/2} \int_M K(\epsilon, x, y) u(y) dV_y = m(x) u(x) + \epsilon (\omega(x) u(x) + m(x) \mathcal{L}_3 u(x)) + \mathcal{O}(\epsilon^2), \quad (8)$$

where $m(x) = (2\pi)^{d/2} \det(c(x))^{1/2}$ which can be approximated by $G_{K,\epsilon} 1(x) = m(x) + \mathcal{O}(\epsilon)$. Employing the same algebraic manipulation as in (3), we obtain

$$L_{3,\epsilon} u(x) := \frac{(G_{K,\epsilon} 1(x))^{-1} G_{K,\epsilon} u(x) - u(x)}{\epsilon} = \mathcal{L}_3 u(x) + \mathcal{O}(\epsilon). \quad (9)$$

We note that the evaluation of the prototypical kernel in (6) requires the knowledge of either the intrinsic representation b and c together with the embedding function ι or the ambient representation B and C in (7). If the embedding

function ι is unknown, we need to learn the structure of the manifold from the data points and restrict B and C to the tangent space $T_x M$.

In the local kernel method, the backward Kolmogorov operator \mathcal{L}_3 becomes the Laplace-Beltrami operator $\mathcal{L}_1 = \Delta_g$ when $B(x) = 0$ and $C(x) = 2I$, where I denotes the identity matrix. This can be found from the following fact,

$$\Delta_g u = g^{ij} \left(\frac{\partial^2 u}{\partial \theta^i \partial \theta^j} - \Gamma_{ij}^k \frac{\partial u}{\partial \theta^k} \right), \quad (10)$$

$$\mathcal{L}_3 u = b^k \frac{\partial u}{\partial \theta^k} + \frac{1}{2} c^{ij} \left(\frac{\partial^2 u}{\partial \theta^i \partial \theta^j} - \Gamma_{ij}^k \frac{\partial u}{\partial \theta^k} \right), \quad (11)$$

where $\{\theta^i\}$ form local coordinates, $g_{ij} = g\left(\frac{\partial}{\partial \theta^i}, \frac{\partial}{\partial \theta^j}\right)$ are the components of the Riemannian metric tensor, g^{ij} are the components of the inverse of the metric tensor ($g_{ij} g^{jk} = \delta_i^k$ with δ_i^k the Kronecker delta), and Γ_{ij}^k are the Christoffel symbols of the connection. One can observe from (7) that $B(x) = 0$ and $C(x) = 2I$ yields the components of intrinsic representation being $b^k = 0$ and $c^{ij} = 2g^{ij}$, where $g_{ij} = [Dt(x)^\top Dt(x)]_{ij}$.

Numerically, given points in ambient coordinate $\{x_i \in M\}_{i=1, \dots, N}$, one can approximate the integrals in $L_{1,\epsilon}$ or $L_{2,\epsilon}$ or $L_{3,\epsilon}$ via a Monte-Carlo average accounting for the sampling density of the data $x_i \sim q(x)$ that is not necessarily uniform. Since such discretizations have been reported for \mathcal{L}_1 and \mathcal{L}_3 (see [14, 15, 23]), here we only discuss the discrete estimator for \mathcal{L}_2 , which involves an importance sampling to debias the effect of the sampling density of the data. To compute $G_\epsilon \sqrt{\kappa(x)}$, we first construct an $N \times N$ matrix with entry $\mathbf{K}_{ij} = K(\epsilon, x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right)$. Then, the estimated unnormalized density evaluated at x_i can be estimated by the i th component of vector \mathbf{q} , that is, $q(x_i) \approx \mathbf{q}_i = \epsilon^{-d/2} N^{-1} \sum_{j=1}^N \mathbf{K}_{ij}$. Then,

$$\begin{aligned} \epsilon^{d/2} G_\epsilon \sqrt{\kappa(x_i)} &= \int_M h\left(\frac{|x_i - y|^2}{4\epsilon}\right) \sqrt{\kappa(y)} dV(y) \approx \frac{1}{N} \sum_{j=1}^N \mathbf{K}_{ij} \frac{\sqrt{\kappa(x_j)}}{\mathbf{q}_j}, \\ \epsilon^{d/2} G_\epsilon (u(x_i) \sqrt{\kappa(x_i)}) &= \int_M h\left(\frac{|x_i - y|^2}{4\epsilon}\right) u(y) \sqrt{\kappa(y)} dV(y) \approx \frac{1}{N} \sum_{j=1}^N \mathbf{K}_{ij} \frac{\sqrt{\kappa(x_j)} u(x_j)}{\mathbf{q}_j}. \end{aligned} \quad (12)$$

Defining \mathbf{W} as an $N \times N$ matrix with entries $\mathbf{W}_{ij} = \mathbf{K}_{ij} \frac{\sqrt{\kappa(x_j)}}{\mathbf{q}_j}$, let \mathbf{D} be a diagonal matrix with diagonal entries $\mathbf{D}_{ii} = \sum_{j=1}^N \mathbf{W}_{ij}$ and \mathbf{S} be a diagonal matrix with diagonal entries $\mathbf{S}_{ii} = \kappa(x_i)$, then the discrete estimator for \mathcal{L}_2 is given by,

$$L_{2,\epsilon} \approx \mathbf{L}_2 = \frac{1}{\epsilon} \mathbf{S} (\mathbf{D}^{-1} \mathbf{W} - \mathbf{I}). \quad (13)$$

We should point out that the discrete error estimate for the Monte-Carlo average [24, 25, 17] for the pointwise convergence of $L_{j,\epsilon} \rightarrow \mathcal{L}_j$ (for each $j = 1, 2, 3$) only holds in probability (see Lemma C.1 in Appendix C). If the goal is to estimate the range of these differential operators, then one can expect accurate estimation using regularly spaced (or well sampled) discrete points. However, if the goal is to solve the eigenvalue problem associated to these operators, then one can still obtain reasonably accurate estimates even with random data. We suspect that this is due to a much faster decaying rate of the spectral error bound as a function of sample size N (see e.g., [26, 27]). For the numerical demonstration in this paper, we will only consider reasonably well-sampled data for more accurate estimation of the diffusion operators as well as their inversion and eigenvalue problems.

Now, let us illustrate the problem near the boundary of the asymptotic approximation of the weighted Laplacian in (4) with a simple example.

Example 2.1. Here, we compare the estimation of the differential operator \mathcal{L}_2 on a one-dimensional ellipse, defined with the Riemannian metric,

$$g(\theta) = \sin^2 \theta + a^2 \cos^2 \theta, \quad \text{for } 0 \leq \theta \leq \pi, \quad (14)$$

where $a = 3 > 1$. The diffusion coefficient in the weighted Laplacian (4) is $\kappa = 1.1 + \sin \theta$. In local coordinates, the diffusion operator acting on function u is given as,

$$\mathcal{L}_2 u := \operatorname{div}(\kappa \nabla u) = \frac{1}{\sqrt{|g|}} \frac{\partial}{\partial \theta} \left(\sqrt{|g|} \kappa g^{-1} \frac{\partial u}{\partial \theta} \right). \quad (15)$$

In Fig. 1, we plot the explicit equation in (15) acting on a test function $u = \cos(3\theta/2 - \pi/4)$, defined on a semi-ellipse with $a = 3$ as a function of intrinsic coordinate $\theta \in [0, \pi]$. The discrete estimator, \mathbf{L}_2 , of \mathcal{L}_2 is constructed using $N = 400$ data points distributed in equal angle. Notice the agreement between the DM estimate and the truth except near the boundaries. In the same figure, we also show the improved estimate using the Ghost Point Diffusion Maps (GPDM) near the boundaries that we will explain in the next section.

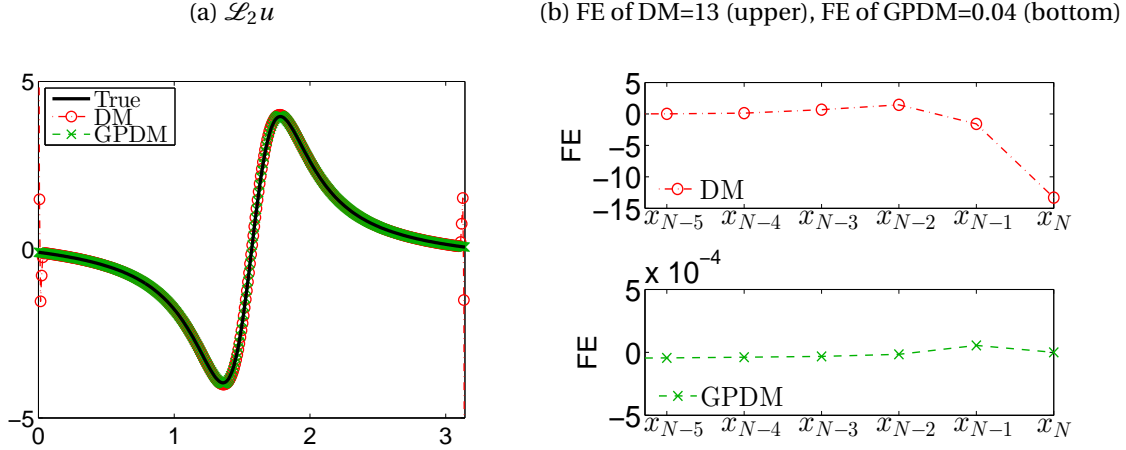


Figure 1: (Color online) A numerical approximation of $\mathcal{L}_2 u$ on a semi-ellipse example where \mathcal{L}_2 is the weighted Laplacian in (4). (a) Comparison of $\mathcal{L}_2 u$ and DM and GPDM estimates. (b) Absolute error of DM (upper panel) and GPDM (bottom panel) near the boundary. One can see that the Forward Error (FE), defined with the uniform norm, using the standard DM is relatively large up to 13 near the boundary [red circles in upper panel of (b)]. However, by applying the GPDM, the FE reduces to 0.04 [green crosses in bottom panel of (b)]. Note that for GPDM, the FE does not reach its maximum near the boundary but in the interior of the domain instead.

3 Ghost Point Diffusion Maps for 1D and 2D manifolds

In this section, we introduce an improved method, the ghost point diffusion maps, for approximating differential operators in (3), (4), (5) defined on one and two-dimensional manifolds with boundaries. To facilitate the discussion, we use the conventional notations ∂M and M° to denote the boundary and interior sets of manifold M , respectively, that satisfy $M = M^\circ \cup \partial M$ and $M^\circ \cap \partial M = \emptyset$. We assume that M is a compact domain such that the closed subset ∂M is also a compact set. For two-dimensional problems, we also assume that the boundary ∂M is a smooth curve.

The basic idea is to follow the classical ghost point method [22] for solving the Neumann boundary condition with the finite-difference method. In our configuration, we supplement ghost points near the boundary such that the diffusion maps asymptotic expansion for the estimation of the diffusion operator is valid even for points near the boundary, where the second-order differential operator is approximated with an appropriate affine linear operator. In this work, we note that all the boundary points have been labeled (or known) before our estimation of operators. For problems with unknown boundary points, one can refer to [21] for a method to estimate the distance to boundary points.

We now describe the proposed algorithm, which we refer to as the Ghost Point Diffusion Maps (GPDM). Particularly, the construction of the GPDM requires the following technical tools. In Section 3.1, we estimate the exterior normal direction \mathbf{v} to the boundary. In Section 3.2, we estimate the normal derivative $\partial_{\mathbf{v}} u$ for $x \in \partial M$, which will be used in the estimation of the boundary condition as well as the extrapolation function to obtain the function value at the ghost points. In Section 3.3, we describe the construction of the ghost points along the direction \mathbf{v} from boundary points. In Section 3.4, we discuss how to extrapolate the unknown function values at the ghost points. Here, we introduce a set of artificial boundary conditions on the ghost points, which ensures the consistency of the affine estimator of \mathcal{L}_j in the limit of $\epsilon \rightarrow 0$ and $N \rightarrow \infty$, as reported in Section 3.5. Finally, we give numerical examples to validate the theory in Section 3.6.

3.1 Estimation of the exterior normal direction at the boundaries

In this section, we provide a numerical method to estimate the exterior normal direction using the point cloud data, assuming that the boundary points are known. We split the discussion for 1D and 2D manifolds.

3.1.1 One-dimensional manifolds

Suppose that $\gamma: \mathbb{R} \rightarrow M$ is a parameterization of the one-dimensional manifold M . The unit normal direction \mathbf{v} to the boundary is given by the unit tangent vector $\mathbf{t} = \frac{\gamma'(s)}{|\gamma'(s)|}$, where s is evaluated at the parameter corresponding to the boundary points. Here, the sign of \mathbf{t} can be either positive or negative depending on the orientation of the

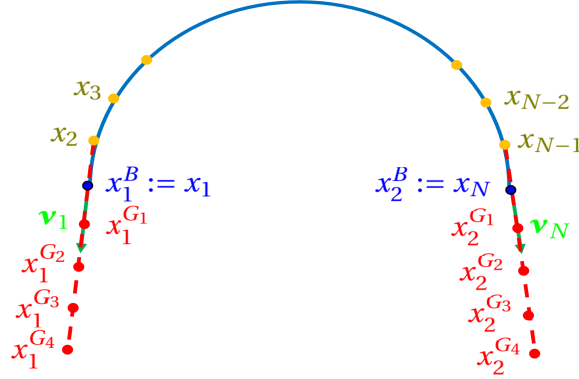


Figure 2: (Color online) Sketch of a specification of ghost points $\{x_j^{Gk}\}$ starting from the boundary, along the estimated exterior normal direction \mathbf{v} on a 1D manifold.

curve. When the parameterization γ is unknown, we can use finite-difference method to estimate this normal direction \mathbf{v} to the boundary. For example, Fig. 2 shows the dataset $\{x_i\}_{i=1,\dots,N}$ on the 1D semi-ellipse with x_1 and x_N as the boundary points. The dataset are well-ordered along the curve (x_i and x_{i+1} are neighboring points for $i = 1, \dots, N-1$) so that the unit tangent vector along the semi-ellipse can be approximated by the vector

$$\tilde{\mathbf{t}} = \frac{x_{i+1} - x_i}{|x_{i+1} - x_i|}, \quad (16)$$

where $|\cdot|$ denotes the Euclidean \mathbb{R}^n -norm. In particular, the outward normal direction \mathbf{v} at two boundary points x_1 and x_N can be approximated by

$$\tilde{\mathbf{v}}_1 = \frac{x_1 - x_2}{|x_1 - x_2|} \text{ and } \tilde{\mathbf{v}}_N = \frac{x_N - x_{N-1}}{|x_N - x_{N-1}|}, \quad (17)$$

respectively, as shown in Fig. 2. To provide an error estimate, we let $\gamma(s)$ be the geodesic curve. Without loss of generality, we consider only the geodesic parameterization between any two consecutive points such that $\gamma(0) = x_i$ and $\gamma(s) = x_{i+1}$ and the arc-length parameterization $s = \int_0^s |\gamma'(t)| dt$, such that $|\gamma'(t)| = 1$ for all $t \in [0, s]$. In the proposition below, we will compute the error of approximating the unit tangent vector $\mathbf{t} = \gamma'(0)$ at $x_i = \gamma(0)$ with the vector $\tilde{\mathbf{t}}$ in (16). Then, the error estimate for the normal direction \mathbf{v} to the boundary follows from the proposition directly for the one-dimensional curves.

Proposition 3.1. *Let $\gamma(s)$ be a geodesic curve parameterized with the arc-length s , connecting discrete points $\{x_i\}_{i=1,\dots,N} \in M$ along the one-dimensional manifold M in \mathbb{R}^n . Let x_i, x_{i+1} denote two arbitrary neighboring points on the curve such that $|x_i - x_{i+1}| = \mathcal{O}(h)$, where $|\cdot|$ denotes the Euclidean \mathbb{R}^n -norm. Then, the unit tangent vector $\mathbf{t} = \gamma'(0)$ at point $x_i = \gamma(0)$ can be estimated by $\tilde{\mathbf{t}}$ in (16) with error $|\mathbf{t} - \tilde{\mathbf{t}}| = \mathcal{O}(h)$, where the constant in the error bound depends on the local curvature $\omega = |\gamma''(0)|$ of the curve at $x_i = \gamma(0)$.*

Proof. For small s , applying Taylor's expansion on γ , we get

$$\gamma(s) = \gamma(0) + s\gamma'(0) + \frac{s^2}{2}\gamma''(0) + \frac{s^3}{6}\gamma'''(0) + \mathcal{O}(s^4).$$

Since $\gamma''(s) \perp T_x M$ for any $x \in M$ (by geodesic curve), we obtain:

$$|\gamma(s) - \gamma(0)|^2 = s^2 + s^4 \left(\frac{1}{4} |\gamma''(0)|^2 + \frac{1}{3} \langle \gamma'(0), \gamma'''(0) \rangle \right) + \mathcal{O}(s^5).$$

This also means that,

$$\begin{aligned} |\gamma(s) - \gamma(0)| &= s + s^3 \left(\frac{1}{8} |\gamma''(0)|^2 + \frac{1}{6} \langle \gamma'(0), \gamma'''(0) \rangle \right) + \mathcal{O}(s^4), \\ |\gamma(s) - \gamma(0)|^{-1} &= s^{-1} \left(1 - s^2 \left(\frac{1}{8} |\gamma''(0)|^2 + \frac{1}{6} \langle \gamma'(0), \gamma'''(0) \rangle \right) + \mathcal{O}(s^3) \right). \end{aligned} \quad (18)$$

By the definitions of \mathbf{t} and $\tilde{\mathbf{t}}$ and after some algebra, we have

$$|\mathbf{t} - \tilde{\mathbf{t}}| = \left| \gamma'(0) - \frac{\gamma(s) - \gamma(0)}{|\gamma(s) - \gamma(0)|} \right| = \frac{s}{2} |\gamma''(0)| + \mathcal{O}(s^2).$$

Since $s = \mathcal{O}(h)$, it is clear that $|\mathbf{t} - \tilde{\mathbf{t}}| = \mathcal{O}(h)$ with constant that depends on the curvature $\omega = |\gamma''(0)|$.

□

3.1.2 Two-dimensional manifolds

We now estimate the unit exterior normal direction \mathbf{v} to the boundary ∂M of 2D manifolds, embedded in \mathbb{R}^n . If the parametrization $x = x(\theta_1, \theta_2)$ is known, we can obtain the tangent space of M spanned by two linearly independent vectors $\partial x / \partial \theta_1$ and $\partial x / \partial \theta_2$. At the boundary, we can obtain the tangent vector \mathbf{t} along the 1D boundary ∂M in the same manner as discussed in the previous section. Then, the normal direction \mathbf{v} to the boundary can be calculated by subtracting the tangent vector \mathbf{t} from the tangent space spanned by $\partial x / \partial \theta_1$ and $\partial x / \partial \theta_2$ using Gram-Schmidt process. The sign of \mathbf{v} can be finally determined by the orientation of the manifold M . If the parametrization is unknown, we need to numerically estimate the two tangent vectors that span the surface M and the tangent direction \mathbf{t} along the boundary ∂M .

First, we can estimate the two tangent vectors \mathbf{t}_1 and \mathbf{t}_2 that span the surface M using a kernel-based nonlinear regression method as introduced in Corollary 3.2. of [28]. Here, we give a quick review of the algorithm to estimate the two tangent vectors at $x \in M \subseteq \mathbb{R}^n$. Define \mathbf{X} to be the $n \times K$ matrix with columns $\mathbf{X}_j = D(x)^{-1/2} \exp(-|x - x_j|^2 / 4\epsilon) (x_j - x)$ where $D(x) = \sum_{i=1}^K \exp(-|x - x_i|^2 / 2\epsilon)$ with x_i being K nearest neighbors of x . Then, the singular value of matrix \mathbf{X} will either be order- $\sqrt{\epsilon}$ if the associated singular vector is parallel to the tangent space, or order- ϵ if the singular vector is orthogonal to the tangent space. For the corresponding two singular vectors as the estimator of the two tangent vectors, the error estimate of the two tangent directions is on order- $\sqrt{\epsilon}$, where ϵ characterizes the bandwidth of kernels (see Appendix A in [28] for detailed discussion).

Second, the tangent direction \mathbf{t} along the boundary can be estimated using the finite-difference method as in (16) for the 1D manifold ∂M . The error estimate of \mathbf{t} is order- h where h characterizes the distance of neighboring points, $|x_i - x_{i+1}| = \mathcal{O}(h)$. Finally, the normal direction \mathbf{v} to the boundary can be approximated by subtracting the orthogonal projection of \mathbf{t}_1 or \mathbf{t}_2 onto \mathbf{t} from the tangent vector \mathbf{t}_1 or \mathbf{t}_2 using the Gram-Schmidt process. The error estimate for the normal direction \mathbf{v} is thereafter $\mathcal{O}(h, \sqrt{\epsilon})$, that is, $|\mathbf{v} - \tilde{\mathbf{v}}| = \mathcal{O}(h, \sqrt{\epsilon})$.

In particular, if the ambient space has dimension $n = 3$, one can first numerically estimate the normal direction \mathbf{n} to the surface M and the tangent direction \mathbf{t} along the boundary ∂M . Then, the normal direction \mathbf{v} to the boundary can be approximated from the cross product between the estimated \mathbf{n} and \mathbf{t} , that is, $\tilde{\mathbf{v}} = \tilde{\mathbf{n}} \times \tilde{\mathbf{t}}$, where the tilde denotes the estimated vector.

Figure 3 displays an example on a semi-torus manifold for comparisons of the numerical estimates and analytic results, $\mathbf{n}, \mathbf{t}, \mathbf{v}$. In this example, the semi-torus is defined with the standard embedding function:

$$x := \iota(\theta, \phi) = \begin{pmatrix} (a + \cos \theta) \cos \phi \\ (a + \cos \theta) \sin \phi \\ \sin \theta \end{pmatrix}, \quad \text{for } \begin{matrix} 0 \leq \theta \leq 2\pi, \\ 0 \leq \phi \leq \pi, \\ a = 2, \end{matrix} \quad (19)$$

where θ, ϕ are the intrinsic coordinates and a is the radius of the semi-torus. The induced Riemannian metric is given by,

$$g_{x^{-1}(\theta, \phi)}(u, v) = u^\top \begin{pmatrix} 1 & 0 \\ 0 & \sin^2 \theta \end{pmatrix} v, \quad \forall u, v \in T_{x^{-1}(\theta, \phi)} M. \quad (20)$$

It can be seen from Fig. 3 that the estimations of the three directions \mathbf{n}, \mathbf{t} , and \mathbf{v} are in good agreement with the analytic results.

3.2 Estimation of the normal derivatives on the boundaries

In this section, we discuss a numerical method to estimate normal derivatives on the boundaries using the point cloud data. For 1D manifolds, the exterior normal direction \mathbf{v} is exactly along the tangent direction of the curve. Then, the normal derivatives $\partial_{\mathbf{v}} u$ at two boundary points can be approximated using a finite-difference method. For the semi-ellipse example in Fig. 2, the normal derivatives at x_1 and x_N can be approximated as,

$$\frac{\partial u}{\partial \mathbf{v}}(x_1) \approx \frac{\Delta u}{\Delta \tilde{\mathbf{v}}}(x_1) := \frac{u(x_1) - u(x_2)}{|x_1 - x_2|}, \quad \frac{\partial u}{\partial \mathbf{v}}(x_N) \approx \frac{\Delta u}{\Delta \tilde{\mathbf{v}}}(x_N) := \frac{u(x_N) - u(x_{N-1})}{|x_N - x_{N-1}|}. \quad (21)$$

For 2D manifolds in \mathbb{R}^n , we can estimate the normal derivatives following the steps in the following algorithm.

Algorithm 3.2. Assume that \mathbf{v} is the exterior normal direction to the boundary ∂M and $\tilde{\mathbf{v}}$ is its numerical estimate as defined in Section 3.1.2 at a boundary point $x^B \in \partial M$ as shown in the sketch in Fig. 4(a). Then, the normal derivative $\partial_{\mathbf{v}} u$ at x^B is estimated as follows:

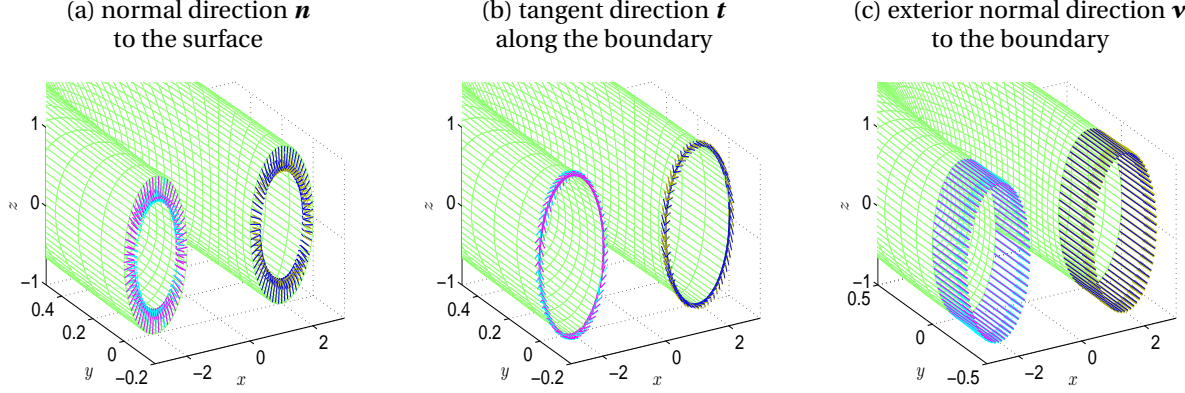


Figure 3: (Color online) Numerical estimation of (a) the normal direction \mathbf{n} to the surface, (b) the tangent direction \mathbf{t} along the boundary, (c) the exterior normal direction \mathbf{v} to the boundary when the parametrization of the semi-torus M is unknown. Numerical estimates are depicted with yellow and cyan arrows, while the analytic results are depicted with blue and magenta arrows.

1. Find the "left" nearest neighbor x^L and "right" nearest neighbor x^R for the boundary point $x^B \in \partial M$. Then, one can compute the normalized vectors,

$$\tilde{\mathbf{v}}^L := \frac{x^L - x^B}{|x^L - x^B|} \text{ and } \tilde{\mathbf{v}}^R := \frac{x^R - x^B}{|x^R - x^B|}.$$

Here, x^L is the nearest point to x^B in the region such that the angle between $\tilde{\mathbf{v}}^L$ and $-\tilde{\mathbf{v}}$ satisfying $\Theta(\tilde{\mathbf{v}}^L, -\tilde{\mathbf{v}}) < \Theta_0$ (in our implementation, $\Theta_0 = \pi/4$). Similarly, idea applies for x^R . Moreover, the "left" and "right" can be numerically distinguished by the negative inner product $\langle \tilde{\mathbf{w}}^L, \tilde{\mathbf{w}}^R \rangle < 0$ where $\tilde{\mathbf{w}}^L$ and $\tilde{\mathbf{w}}^R$ are components orthogonal to $-\tilde{\mathbf{v}}$, that is, $\tilde{\mathbf{w}}^L = \tilde{\mathbf{v}}^L - (\tilde{\mathbf{v}}^L \cdot \tilde{\mathbf{v}})(\tilde{\mathbf{v}})$ and $\tilde{\mathbf{w}}^R = \tilde{\mathbf{v}}^R - (\tilde{\mathbf{v}}^R \cdot \tilde{\mathbf{v}})(\tilde{\mathbf{v}})$.

2. Write $-\tilde{\mathbf{v}}$ as a linear combination of $\tilde{\mathbf{v}}^L$ and $\tilde{\mathbf{v}}^R$ using the linear regression,

$$-\tilde{\mathbf{v}} = \tilde{a}^L \tilde{\mathbf{v}}^L + \tilde{a}^R \tilde{\mathbf{v}}^R, \quad (22)$$

where \tilde{a}^L and \tilde{a}^R are the regression coefficients.

3. Estimate the normal derivative $-\partial_{\tilde{\mathbf{v}}} u$ numerically using the difference method,

$$\frac{\partial u}{\partial(-\tilde{\mathbf{v}})}(x^B) \approx \frac{\Delta u}{\Delta(-\tilde{\mathbf{v}})}(x^B) := \left[\tilde{a}^L \frac{\Delta u}{\Delta \tilde{\mathbf{v}}^L} + \tilde{a}^R \frac{\Delta u}{\Delta \tilde{\mathbf{v}}^R} \right](x^B) := \tilde{a}^L \frac{u(x^L) - u(x^B)}{|x^L - x^B|} + \tilde{a}^R \frac{u(x^R) - u(x^B)}{|x^R - x^B|}, \quad (23)$$

where we have used Eq. (22) and the fact that $\tilde{\mathbf{v}}$, $\tilde{\mathbf{v}}^L$, and $\tilde{\mathbf{v}}^R$ are all unit vectors. Then, the normal derivative $\partial_{\tilde{\mathbf{v}}} u$ term in the boundary condition (32) in the following section can be numerically estimated using Eq. (23) for all points on the boundary.

Next, we provide the error rate for estimating the directional derivative $\partial_{\tilde{\mathbf{v}}} u$ with Eq. (23) at the boundary points.

Proposition 3.3. Let $u \in C^3(M)$ be a smooth function on a 2D manifold M with 1D boundary ∂M . Let $\{x_i\}_{i=1,\dots,N} \in M$ be a set of data points, among which some labeled points lie on the boundary ∂M . Let $x^B \in \partial M$ be a boundary point on the 1D smooth ∂M and $\tilde{\mathbf{v}}$ be the unit exterior normal direction to the boundary ∂M at x^B . Let x^L and $x^R \in \{x_i\}_{i=1,\dots,N}$ be the "left" and "right" nearest neighbors, respectively, for the boundary point x^B . Then, the normal derivative $\partial_{\tilde{\mathbf{v}}} u$ at x_B estimated by Eq. (23) in Algorithm 3.2 has an error rate of

$$\left| \frac{\partial u}{\partial \tilde{\mathbf{v}}}(x^B) - \frac{\Delta u}{\Delta \tilde{\mathbf{v}}}(x^B) \right| = \mathcal{O}(h, \sqrt{\epsilon}),$$

where h characterizes the distance of the neighboring points and ϵ characterizes the bandwidth of the kernel. The constant depends on the local curvature and the norm of the second-order derivative of u (that is, $|\nabla_i \nabla_j u(x^B)|$ with $\nabla_i \nabla_j$ being the Hessian operator).

Proof. See Appendix A. □

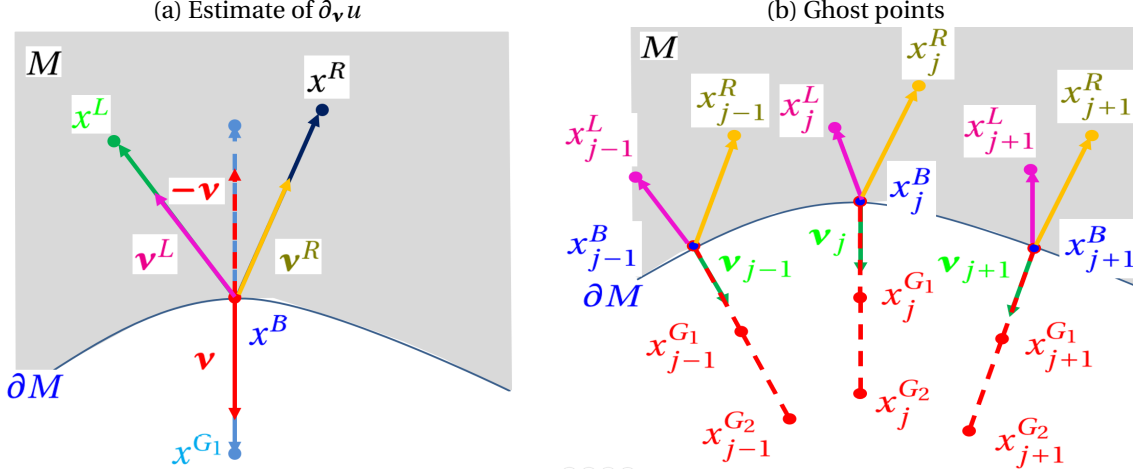


Figure 4: (Color online) Sketch (a) for computing $\partial_v u$ on 2D manifold, and (b) for generating ghost points and setting up the artificial boundary conditions on 2D manifold.

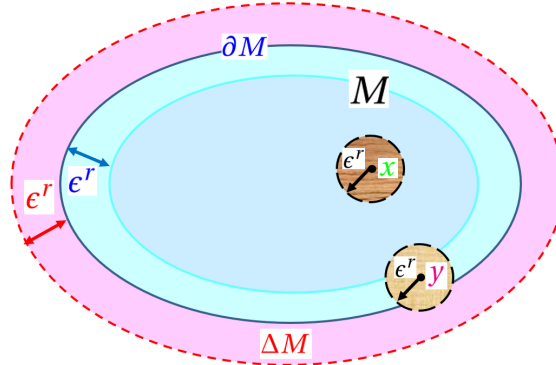


Figure 5: (Color online) Sketch of the extended manifold $M \cup \Delta M$.

From the proof of this proposition, we should point out that the error rate for the one-dimensional approximation in Eq. (21) is also of order- h . Particularly, one can obtain this error bound as a direct application to the error rate in (57), where x^R and x^B can be replaced by the adjacent and the boundary points, respectively, in one dimensional case. Another remark is that this result can be generalized to any d -dimensional manifold as long as the normal direction $\hat{\mathbf{v}}$ has been estimated numerically.

3.3 Ghost points

It is well known that the diffusion operators defined in (3)-(5) cannot be approximated accurately near the boundary of manifold using the standard diffusion maps algorithm. The reason is that the asymptotic expansion (2) is valid only for points $x \in M$ whose distance from the boundary ∂M is larger than ϵ^r , for $0 < r < 1/2$. For points $y \in M$ whose distance from ∂M is smaller than ϵ^r , an order- $\sqrt{\epsilon}$ term appears in the asymptotic expansion (2); geometrically, the local integral is inaccessible if there are no data beyond M (see Fig. 5).

To address this issue, our basic idea here is to supplement the original data points on M with a set of ghost points. Since the integral in the diffusion maps asymptotic expansion is effectively a local integral over a ball of radius ϵ^r as discussed in (1), we will devise a scheme to specify these new points such that one can approximate the local integral over the ball of radius ϵ^r even when the integral operator is evaluated at points in M whose distances are less than ϵ^r from the closest point on the boundary, ∂M (e.g., y in Fig. 5). To facilitate the analysis below, let us define ΔM as a smooth manifold where all ghost points lie on such that $\Delta M \cap M$ is smooth and $\Delta M \cap M = \partial M$. We should point out that the existence of ΔM follows from the classical interpolation theory for 1D case or the sufficient condition described in [29] for the 2D case. Since the proposed method involves an implementation of diffusion maps algorithm on the smooth extended manifolds $M \cup \Delta M$, we referred to the new algorithm as the ghost point diffusion maps (GPDM). If the embedding is known, we can specify ΔM by extending points on the

intrinsic coordinates and specify the points on $M \cup \Delta M$ using the embedding function. Now we will describe an algorithm for specifying the ghost points and describe the detail assumption on ΔM when the embedding function is unknown for 1D and 2D manifolds.

For 1D manifold (e.g., as shown in Fig. 2), the data $\{x_i\}_{i=1}^N$ are well aligned along the manifold M , and x_1 and x_N are the two boundary points on ∂M . For one side at x_1 , the ghost points $x_1^{G_1}, x_1^{G_2}, x_1^{G_3}, x_1^{G_4}$ are added along the exterior normal direction \mathbf{v}_1 that is numerically approximated by $\tilde{\mathbf{v}}_1$ in (17) with the equal spacing $|x_1 - x_2|$. For the other side at x_N , ghost points can be added in a similar way. In this case, the boundary manifold, ΔM is smooth such that it can be parameterized by two smooth curves: One of them interpolates $x_1^{G_1}, x_1^{G_2}, x_1^{G_3}, x_1^{G_4}$ such that $\gamma_{x_1}(0) = x_1$, $\gamma_{x_1}(s_1) = x_1^{G_1}, \dots, \gamma_{x_1}(s_4) = x_1^{G_4}$, and $\gamma'_{x_1}(0) = \mathbf{v}_1$, where $s_j < \epsilon^r$ are the geodesic distances between the adjacent points. The other one interpolates x_N and $x_2^{G_1}, x_2^{G_2}, x_2^{G_3}, x_2^{G_4}$ in the same manner. In practice, at most 3–7 ghost points are added along the exterior normal direction to each boundary point in ∂M .

For 2D manifolds, we add the ghost points along the exterior normal direction to the boundary ∂M . For example, as shown in Fig. 4(b), we denote the boundary points on ∂M by $\{x_j^B\}_{j=1,\dots,J}$ and the corresponding exterior normal vectors by $\{\mathbf{v}_j\}_{j=1,\dots,J}$. Along each normal direction \mathbf{v}_j at $x_j^B \in \partial M$, one can add ghost points $x_j^{G_1}, x_j^{G_2}, \dots, x_j^{G_K}$ one by one with the equal spacing \tilde{d}_j . Numerically, \mathbf{v}_j will be approximated by $\tilde{\mathbf{v}}_j$ as discussed in Section 3.1.2. Here, \tilde{d}_j is estimated by the mean distance from x_j^B to its P (at most 10 in our numerical examples) nearest neighbors. Let $x_{j,p}^B \in M$ denotes the p th nearest neighbor of x_j^B for $p = 1, \dots, P$. Since the distance to the nearest neighbor is a density estimator [30], that is, $|x_j^B - x_{j,p}^B| \propto q(x_j^B)^{-1/d}$, where q denotes the sampling density and d denotes the dimension of the manifold M , then the distance

$$\tilde{d}_j = \frac{1}{P} \sum_{p=1}^P |x_j^B - x_{j,p}^B| = \mathcal{O}\left(q(x_j^B)^{-\frac{1}{d}}\right), \quad (24)$$

for $d = 2$ dimensional manifolds. Together with these ghost points, we assume that

Assumption 3.4. *There exists a smooth surface ΔM that interpolates the ghost points $\{x_j^{G_k}\}$ and is connected to M at the boundary ∂M such that along the boundary ∂M , the outward normal vectors \mathbf{v} of M is consistent with the inward normal vector of ΔM . The second condition is to guarantee the smoothness of $M \cup \Delta M$.*

Recall that since the whole point of augmenting ΔM is such that the Euclidean distance between any point $x \in M_o$ and $\partial(M \cup \Delta M)$ is at least ϵ^r , the Assumption 3.4 guarantees the validity of the asymptotic expansion in (2) for all points on M^o (including points that are close to the boundary ∂M). Therefore, the estimators $L_{j,\epsilon}$ in (3), (4), and (9) are valid on all $x \in M^o$, with additional errors induced by extrapolating u and κ on the ghost points.

3.4 Extrapolation of functions on the ghost points

We now address the extrapolation problem on the ghost points. In particular, we need to extrapolate the function κ in \mathcal{L}_2 , as well as the solution u on the ghost points. Popular extrapolation techniques include the linear and quadratic extrapolation methods, the level set method, the ghost fluid method [31]. One idea is to extend the function of interest with a set of artificial boundary conditions, imposed on the ghost points. This leads us to the problem of specifying the boundary conditions on the ghost points. In particular, we will consider a discrete analog of matching the first-order or second-order derivatives of the functions evaluated at the ghost points as the extrapolation condition, which mimics the cubic spline condition as proposed in [32]. For the extrapolation of u , we also include a condition that mimics the finite-difference solution of Neumann (or Robin) boundary value problems with ghost points.

3.4.1 Extrapolation of κ at the ghost points

In this section, we discuss the extrapolation method for $\kappa \in C^2(M \cup \Delta M)$. To this end, we define a function $\tilde{\kappa}_\epsilon$ such that,

$$\tilde{\kappa}_\epsilon(x) = \begin{cases} \kappa(x_i), & x = x_i \in M^o, \\ \kappa(x_j^B), & x = x_j^B \in \partial M, \\ \tilde{\kappa}_{\epsilon,j}^{G_k}, & x = x_j^{G_k} \in \Delta M. \end{cases} \quad (25)$$

Here, the discrete extrapolation problem is to extend the function κ (identified by the function values on x_i, x_j^B) to find the function values $\tilde{\kappa}_{\epsilon,j}^{G_k}$ that satisfy the following algebraic equations,

$$\begin{aligned}\tilde{\kappa}_{\epsilon,j}^{G_1} - \kappa(x_j^B) - \tilde{d}_j \frac{\Delta \kappa}{\Delta \mathbf{v}_j}(x_j^B) &= 0, \\ \tilde{\kappa}_{\epsilon,j}^{G_2} - 2\tilde{\kappa}_{\epsilon,j}^{G_1} + \tilde{\kappa}_{\epsilon,j}^B &= 0, \\ \tilde{\kappa}_{\epsilon,j}^{G_k} - 2\tilde{\kappa}_{\epsilon,j}^{G_{k-1}} + \tilde{\kappa}_{\epsilon,j}^{G_{k-2}} &= 0, \quad k = 3, \dots, K,\end{aligned}\tag{26}$$

for $j = 1, \dots, J$. The first equation is a discrete analog of matching the directional derivative along \mathbf{v} at the boundary ∂M , where the operator $\frac{\Delta \kappa}{\Delta \mathbf{v}}$ is defined in Eq. (23). These equations are analogous to the discrete approximation of the second-derivatives in the natural spline. Numerically, (26) consists of J independent systems of K linear equations, each of which has a unique solution. For this extrapolation formula, we have

Proposition 3.5. (Extrapolation error rate for κ) Let $\kappa \in C^2(M \cup \Delta M)$, where ΔM satisfies the Assumption 3.4 and the Euclidean distance between any point in $x \in M$ and the extended boundary $\partial(M \cup \Delta M)$ is at least ϵ^r , where $0 < r < 1/2$ and suppose that $h = \mathcal{O}(\epsilon^r)$. Let $\tilde{\kappa}_\epsilon$ be defined as in (25) and (26). For any fixed $j = 1, \dots, J$, the error rate $|\kappa(x_j^{G_k}) - \tilde{\kappa}_{\epsilon,j}^{G_k}| = \mathcal{O}(h^2) = \mathcal{O}(\epsilon^{2r})$.

Proof. See Appendix B. □

While the same extrapolation formula in (26) can be used for estimating the function value u at the ghost points, it is not sufficient to achieve the consistency of the estimator. Thus, we consider a higher-order extrapolation method as discussed next.

3.4.2 Extrapolation of u at the ghost points

Let $u \in C^3(M \cup \Delta M)$, given the function values $u_i := u(x_i)$ for $x_i \in M$, our goal is to extrapolate u on the set of ghost points, $\{x_j^{G_k} \in \Delta M\}_{j=1, \dots, J}^{k=1, \dots, K}$, where each index j corresponds to the extension along the normal vector \mathbf{v}_j at the boundary point $x_j^B \in \partial M$. For each j , the Euclidean distance between two consecutive points, $x_j^{G_k}$ and $x_j^{G_{k+1}}$, is simply the distance between consecutive points at the boundary in the 1D case or the average distance \tilde{d}_j in (24) for the 2D case. With these notations, we define a function \tilde{u}_ϵ that satisfies:

$$\tilde{u}_\epsilon(x) = \begin{cases} u(x_i), & x = x_i \in M^o \\ u(x_j^B), & x = x_j^B \in \partial M, \\ \tilde{u}_{\epsilon,j}^{G_k}, & x = x_j^{G_k} \in \Delta M. \end{cases}\tag{27}$$

Here, the discrete extrapolation problem is to extend the function u (identified by the function values on x_i, x_j^B) to estimate the function values $\tilde{u}_{\epsilon,j}^{G_k}$ that satisfy the following algebraic equations,

$$\begin{aligned}(\mathbf{L}\tilde{u}_\epsilon)_{B_j} &= f(x_j^B), \\ \tilde{u}_{\epsilon,j}^{G_2} - 2\tilde{u}_{\epsilon,j}^{G_1} + u(x_j^B) &= \tilde{u}_{\epsilon,j}^{G_1} - u(x_j^B) - \tilde{d}_j \frac{\Delta u}{\Delta \mathbf{v}_j}(x_j^B), \\ \tilde{u}_{\epsilon,j}^{G_3} - 2\tilde{u}_{\epsilon,j}^{G_2} + \tilde{u}_{\epsilon,j}^{G_1} &= \tilde{u}_{\epsilon,j}^{G_2} - 2\tilde{u}_{\epsilon,j}^{G_1} + u(x_j^B), \\ \tilde{u}_{\epsilon,j}^{G_k} - 2\tilde{u}_{\epsilon,j}^{G_{k-1}} + \tilde{u}_{\epsilon,j}^{G_{k-2}} &= \tilde{u}_{\epsilon,j}^{G_{k-1}} - 2\tilde{u}_{\epsilon,j}^{G_{k-2}} + \tilde{u}_{\epsilon,j}^{G_{k-3}}, \quad k = 4, \dots, K,\end{aligned}\tag{28}$$

for $j = 1, \dots, J$. We have also defined $\tilde{u}_\epsilon = (u_\epsilon(x_1), \dots, u_\epsilon(x_N), \tilde{u}_{\epsilon,1}^{G_1}, \dots, \tilde{u}_{\epsilon,1}^{G_K}, \dots, \tilde{u}_{\epsilon,J}^{G_1}, \dots, \tilde{u}_{\epsilon,J}^{G_K}) \in \mathbb{R}^{\tilde{N}}$, where $\tilde{N} = N + JK$. The \mathbf{L} in first equation in (28) is the diffusion maps approximation to one of the diffusion operators in (3)-(5), dropping the sub-index to simplify the notation. Here, we have used the subscript- B_j to denote the component corresponding to the boundary point x_j^B . This condition is motivated by the classical finite-difference approach for solving Neumann (or Robin) problems with a given function value f_{B_j} , which imposes the operator estimation to be consistent at the boundary. The last three equations in (28) are the discrete analog of matching the second-order derivatives along \mathbf{v}_j at the ghost points and the corresponding boundary point x_j^B . Here the operator $\frac{\Delta u}{\Delta \mathbf{v}}$ is defined in Eq. (23).

Now we report the error between the function value $u(x_j^{G_k})$ and the discrete extrapolated function value $\tilde{u}_{\epsilon,j}^{G_k}$ obtained from solving the algebraic condition (27) and (28).

Proposition 3.6. (Extrapolation error rate for u) Let $u \in C^3(M \cup \Delta M)$, where ΔM satisfies the Assumption 3.4 and the Euclidean distance between any point in $x \in M$ and the extended boundary $\partial(M \cup \Delta M)$ is at least ϵ^r , where $0 < r < 1/2$ and suppose that $h = \mathcal{O}(\epsilon^r)$. Let \tilde{u}_ϵ be defined as in (27) and (28). For any fixed $j = 1, \dots, J$, the error rate

$$\left| u(x_j^{G_k}) - \tilde{u}_{\epsilon,j}^{G_k} \right| = \mathcal{O} \left(\epsilon^{3r}, \tilde{N}^{-1/2} \epsilon^{-(1+d/4)}, \tilde{N}^{-1/2} \epsilon^{(1/2-d/4)} \right).$$

Proof. See Appendix C. \square

Here, Propositions 3.5 and 3.6 are deduced based on the smoothness assumption of $M \cup \Delta M$ while the previous results, Propositions 3.1 and 3.3, are deduced based on the smoothness assumption of M .

3.5 The ghost point diffusion maps estimator

Here, we continue using the notation $\tilde{u}_\epsilon = (u_\epsilon(x_1), \dots, u_\epsilon(x_N), \tilde{u}_{\epsilon,1}^{G_1}, \dots, \tilde{u}_{\epsilon,1}^{G_K}, \dots, \tilde{u}_{\epsilon,J}^{G_1}, \dots, \tilde{u}_{\epsilon,J}^{G_K})^\top$ and $\tilde{u} = (u(x_1), \dots, u(x_N), u(x_1^{G_1}), \dots, u(x_1^{G_K}), \dots, u(x_J^{G_1}), \dots, u(x_J^{G_K}))^\top$. In addition, we also define $\tilde{u}_\epsilon^M = (u_\epsilon(x_1), \dots, u_\epsilon(x_N))^\top$, $\tilde{u}_\epsilon^G = (\tilde{u}_{\epsilon,1}^{G_1}, \dots, \tilde{u}_{\epsilon,J}^{G_K})^\top$, $\tilde{u}^M = (u(x_1), \dots, u(x_N))^\top$, $\tilde{u}^G = (u(x_1^{G_1}), \dots, u(x_1^{G_K}), \dots, u(x_J^{G_1}), \dots, u(x_J^{G_K}))^\top$ and notice that $\tilde{u}_\epsilon^M = \tilde{u}^M$.

Here, Eq. (28) consists of a system of JK equations and it has a unique solution that can be written in compact form as,

$$\tilde{u}_\epsilon^G = \mathbf{A} \tilde{u}_\epsilon^M + \tilde{b}, \quad (29)$$

where one can see the detailed expression of $\mathbf{A} \in \mathbb{R}^{JK \times N}$ and $\tilde{b} \in \mathbb{R}^{JK}$ for the 1D case in Appendix D. Here, components of \tilde{b} depend on $f(x_j^B)$. To this end, we denote the discrete approximation with a non-square matrix $\mathbf{L} = (\mathbf{L}^{(1)}, \mathbf{L}^{(2)}) \in \mathbb{R}^{N \times \tilde{N}}$ that maps vectors $\tilde{u} \in \mathbb{R}^{\tilde{N}}$ into $\mathbf{L}\tilde{u} \in \mathbb{R}^N$, whose i th component approximates one of the diffusion operators (3)-(5), $\mathcal{L}u(x_i)$ at $x_i \in M$. Since \tilde{u}^G is not available, the GPDM with the extrapolated function values \tilde{u}_ϵ^G approximates the diffusion operator \mathcal{L} with following affine operator,

$$\mathbf{L}\tilde{u}_\epsilon = \mathbf{L}^{(1)} \tilde{u}_\epsilon^M + \mathbf{L}^{(2)} \tilde{u}_\epsilon^G = \mathbf{L}^{(1)} \tilde{u}^M + \mathbf{L}^{(2)} (\mathbf{A} \tilde{u}_\epsilon^M + \tilde{b}) = (\mathbf{L}^{(1)} + \mathbf{L}^{(2)} \mathbf{A}) \tilde{u}_\epsilon^M + \mathbf{L}^{(2)} \tilde{b}. \quad (30)$$

We now give an error estimate for this operator approximation.

Theorem 3.7. (Consistency of the GPDM) Let $u \in C^3(M \cup \Delta M)$, where ΔM satisfies the Assumption 3.4 and the Euclidean distance between any point in $x \in M$ and the extended boundary $\partial(M \cup \Delta M)$ is at least ϵ^r , where $0 < r < 1/2$ and suppose that $h = \mathcal{O}(\epsilon^r)$. Let \tilde{u}_ϵ be defined as in (27) and (28) and $\tilde{u}_\epsilon = (u_\epsilon(x_1), \dots, u_\epsilon(x_N), \tilde{u}_{\epsilon,1}^{G_1}, \dots, \tilde{u}_{\epsilon,1}^{G_K}, \dots, \tilde{u}_{\epsilon,J}^{G_1}, \dots, \tilde{u}_{\epsilon,J}^{G_K})^\top$. Then,

$$|(\mathbf{L}\tilde{u}_\epsilon)_i - \mathcal{L}u(x_i)| = \mathcal{O} \left(\epsilon^{3r-1}, \tilde{N}^{-1/2} \epsilon^{-(2+d/4)}, \tilde{N}^{-1/2} \epsilon^{-(1/2+d/4)} \right).$$

Proof. For each $i = 1, \dots, N$, we have

$$\begin{aligned} |(\mathbf{L}\tilde{u}_\epsilon)_i - \mathcal{L}u(x_i)| &= |(\mathbf{L}^{(1)} \tilde{u}_\epsilon^M)_i + (\mathbf{L}^{(2)} \tilde{u}_\epsilon^G)_i - \mathcal{L}u(x_i)| \\ &\leq |(\mathbf{L}^{(1)} \tilde{u}^M)_i + (\mathbf{L}^{(2)} \tilde{u}^G)_i - \mathcal{L}u(x_i)| + |(\mathbf{L}^{(2)} (\tilde{u}_\epsilon^G - \tilde{u}^G))_i| \\ &= |(\mathbf{L}\tilde{u})_i - \mathcal{L}u(x_i)| + |(\mathbf{L}^{(2)} (\tilde{u}_\epsilon^G - \tilde{u}^G))_i|. \end{aligned} \quad (31)$$

Since each component of $\mathbf{L}^{(2)}$ is of order ϵ^{-1} , the proof is completed by balancing the pointwise error bound of the standard diffusion maps in Lemma C.1 (the first term in (31)) and the extrapolation error rates in Proposition 3.6 (the second term in (31)). \square

3.6 Numerical verification

In this section, we provide supporting numerical results of the GPDM method on the semi-ellipse Example 2.1 and assess the error of the affine operator in (30) in estimating $\mathcal{L}_2 u$ for functions u that satisfy various boundary conditions. For the Robin boundary condition, we set $\beta_1(x) = 1$, $\beta_2(x) = 3/(2a)$ with the homogeneous $g = 0$ at both boundary points, x_1 and x_N . For this numerical example, we set $\kappa = 1.1 + \sin \theta$. Choosing the true function to be $u = \cos(3\theta/2 - \pi/4)$, one can check that this function satisfies the above Robin boundary condition. The analytic

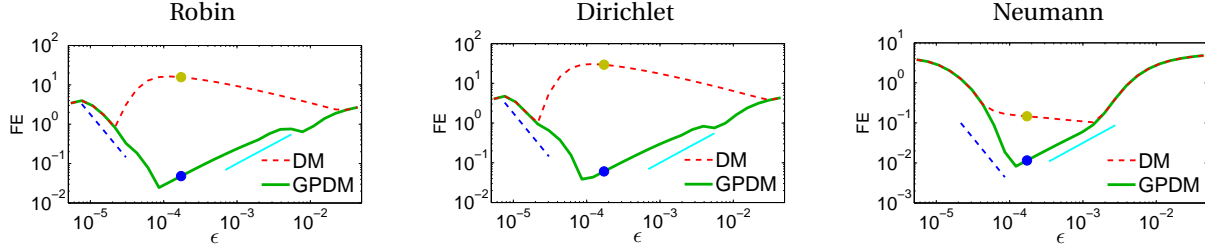


Figure 6: (Color online) Forward Error (FE) of the operator estimation as a function of the bandwidth ϵ . The operator acts on a test function satisfying homogeneous: (a) Robin, (b) Dirichlet, and (c) Neumann boundary conditions. The yellow point and blue point correspond to the auto-tuned ϵ for DM and GPDM, respectively. We use the auto-tune algorithm in [25]. The cyan solid line and the blue dashed line correspond to $O(\epsilon^1)$ and $O(\epsilon^{-(2+d/4)})$ with the intrinsic dimension $d = 1$, respectively.

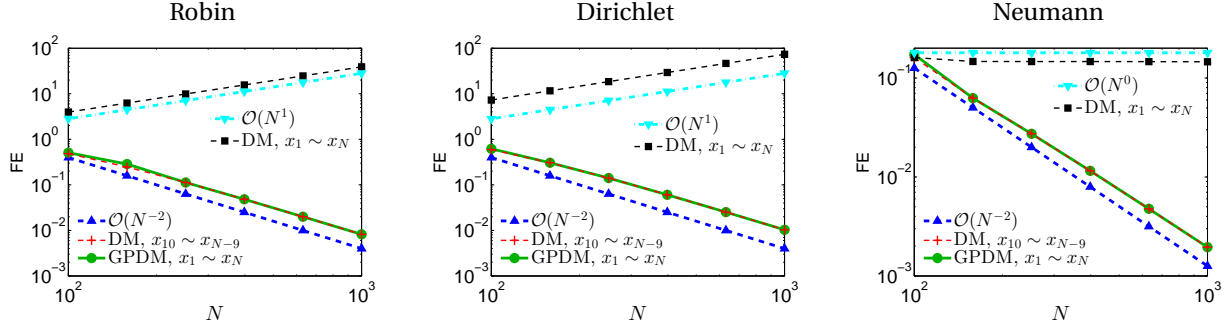


Figure 7: (Color online) Comparisons of Forward Errors (FEs) of the estimated operators as functions of the number of points N . The operator acts on a test function satisfying homogeneous: (a) Robin, (b) Dirichlet, and (c) Neumann boundary conditions. For GPDM, the FE $\|\mathbf{L}\tilde{u}_\epsilon - \mathcal{L}_2 u\|_\infty$ is computed on all points on the manifold, M (green solid line). For DM, The FE $\|\mathbf{L}\tilde{u}^M - \mathcal{L}_2 u\|_\infty$ is computed on all points on M (black dashed line) and only on interior points $\{x_i\}_{i=10, \dots, N-9}$ away from the boundary (red dashed line); i.e., neglecting nine closest points from each boundary point. The bandwidth ϵ is auto-tuned for each N .

$f = \mathcal{L}_2 u$ can be calculated from (15). For the Dirichlet and Neumann boundary conditions, we follow the same idea by choosing the appropriate u that satisfies the boundary condition and proceed in the similar fashion.

The components of u are evaluated at equally angle distributed points $\{\theta_i = \frac{(i-1)\pi}{N-1}\}_{i=1, \dots, N}$. In the following numerical experiment, we set $N = 400$ and $k = 50$ nearest neighbors (this is the same configuration that produces Fig. 1). Figure 6 shows the Forward Error (FE) defined as $\|\mathbf{L}\tilde{u}_\epsilon - \mathcal{L}_2 u\|_\infty$ as a function of the bandwidth parameter, ϵ , for various boundary conditions. One can see from the Fig. 6 that with the GPDM, the uniform FE reduces substantially on a wide range of $\epsilon = 10^{-5} - 10^{-2}$. This indicates that the solution of the GPDM becomes much more accurate for the ϵ tuning compared to the standard DM, even in the Neumann case. In addition, the numerical scaling for ϵ is consistent with the theoretical prediction, that is, the error grows on $\mathcal{O}(\epsilon^1)$ on a range of relatively large values of ϵ (cyan solid line) and decays on $\mathcal{O}(\epsilon^{-(2+d/4)})$ on a range of relatively small values of ϵ (blue dashed line).

Fig. 7 shows the FE as a function of the number of points N . The GPDM FE $\|\mathbf{L}\tilde{u}_\epsilon - \mathcal{L}_2 u\|_\infty$ (green curve) is a uniform error computed at all N points on manifold M . The DM FE $\|\mathbf{L}\tilde{u}^M - \mathcal{L}_2 u\|_\infty$ depicted by the black dashed curve is computed at all N points, whereas the DM FE depicted by the red dashed curve is computed only at points $x_{10} - x_{N-9}$ away from boundary. One can see from Fig. 7 that the DM FE on the interior of M and the GPDM error on all points of M decay on $\mathcal{O}(N^{-2})$ whereas the DM FE on M increases on $\mathcal{O}(N^1)$ for both Robin and Dirichlet BC's and of $\mathcal{O}(1)$ for Neumann BC's. This indicates that for DM, the increasing FE comes from the boundary when N increases. Incidentally, we notice that for the case of no boundary for manifold M , FE decays as $\mathcal{O}(N^{-2})$ (see [17]). However, in the presence of boundary condition, only the GPDM FE decays as $\mathcal{O}(N^{-2})$ and this decay rate is somewhat faster than the theoretical estimate in Theorem 3.7.

Fig. 8 shows that our results are not sensitive to different specifications of ghost points along the boundary. While there are various approaches to extend the original manifold M onto ghost points on ΔM , our method supplements the ghost points along the estimated normal direction of the boundary. Because of the persistent existence of error from the estimation of the normal direction \mathbf{v} , it is important to examine whether the GPDM is sensitive to different

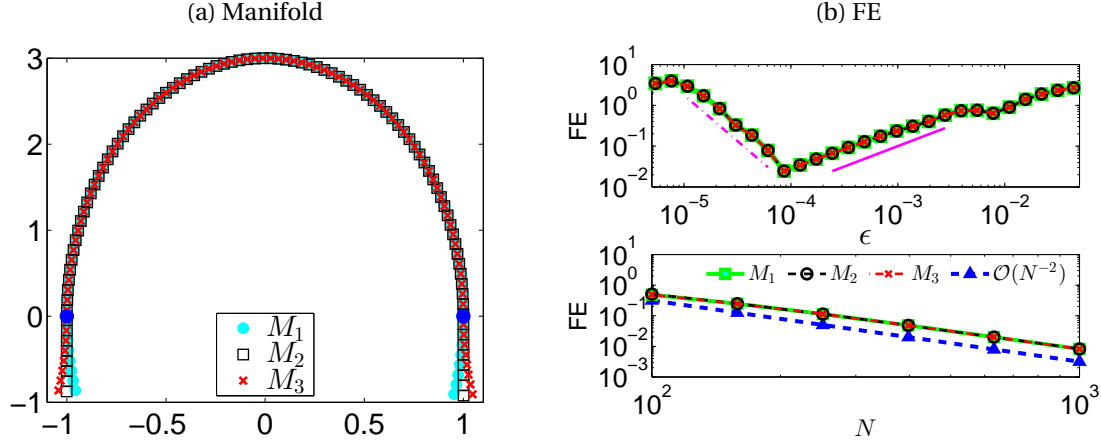


Figure 8: (Color online) Using different ghost points in (a), plotted are comparisons of FEs of operators (b) as functions of bandwidth ϵ and number of points N . The magenta solid line and the magenta dashed line correspond to $O(\epsilon^1)$ and $O(\epsilon^{-(2+d/4)})$ with the intrinsic dimension $d = 1$, respectively. One can see that our GPDM method is robust under different choices of ghost points.

choices of ghost points. Here, we examine three different choices of ghost points and their corresponding manifolds are denoted by M_1, M_2, M_3 as shown in Fig. 8(a). Fig. 8(b) displays the FE as a function of ϵ (upper) and as a function of N (bottom). One can see that all the FEs are almost the same under different M_1, M_2, M_3 . Therefore, our GPDM results are not sensitive to the choice of extensions of ghost points.

We will provide numerical examples on 2D manifolds in the next sections.

4 Applications: Solving linear elliptic PDE's

In this section, we consider solving the elliptic PDE's problem,

$$\begin{aligned} \mathcal{L}u &= f, & x \in M^o, \\ \mathcal{B}u &:= (\beta_1 \partial_\nu + \beta_2)u = g, & x \in \partial M, \end{aligned} \quad (32)$$

where β_1, β_2 are continuous real-valued functions, defined on ∂M . Here, the differential operator \mathcal{L} is one of (3)-(5) and is assumed to be uniformly elliptic with uniformly bounded coefficients (if any). When $\beta_1 = 0$, we have the Dirichlet boundary condition, when $\beta_2 = 0$, we have the Neumann boundary condition, and when both are nonzero, we have the Robin boundary condition. For the Neumann boundary condition, we will consider the PDE $(\mathcal{L} - a)u = f$ with $a(x) \geq a_{\min} > 0, \forall x \in M$ for a well-posed problem. For the Robin boundary condition, we also add $-a$ for convenience of the convergence study. Given $f \in C(M)$ and $g \in C(\partial M)$, the PDE problem admits a unique classical solution $u \in C^3(M)$.

In Section 4.1, we present and report the convergence of the proposed solver, constructed using the GPDM discretization. In Section 4.2, we provide supporting numerical examples on simple manifolds. In Section 4.3, we test the PDE solver on problems defined on an “unknown” manifold and compare the estimates with the Finite Element Method (FEM) solution.

4.1 The GPDM discretization method

Numerically, we will approximate the PDE in (32) with the affine operator in (30) for our GPDM method. Specifically, consider the $N - J$ rows corresponding to the interior points $x_i \in M^o$ of the following equation,

$$\mathbf{L}\hat{u} = (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})\hat{u}^M + \mathbf{L}^{(2)}\vec{b} = \vec{f}, \quad (33)$$

where $\vec{f} \in \mathbb{R}^N$, with components $f_i = f(x_i), x_i \in M$. In (33), recall that $(\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})$ is an $N \times N$ matrix, and $\mathbf{L}^{(2)}\vec{b} \in \mathbb{R}^N$. Also, we have introduced the notation $\hat{u} = (\hat{u}_1, \dots, \hat{u}_N, \hat{u}_{\epsilon,1}^{G_1}, \dots, \hat{u}_{\epsilon,1}^{G_K}, \dots, \hat{u}_{\epsilon,J}^{G_1}, \dots, \hat{u}_{\epsilon,J}^{G_K})^\top$ and $\hat{u}^M = (\hat{u}_1, \dots, \hat{u}_N)$, with \hat{u}_i denoting the numerical approximation to the solution $u(x_i)$ at $x_i \in M$.

As for the boundary condition, we discretize the boundary operator for each $x_i \in \partial M$ as,

$$\beta_1(x_i)\partial_v + \beta_2(x_i) \approx \beta_1(x_i) \frac{\Delta}{\Delta v} + \beta_2(x_i)\delta(x_i) = \beta_1(x_i) \left(\tilde{a}^L \frac{\delta(x^L) - \delta(x_i)}{|x^L - x_i|} + \tilde{a}^R \frac{\delta(x^R) - \delta(x_i)}{|x^R - x_i|} \right) + \beta_2(x_i)\delta(x_i) := \mathbf{B}_i, \quad (34)$$

where the coefficients \tilde{a}_L, \tilde{a}_R and the interpolation points $x_L, x_R \in M$ (all depend on x_i) are specified using the Algorithm 3.2 for 2D manifolds. For 1D manifolds, the discretization is much simpler, following Eq. (21). The Kronecker delta notation $\delta(x)$, which is equal to 1 on x and 0 otherwise, is used to clarify that the row vector \mathbf{B}_i (of size $1 \times N$) has nonzero components on entries associated to $x^L, x^R \in M$, and $x_i \in \partial M$. With this notation, the estimated boundary condition can be written in a compact form as,

$$\mathbf{B}\hat{u}^M = \tilde{\mathbf{g}}, \quad (35)$$

where $\tilde{\mathbf{g}} \in \mathbb{R}^J$, with components $g_i = g(x_i)$ for all $x_i \in \partial M$. Denoting $\tilde{u}^M = (u(x_1), \dots, u(x_M))^\top$ as the vector with true solution as the components,

$$(\mathbf{B}\hat{u}^M - \mathbf{B}\tilde{u}^M)_j = g(x_j^B) - (\mathbf{B}\tilde{u}^M)_j = \mathcal{B}u(x_j^B) - (\mathbf{B}\tilde{u}^M)_j = \mathcal{O}(\epsilon^{1/2}), \quad (36)$$

for $j = 1, \dots, J$, where we have used the error bound of the approximation of the directional derivative in Proposition 3.3.

Dirichlet Problem: Numerically, we consider solving an $(N-J) \times (N-J)$ linear problem that is obtained by asserting (35) to the first $N-J$ row of (33). To clarify, let us define the submatrices $\mathbf{L}^I \in \mathbb{R}^{(N-J) \times (N-J)}$, $\mathbf{L}^B \in \mathbb{R}^{(N-J) \times J}$ that satisfy,

$$(\mathbf{L}^I \quad \mathbf{L}^B) = \begin{pmatrix} (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})_1 \\ \vdots \\ (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})_{N-J} \end{pmatrix} \in \mathbb{R}^{(N-J) \times N}. \quad (37)$$

and decompose $\hat{u}^M = (\hat{u}^I, \hat{u}^B)^\top$ into the interior component, $\hat{u}^I = (\hat{u}_1, \dots, \hat{u}_{N-J})^\top$, and the boundary component, $\hat{u}^B = (\hat{u}_{N-J+1}, \dots, \hat{u}_N)^\top$. Similarly, we will use the notation $\tilde{u}^I = (u(x_1), \dots, u(x_{N-J}))^\top$ for the true solution at the interior points.

For the Dirichlet boundary condition, $u(x_j^B) = g(x_j^B)$ for $j = 1, \dots, J$, then one can directly replace $\hat{u}_{N-J+j} = u(x_j^B) = g(x_j^B)$, applying the decomposition in (37) on the first $N-J$ rows of (33), we arrive at the following reduced system,

$$\mathbf{L}^I \hat{u}^I = \tilde{f}^I - \mathbf{L}^B \tilde{\mathbf{g}} \quad (38)$$

where we have also defined $\tilde{f}^I = (f(x_1) - (\mathbf{L}^{(2)}\tilde{\mathbf{b}})_1, f(x_2) - (\mathbf{L}^{(2)}\tilde{\mathbf{b}})_2, \dots, f(x_{N-J}) - (\mathbf{L}^{(2)}\tilde{\mathbf{b}})_{N-J})^\top$. We can show that the solution of (38) converges to the solution of the PDE in (32) with Dirichlet boundary condition.

Theorem 4.1. (Convergence of the Dirichlet Problem) *Let u be the solution of PDE in (32) with Dirichlet boundary condition, $u(x_j^B) = g(x_j^B)$ for $j = 1, \dots, J$. Let \hat{u}_i be the solution of the linear system in (38), where the diffusion operator \mathcal{L} is approximated using the GPDM affine estimator in (30), constructed with N grid points on the manifold and ghost points of distance, \tilde{d} , where $\tilde{d} = \mathcal{O}(\epsilon^r)$ and $0 < r < 1/2$. Assume that the differential operator \mathcal{L} satisfies the maximum principle, then \hat{u}_i converges to $u(x_i)$ with error bound given as,*

$$|\hat{u}_i - u(x_i)| = \mathcal{O}\left(\epsilon^{3r-1}, \tilde{N}^{-1/2}\epsilon^{-(2+d/4)}, \tilde{N}^{-1/2}\epsilon^{-(1/2+d/4)}\right),$$

as $r \rightarrow 1/2$, for any $x_i \in M^\circ$.

Proof. See Appendix D. □

Robin and Neumann problems: Here, we consider

$$\begin{aligned} (-a + \mathcal{L})u &= f, & x \in M^\circ, \\ \mathcal{B}u &:= (\beta_1\partial_v + \beta_2)u = g, & x \in \partial M, \end{aligned} \quad (39)$$

with $a(x) \geq a_{\min} > 0$, $\forall x \in M$ such that $-a + \mathcal{L}$ is strictly negative definite. Here, the additional $-a$ term is to ensure the well-posedness of the Neumann problem and for convenience of the convergence study of the Robin problem.

For the discussion below, we write the discrete approximation of the boundary operator as $\mathbf{B} = (\mathbf{B}^I; \mathbf{B}^B)$, where $\mathbf{B}^I \in \mathbb{R}^{J \times (N-J)}$ and $\mathbf{B}^B \in \mathbb{R}^{J \times J}$. Then, the discrete approximation to the PDE problem in (39) is to solve the following $N \times N$ system,

$$\mathbf{N}\hat{u}^M := \begin{pmatrix} -\mathbf{a} + \mathbf{L}^I & \mathbf{L}^B \\ \mathbf{B}^I & \mathbf{B}^B \end{pmatrix} \begin{pmatrix} \hat{u}^I \\ \hat{u}^B \end{pmatrix} = \begin{pmatrix} \tilde{f}^I \\ \tilde{\mathbf{g}} \end{pmatrix}, \quad (40)$$

where \mathbf{a} denotes a diagonal matrix with diagonal components $\{a(x_i)\}$. Numerically, one can also solve the last J rows corresponding to the boundary conditions,

$$\hat{u}^B = (\mathbf{B}^B)^{-1}(\vec{g} - \mathbf{B}^I \hat{u}^I), \quad (41)$$

and insert this solution to the first $(N - J)$ -rows in problem (40) to obtain a reduced $(N - J) \times (N - J)$ system.

For the Robin problem, we have the following convergence result.

Theorem 4.2. (Convergence of the Robin Problem) *Let u be the solution of PDE in (39) with Robin boundary condition and $\beta_1, \beta_2 > 0$. The solution \hat{u}_i of the linear system in (40) converges to $u(x_i)$ with error bound given as,*

$$|\hat{u}_i - u(x_i)| = \mathcal{O}\left(\epsilon^{3r-1}, \tilde{N}^{-1/2} \epsilon^{-(2+d/4)}, \tilde{N}^{-1/2} \epsilon^{-(1/2+d/4)}\right),$$

as $r \rightarrow 1/2$, for any $x_i \in M$.

Proof. Using the definition of \tilde{f}^I and the decomposition in (37), one can immediately see the consistency. Multiplying the matrix \mathbf{N} in (40) with a vector consists of the difference between the estimated and the true solutions, we obtain

$$\begin{aligned} ((-\mathbf{a} + \mathbf{L}^I)(\hat{u}^I - \tilde{u}^I) + \mathbf{L}^B(\hat{u}^B - \tilde{u}^B))_i &= \left(\tilde{f}^I - (-\mathbf{a} + \mathbf{L}^I)\tilde{u}_I - \mathbf{L}^B\tilde{u}_B\right)_i \\ &= f(x_i) + a(x_i)u(x_i) - (\mathbf{L}^{(2)}\tilde{b} + \mathbf{L}^I\tilde{u}_I + \mathbf{L}^B\tilde{u}_B)_i \\ &= \mathcal{L}u(x_i) - (\mathbf{L}\tilde{u}_\epsilon)_i. \end{aligned} \quad (42)$$

for $i = 1, \dots, N - J$. The last J rows corresponding to the boundary points are nothing but (36).

From Eq. (88) in Appendix D, the column sum of each row of the matrix $\mathbf{M} = \epsilon(\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})$ is zero and that $\mathbf{M}_{i,i} < 0$ and $\mathbf{M}_{i,j} > 0$ for all $j \neq i$. Since the first $N - J$ rows of \mathbf{N} is nothing but $-a(x_i) + (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})_i$, we have

$$|\mathbf{N}_{i,i}| - \sum_{\substack{j=1 \\ j \neq i}}^N |\mathbf{N}_{i,j}| = |-a(x_i) + \epsilon^{-1}\mathbf{M}_{i,i}| - \epsilon^{-1} \sum_{\substack{j=1 \\ j \neq i}}^N |\mathbf{M}_{i,j}| = a(x_i) - \epsilon^{-1} \sum_{j=1}^N \mathbf{M}_{i,j} = a(x_i) \geq a_{\min} > 0, \quad (43)$$

for $i = 1, \dots, N - J$. Also, the last J rows of the matrix \mathbf{N} are strictly diagonal dominant as long as $\beta_2 > 0$. For example, in 1D case where $J = 2$, the last two rows of (40) is given as,

$$\mathbf{B}^B \hat{u}_B + \mathbf{B}^I \hat{u}_I := \begin{pmatrix} \frac{\beta_1(x_1)}{d} + \beta_2(x_1) & 0 \\ 0 & \frac{\beta_1(x_N)}{d} + \beta_2(x_N) \end{pmatrix} \begin{pmatrix} \hat{u}_1 \\ \hat{u}_N \end{pmatrix} + \begin{pmatrix} -\frac{\beta_1(x_1)}{d} & 0 & \dots & 0 \\ 0 & \dots & 0 & -\frac{\beta_1(x_N)}{d} \end{pmatrix} \begin{pmatrix} \hat{u}_2 \\ \hat{u}_3 \\ \vdots \\ \hat{u}_{N-1} \end{pmatrix} = \begin{pmatrix} g(x_1) \\ g(x_N) \end{pmatrix} := \vec{g}.$$

In this case, $|\mathbf{N}_{i,i}| - \sum_{\substack{j=1 \\ j \neq i}}^N |\mathbf{N}_{i,j}| = \beta_2(x_i) > 0$ for $i > N - J$. Therefore, the matrix \mathbf{N} is strictly diagonal dominant and nonsingular. By the Ahlberg-Nilson-Varah bound [33, 34], the inverse matrix is uniformly bounded,

$$\|\mathbf{N}^{-1}\|_\infty \leq \frac{1}{\min_i (|\mathbf{N}_{i,i}| - \sum_{\substack{j=1 \\ j \neq i}}^N |\mathbf{N}_{i,j}|)} = \frac{1}{\min\{a_{\min}, \beta_2\}}. \quad (44)$$

Thus, multiplying \mathbf{N}^{-1} to a vector where the first $N - J$ components consist of (42) and the last J components consist of (36), we have

$$\begin{aligned} |\hat{u}_i - u(x_i)| &\leq \|\mathbf{N}^{-1}\|_\infty \left(\max_{i=1, \dots, N-J} |((-\mathbf{a} + \mathbf{L}^I)(\hat{u}^I - \tilde{u}^I) + \mathbf{L}^B(\hat{u}^B - \tilde{u}^B))_i|, \max_{j=1, \dots, J} |(\mathbf{B}(\hat{u}^M - \tilde{u}^M))_j| \right) \\ &= \|\mathbf{N}^{-1}\|_\infty \left(\max_{i=1, \dots, N-J} |\mathcal{L}u(x_i) - (\mathbf{L}\tilde{u}_\epsilon)_i|, \max_{j=1, \dots, J} |(\mathbf{B}(\hat{u}^M - \tilde{u}^M))_j| \right), \end{aligned} \quad (45)$$

for all $x_i \in M$. Since the GPDM is consistent, $|\mathcal{L}u(x_i) - (\mathbf{L}\tilde{u}_\epsilon)_i| \rightarrow 0$ as $\epsilon \rightarrow 0, N \rightarrow \infty$ with error rate given in Theorem 3.7 and using the error bound in (36), the proof is completed. \square

For the Neumann problem, the last J components of \mathbf{N} is not strictly diagonal dominant, since $\beta_2 = 0$. To achieve the convergence, one can consider (without loss of generality) the homogeneous Neumann problem $g = 0$ such that (41) simplifies to $\hat{u}_B = -(\mathbf{B}^B)^{-1}\mathbf{B}^I \hat{u}^I$. For example in 1D case, the discrete approximation in (21) yields $\hat{u}_N = \hat{u}_{N-1}$ and $\hat{u}_1 = \hat{u}_2$. Substituting these solutions (J equations in general) to the first $N - J$ rows of (40), one can verify that the reduced $N - J$ problem is nonsingular and it has an inverse that is uniformly bounded by $1/a_{\min}$. Thus, the convergence can be achieved using the similar argument as in the proof above.

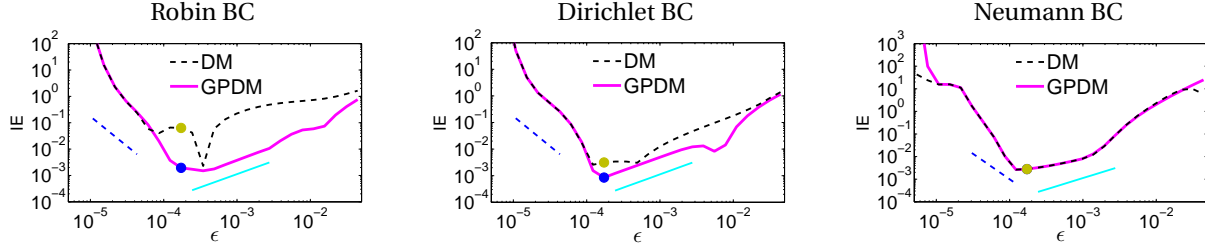


Figure 9: (Color online) Pointwise Inverse Error (IE) of the solution of (46) as a function of the bandwidth ϵ . The yellow point and blue point correspond to the auto-tuned ϵ for DM and GPDM, respectively. The cyan solid line and the blue dashed line correspond to $O(\epsilon^1)$ and $O(\epsilon^{-(2+d/4)})$ with the intrinsic dimension $d = 1$, respectively.

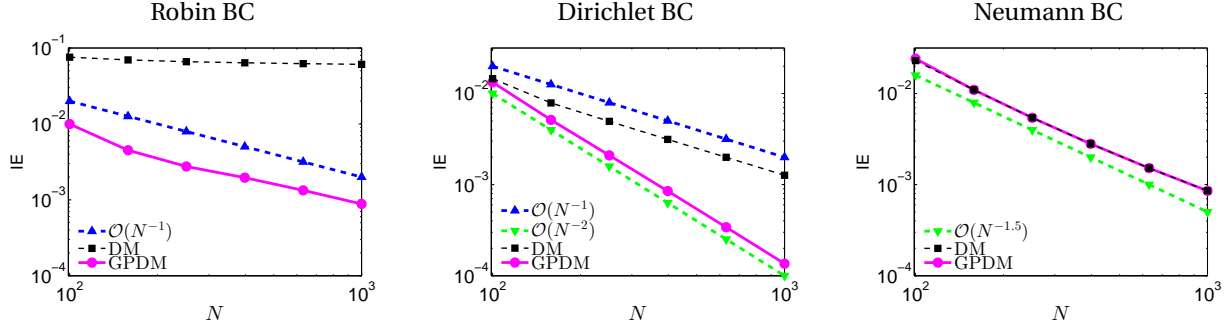


Figure 10: (Color online) Comparisons of the Inverse Errors (IEs) of the solutions of (46) as functions of N .

4.2 Numerical examples on simple manifolds

In this section, we discuss three examples on problems defined on simple manifolds. First, we verify the convergence rate with the 1D example in Example 2.1. In the second example, we test the solver on a semi-torus embedded in \mathbb{R}^3 with a mixed-type Dirichlet-Neumann, boundary condition. In the third example, we verify the effectiveness of the proposed method on a 2D disk embedded in \mathbb{R}^5 .

Numerically, we will compare GPDM with the standard DM. For non-Neumann boundary conditions, we modified the standard DM as follows. We consider the $N - J$ rows corresponding to the interior points $x_i \in M^o$ of the equation, $\mathbf{L}_{\text{DM}} \hat{u} = \hat{f}$, where \mathbf{L}_{DM} is the standard diffusion maps operator. Then the PDE solver is constructed exactly as in GPDM with \mathbf{L} in (33) replaced with \mathbf{L}_{DM} .

Anisotropic diffusion on a semi-ellipse: First, let us present the results of the 1D example in Example 2.1, solving

$$\mathcal{L}_2 u = f, \quad (46)$$

on the three boundary conditions. In this numerical experiment, the configuration is exactly the same as in Section 3.6. Particularly, Fig. 9 demonstrates the error of the solutions, $\|\hat{u}^M - \bar{u}^M\|_\infty$, which we refer as the Inverse Error (IE), as a function of ϵ for $N = 400, k = 50$. Compared to the standard diffusion maps, notice that GPDM is more robust for the case of Robin and Dirichlet boundary conditions, as expected. The advantage of GPDM over DM on Robin and Dirichlet boundary conditions is more apparent in Fig. 10. Particularly, for the Robin BC, one can see that the GPDM IE decays on $\mathcal{O}(N^{-1})$ whereas the DM IE never decays and is nearly constant. For the Dirichlet BC, GPDM IE decays faster compared to the DM IE. For the Neumann BC, we see comparable IEs as functions of N , as expected.

Non-symmetric backward Kolmogorov elliptic PDE on a semi-torus: In the next example, we consider solving $\mathcal{L}_3 u = f$, with a mixed Dirichlet-Neumann boundary conditions on a semi-torus $M \subset \mathbb{R}^3$. The differential operator \mathcal{L}_3 is defined as in (5) with

$$\begin{aligned} b(\theta, \phi) &= \begin{pmatrix} b^1 \\ b^2 \end{pmatrix} = \begin{pmatrix} 2 + \sin \theta \\ 2 + \cos \theta \end{pmatrix}, \\ c(\theta, \phi) &= \begin{pmatrix} c^{11} & c^{12} \\ c^{21} & c^{22} \end{pmatrix} = \begin{pmatrix} 3 + \cos \phi & 1/10 \\ 1/10 & 2 \end{pmatrix}. \end{aligned} \quad (47)$$

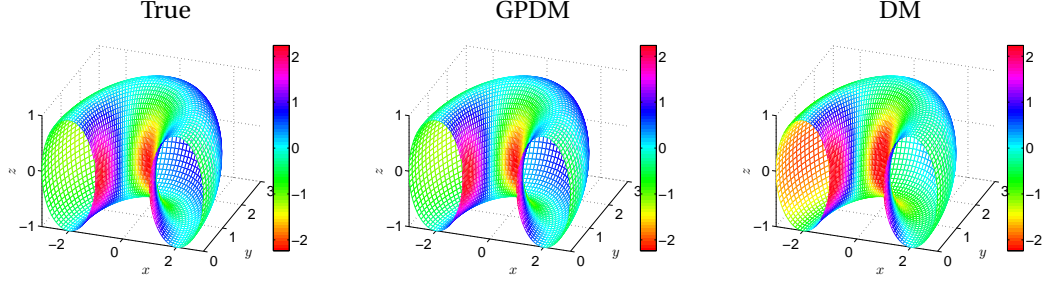


Figure 11: (Color online) Comparison of the solutions u among (a) truth, (b) GPDM, and (c) DM.

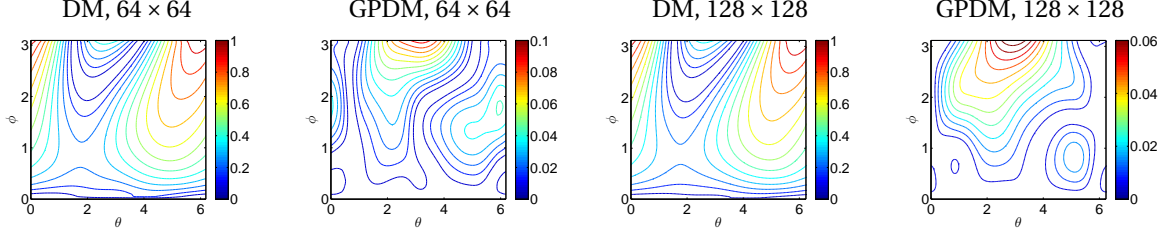


Figure 12: (Color online) Absolute differences in the estimated solutions: (a) DM for $N = 64 \times 64$ with $\|\tilde{u}_\epsilon - u\|_\infty = 0.97$, (b) GPDM for $N = 64 \times 64$ with $\|\tilde{u}_\epsilon - u\|_\infty = 0.10$, (c) DM for $N = 128 \times 128$ with $\|\tilde{u}_\epsilon - u\|_\infty = 0.97$, (d) GPDM for $N = 128 \times 128$ with $\|\tilde{u}_\epsilon - u\|_\infty = 0.06$.

The semi-torus is defined with the standard embedding function as in (19) so that the induced Riemannian metric is given in (20). We set an analytic solution of this problem to be

$$u(\theta, \phi) = \left(\sin 2\phi - \frac{2 \cos 2\phi}{2 + \cos \theta} \right) \cos \theta, \quad (48)$$

and calculate $f := \mathcal{L}_3 u$ and $g := \beta_1 \partial_{\mathbf{v}} u + \beta_2 u$ at $\phi = 0$ and $\phi = \pi$. Then, we approximate the solution in (48), for the PDE problem in (32) subjected to the manufactured f and g . In this semi-torus example, the explicit expression for f is given by

$$f := \mathcal{L}_3 u = b \cdot \nabla u + \frac{1}{2} c^{ij} \nabla_i \nabla_j u = b^1 \frac{\partial u}{\partial \theta} + b^2 \frac{\partial u}{\partial \phi} + \frac{1}{2} c^{11} \frac{\partial^2 u}{\partial \theta^2} + c^{12} \left(\frac{\partial^2 u}{\partial \theta \partial \phi} - \Gamma_{12}^2 \frac{\partial u}{\partial \phi} \right) + \frac{1}{2} c^{22} \left(\frac{\partial^2 u}{\partial \phi^2} - \Gamma_{22}^1 \frac{\partial u}{\partial \theta} \right), \quad (49)$$

where Γ_{12}^2 and Γ_{22}^1 are the only nontrivial Christoffel symbols of the second kind

$$\Gamma_{12}^2 = -\frac{\sin \theta}{2 + \cos \theta}, \quad \Gamma_{22}^1 = \sin \theta (2 + \cos \theta).$$

At one boundary $\phi = 0$, the parameters are $\beta_1 = 0$ and $\beta_2 = 1$ (Dirichlet boundary condition) so that $g := u(\phi = 0)$, where u is the analytic solution in (48). At the other boundary $\phi = \pi$, the parameters are $\beta_1 = 1$ and $\beta_2 = 1$ (Robin boundary condition) so that the expression for g at $\phi = \pi$ is

$$g := \beta_1 \partial_{\mathbf{v}} u + \beta_2 u = \left(\frac{1}{2 + \cos \theta} \frac{\partial u}{\partial \phi} + u \right) (\phi = \pi) = 0,$$

where the analytic u in (48) and $\phi = \pi$ have been used. Numerically, the grid points $\{\theta_i, \phi_j\}$ are uniformly distributed on $[0, 2\pi] \times [0, \pi]$, with $i, j = 1, \dots, 64$ or $i, j = 1, \dots, 128$ points in each direction, resulting in a total of $N = 4096$ or $N = 16384$ grid points. To apply the local kernel in (6), we use $k = 128$ nearest neighbors for computational efficiency and auto tune the kernel bandwidth as $\epsilon = 0.004$ for $N = 4096$ and $\epsilon = 0.001$ for $N = 16384$.

Fig. 11 shows the comparison of solutions for $N = 64 \times 64$ among (a) truth, (b) GPDM, and (c) DM. To quantify the accuracy, Fig. 12 shows the absolute differences between the true and the estimated solutions obtained using DM and GPDM for $N = 64 \times 64$ and $N = 128 \times 128$. For DM, the IE $\|\tilde{u}_\epsilon - u\|_\infty = 0.97$ is relatively large and IE does not decrease even as N increases. On the other hand, the inverse error (IE) of GPDM is one magnitude order smaller than the IE of DM and decreases from 0.10 to 0.06 as N is increased from 64×64 to 128×128 .

Laplace-Beltrami on a disk embedded in \mathbb{R}^5 : We consider solving a Laplace-Beltrami equation $\mathcal{L}_1 u = f$ on a 2D disk, embedded in \mathbb{R}^5 via the following map,

$$x := (x_1, x_2, x_3, x_4, x_5) = (\sin \varphi \cos \theta, \sin \varphi \sin \theta, \sin \varphi \cos 2\theta, \sin \varphi \sin 2\theta, \sqrt{2} \cos \varphi) \in M \subset \mathbb{R}^5,$$

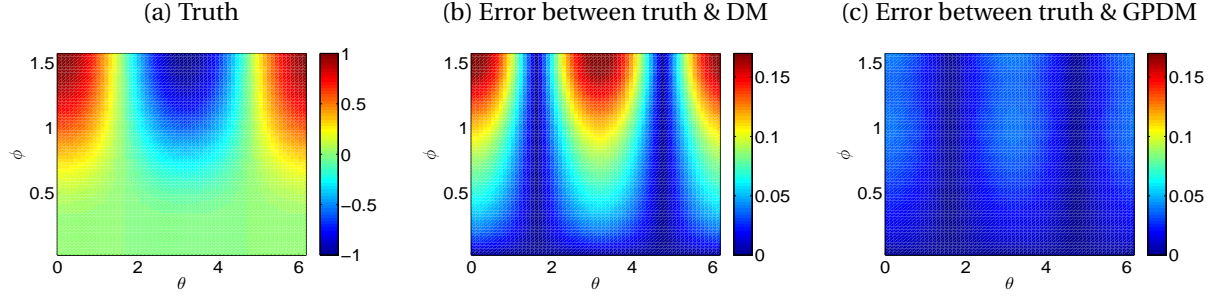


Figure 13: (Color online) Laplace-Beltrami on a disk embedded in \mathbb{R}^5 with Robin boundary condition. (a) True solution. (b) Absolute difference between the true and DM solutions (the inverse error, defined with the uniform norm, is 0.17) (c) Absolute difference between the true and GPDM solutions (the inverse error is 0.04).

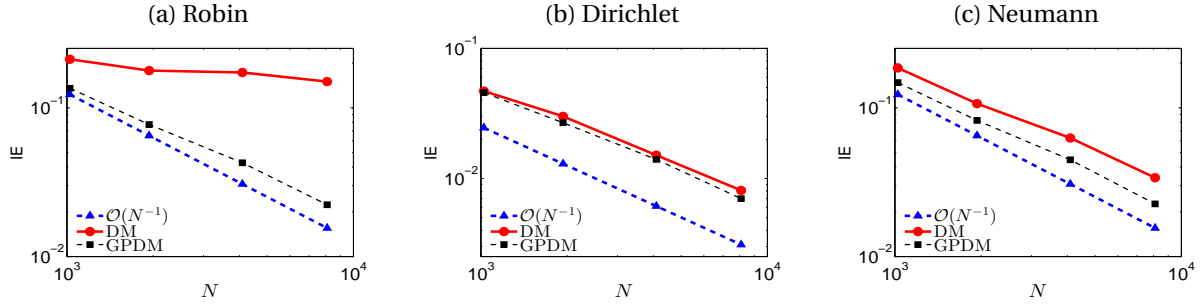


Figure 14: (Color online) Comparison of the inverse errors (IE) as functions of N for the (a) Robin, (b) Dirichlet, and (c) Neumann boundary conditions on a 2D disk embedded in \mathbb{R}^5 .

for $0 \leq \theta \leq 2\pi$ and $0 \leq \varphi \leq \pi/2$. The Riemannian metric is given by

$$g_{x^{-1}(\theta, \varphi)}(u, v) = u^\top \begin{pmatrix} 5 \sin^2 \varphi & 0 \\ 0 & 2 \end{pmatrix} v, \quad u, v \in T_{x^{-1}(\theta, \varphi)} M.$$

We set the true test function to be $u = \sin^3 \varphi \cos \theta$ and calculate $f := \mathcal{L}_1 u$ as

$$f = \frac{1}{\sqrt{|g|}} \frac{\partial}{\partial \theta} \left(\sqrt{|g|} g^{11} \frac{\partial u}{\partial \theta} \right) + \frac{1}{\sqrt{|g|}} \frac{\partial}{\partial \varphi} \left(\sqrt{|g|} g^{22} \frac{\partial u}{\partial \varphi} \right).$$

At the boundary $\varphi = \pi/2$, we set $\beta_1 = 1$ and $\beta_2 = 1$ for the Robin, $\beta_1 = 0$ and $\beta_2 = 1$ for the Dirichlet, $\beta_1 = 1$ and $\beta_2 = 0$ for the Neumann boundary conditions. We can calculate the corresponding $g := \beta_1 \partial_\nu u + \beta_2 u$ at the boundary.

Numerically, the grid points $\{\theta_i, \varphi_j\}$ are uniformly distributed on $[0, 2\pi] \times [0, \pi/2]$, with $i, j = 1, \dots, I$ ($I = 32, 44, 64, 90$) points in each direction resulting in a total of $N = 1024, 1936, 4096, 8100$ grid points. To approximate the Laplace-Beltrami operator, we use $k = 128$ nearest neighbors for computational efficiency and auto tune the kernel bandwidth ϵ for each N .

Figure 13 shows the true solution and the absolute errors of DM and GPDM with $N = 64 \times 64$ grid points for the Robin boundary condition. One can clearly see that GPDM produces a more accurate solution compare to the standard DM. Figure 14 shows the errors of the solutions as functions of N for the (a) Robin, (b) Dirichlet, and (c) Neumann boundary conditions. One can see from Fig. 14(a) that for Robin boundary, GPDM provides the convergent solution whereas DM does not. From Figs. 14 (b) and (c) for the Dirichlet and Neumann boundaries, respectively, one can see that both GPDM and DM provide convergent solutions.

4.3 Anisotropic diffusion on an unknown “face” manifold

In this section, we consider solving the boundary value problem in (46) with $\kappa = 1.1 + \sin^2(10x_1)$ and $f = \cos(10x_2)$ on an unknown manifold example of a two-dimensional “face” $x = (x_1, x_2, x_3) \in M \subset \mathbb{R}^3$. We consider the Robin boundary condition on the one dimensional-closed boundary curve of the face. The surface used in this section is from Keenan Crane’s 3D repository [35]. Notice that we have no access to the analytic solution since we do not know the embedding of the face surface. For comparison, we numerically solve the problem with finite element method (FEM) using the FELICITY FEM Matlab toolbox [36].

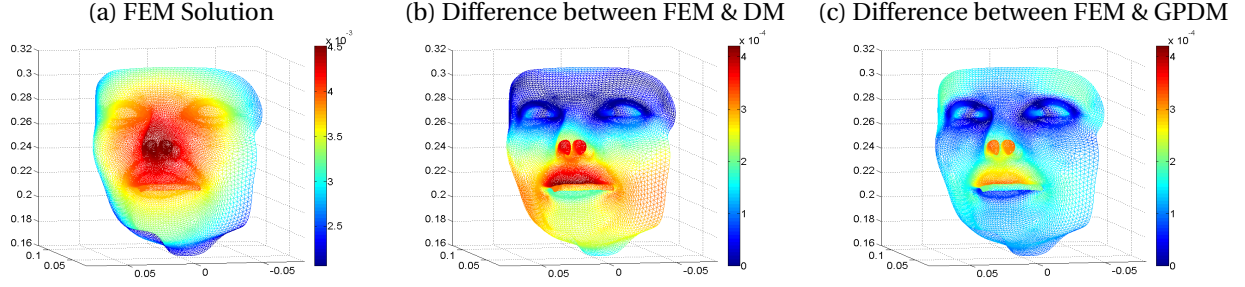


Figure 15: (Color online) Comparison of the PDE solutions among FEM, DM, and GPDM on face example with Robin boundary condition. (a) FEM solution. (b) Absolute difference between FEM and DM solutions. (c) Absolute difference between FEM and GPDM solutions.

Fig. 15 shows the comparison of the solutions among FEM, DM, and GPDM methods corresponding the Robin boundary condition ($\partial_\nu u + 10u = 0$ on ∂M). To compute the FEM solution as a benchmark, we applied FELICITY toolbox in Matlab using the triangulated mesh of the surface, which consisted of 17157 points and a connectivity matrix for the triangle elements. We use a linear finite element space in the FEM algorithm. We used $k = 512$ nearest neighbors and tuned the kernel bandwidth parameter as $\epsilon = 3 \times 10^{-6}$. For GPDM, we used $K = 6$ layers of ghost points for 168 boundary points so that we used 168×6 ghost points in total. In Fig. 15, we found that the inverse error (IE) between GPDM and FEM solutions (about 3.2×10^{-4}) is smaller than that between DM and FEM solutions (about 4.3×10^{-4}). In this case, one can see that larger errors of GPDM are locally concentrated near the nose and the mouth whereas the larger errors for DM are evenly distributed on the lower face. Thus, for the Robin boundary condition, one can see that GPDM exhibits a better performance than the standard DM.

5 Applications: Solving elliptic eigenvalue problems

In this section, we apply the GPDM algorithm for solving eigenvalue problems $\mathcal{L}\psi_k = \lambda_k\psi_k$ on manifold with boundary, where \mathcal{L} is either the Laplace-Beltrami in (3) or the weighted Laplacian operator in (4). Since there are no f in this problem, we do not use the extrapolation formula in (28). Instead, we extrapolate u using the lower-order extrapolation formula in (26). More specifically, we consider the following algorithm.

Algorithm 5.1. GPDM algorithm for eigenvalue problems:

1. Supplement the ghost points as in section 3.3 and construct the augmented $\tilde{N} \times \tilde{N}$ matrix using DM based on all points on manifold and ghost points.
2. Construct the GPDM estimator, an $(N - J) \times (N - J)$ matrix, based on the homogeneous extrapolation formula (26) for u at the ghost points and the homogeneous boundary condition (35). Here, J is the number of boundary points and κ is replaced with u in the extrapolation formula (26). The homogeneous equations (26) and (35) have a unique solution that can be written in compact form as,

$$(\tilde{u}^B, \tilde{u}_\epsilon^G)^\top = \mathbf{C}\tilde{u}^I,$$

where \tilde{u}_ϵ^G , \tilde{u}^B , and \tilde{u}^I are vectors with components consist of function values of u evaluated at the ghost points, boundary points, and interior points, respectively. With this notation, notice that $\tilde{u}_\epsilon = (\tilde{u}^I, \tilde{u}^B, \tilde{u}_\epsilon^G)^\top$. Here, \mathbf{C} is a $(JK + J) \times (N - J)$ matrix. Then, the diffusion operator \mathcal{L} is approximated with the following matrix,

$$\mathbf{L}\tilde{u}_\epsilon = \mathbf{L}^{(1)}\tilde{u}^I + \mathbf{L}^{(2)}(\tilde{u}^B, \tilde{u}_\epsilon^G)^\top = \mathbf{L}^{(1)}\tilde{u}^I + \mathbf{L}^{(2)}\mathbf{C}\tilde{u}^I \equiv (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{C})\tilde{u}^I.$$

Here, we have defined the submatrices $\mathbf{L}^{(1)} \in \mathbb{R}^{(N-J) \times (N-J)}$ and $\mathbf{L}^{(2)} \in \mathbb{R}^{(N-J) \times (JK+J)}$ of the augmented $(N - J) \times \tilde{N}$ matrix $\mathbf{L} \equiv (\mathbf{L}^{(1)}, \mathbf{L}^{(2)})$, and we should point out that these submatrices are different than those defined in (30).

3. Solve the eigenvalue problem of the diffusion matrix $\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{C}$.

For comparison, we also apply the standard DM algorithm for solving eigenvalue problems $\mathcal{L}\psi_k = \lambda_k\psi_k$ with the following modification to incorporate other boundary conditions.

Algorithm 5.2. DM algorithm for eigenvalue problems with non-Neumann boundary conditions:

1. Construct the DM estimator, an $(N - J) \times (N - J)$ matrix, based on the homogeneous boundary condition (35). Here, J is the number of boundary points. The homogeneous boundary condition (35) has a unique solution that can be written in compact form as,

$$\tilde{u}^B = \mathbf{C}_{\text{DM}} \tilde{u}^I,$$

where \tilde{u}^B and \tilde{u}^I are vectors with components consist of function values of u evaluated at the boundary points and interior points, respectively. Here, \mathbf{C}_{DM} is a $(J) \times (N - J)$ matrix. Then, the diffusion operator \mathcal{L} can be approximated with the following matrix,

$$\mathbf{L}_{\text{DM}} \tilde{u}_\epsilon = \mathbf{L}_{\text{DM}}^{(1)} \tilde{u}^I + \mathbf{L}_{\text{DM}}^{(2)} \tilde{u}^B = \mathbf{L}_{\text{DM}}^{(1)} \tilde{u}^I + \mathbf{L}_{\text{DM}}^{(2)} \mathbf{C}_{\text{DM}} \tilde{u}^I \equiv (\mathbf{L}_{\text{DM}}^{(1)} + \mathbf{L}_{\text{DM}}^{(2)} \mathbf{C}_{\text{DM}}) \tilde{u}^I.$$

Here, we have defined the submatrices $\mathbf{L}_{\text{DM}}^{(1)} \in \mathbb{R}^{(N-J) \times (N-J)}$ and $\mathbf{L}_{\text{DM}}^{(2)} \in \mathbb{R}^{(N-J) \times (J)}$ of the $(N - J) \times N$ DM matrix $\mathbf{L}_{\text{DM}} \equiv (\mathbf{L}_{\text{DM}}^{(1)}, \mathbf{L}_{\text{DM}}^{(2)})$.

2. Solve the eigenvalue problem of the diffusion matrix $\mathbf{L}_{\text{DM}}^{(1)} + \mathbf{L}_{\text{DM}}^{(2)} \mathbf{C}_{\text{DM}}$.

Next, we compare the numerical performance of the DM and GPDM in solving the eigenvalue problems $\mathcal{L}\psi_k = \lambda_k \psi_k$ on manifold with boundary on various test examples. We begin with the singular Sturm-Liouville eigenvalue problem of Legendre polynomials on a flat domain $[-1, 1]$. Next, we show numerical results of the Laplace-Beltrami operator on various embedded smooth manifolds, such as a 1D semi-circle in \mathbb{R}^2 with Dirichlet and Robin boundary conditions, and a 2D semi-torus in \mathbb{R}^2 with a mixed boundary condition.

5.1 A singular Sturm-Liouville problem

First, we consider solving the Legendre differential equation on the flat domain $[-1, 1]$,

$$\mathcal{L}\psi_k := \frac{d}{dx} \left[(1 - x^2) \frac{d\psi_k}{dx} \right] = -k(k + 1) \psi_k, \quad (50)$$

where the eigenvalues are $\lambda_k = -k(k + 1)$ with $k = 0, 1, 2, \dots$, and the eigenfunctions ψ_k are Legendre polynomials. The Legendre polynomials are orthogonal with respect to a uniform distribution weight over the domain $[-1, 1]$. The completeness of the set of eigenfunctions follows from the framework of Sturm-Liouville theory. It is well-known that the differential equation (50) has singular points at the boundary $x = \pm 1$, so that the eigenfunctions ψ_k are required to be regular at $x = \pm 1$.

Numerically, the operator \mathcal{L} in (50) is estimated by choosing $\kappa = 1 - x^2$ in the weighted Laplacian operator \mathcal{L}_2 in (4) using the GPDM method. At the boundary $x = \pm 1$, \mathcal{L} reduces to a first-order differential operator $\mathcal{L}\psi_k = -2x \frac{d\psi_k}{dx}$, so that it can be treated as a boundary condition which is estimated using a finite-difference method. In particular, we construct an $N \times N$ diffusion matrix on N equally spaced discrete grids $\{x_i = 2(i - 1)/(N - 1) - 1\}_{i=1, \dots, N}$ on $[-1, 1]$. For efficient computation, the sparse diffusion matrix is represented using the kernel generated from $k = 50$ nearest neighbors based on the Euclidean distance of x_i [23]. The bandwidth $\epsilon = 1.5 \times 10^{-5}$ is chosen for $N = 400$ by auto tuning [14, 23].

Fig. 16 shows the comparison of the eigenvalues and eigenfunctions between the analytic Legendre polynomials and the numerical results from DM and GPDM. It can be seen from Fig. 16 that both eigenvalues and eigenfunctions can be well approximated within numerical accuracy. For a detail inspection, we show the errors of the eigenvalues and of the eigenfunctions as functions of k , respectively, for different number of points N in Fig. 17. It can be seen that both DM and GPDM provide convergent eigenvalues and eigenfunctions as N increases. The errors of GPDM are slightly smaller than those of DM.

5.2 Laplace-Beltrami operator on a semi-circle

In this example, we consider solving the eigenvalue problem for $\Delta\psi_k = \lambda_k \psi_k$ on a 1D semi-circle with Dirichlet and Robin boundary conditions. We neglect showing results with the Neumann boundary condition since the performances of GPDM and DM are identical. The Riemannian metric of the semi-circle is given by (14) with $a = 1$. For the Dirichlet boundary condition $\psi_k = 0$ at both ends $\theta = 0$ and π , one can check that the eigenvalues and eigenfunctions are

$$\lambda_k = -k^2, \quad \psi_k = \sin(kx), \quad \text{for } k = 1, 2, 3, \dots \quad (51)$$

For the Robin boundary condition $-\partial_\nu \psi_k + \psi_k = 0$ at $\theta = 0$ and $\partial_\nu \psi_k + \psi_k = 0$ at $\theta = \pi$, we can find the explicit expression of both the eigenvalues and eigenfunctions,

$$\lambda_k = \begin{cases} 1 \\ -(k-1)^2 \end{cases}, \quad \psi_k = \begin{cases} \exp(-x), & \text{for } k = 1 \\ \sin((k-1)x) - (k-1) \cos((k-1)x), & \text{for } k = 2, 3, \dots \end{cases} \quad (52)$$

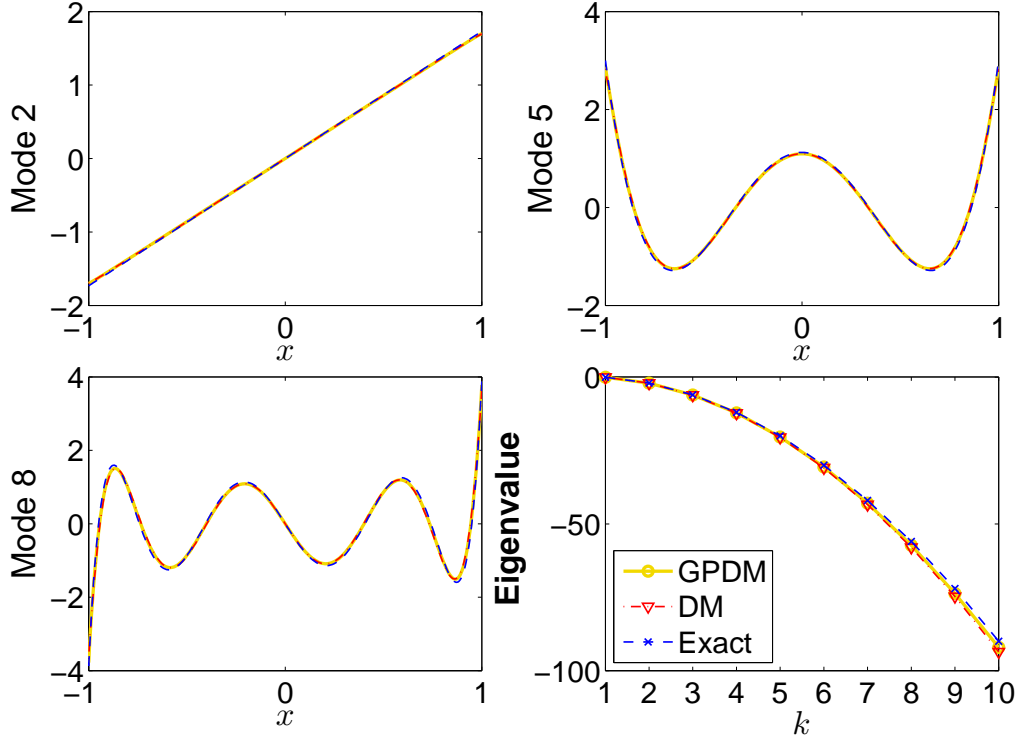


Figure 16: (Color online) DM and GPDM estimation of eigenvalues and eigenfunctions for Legendre polynomials on the flat domain $[-1, 1]$.

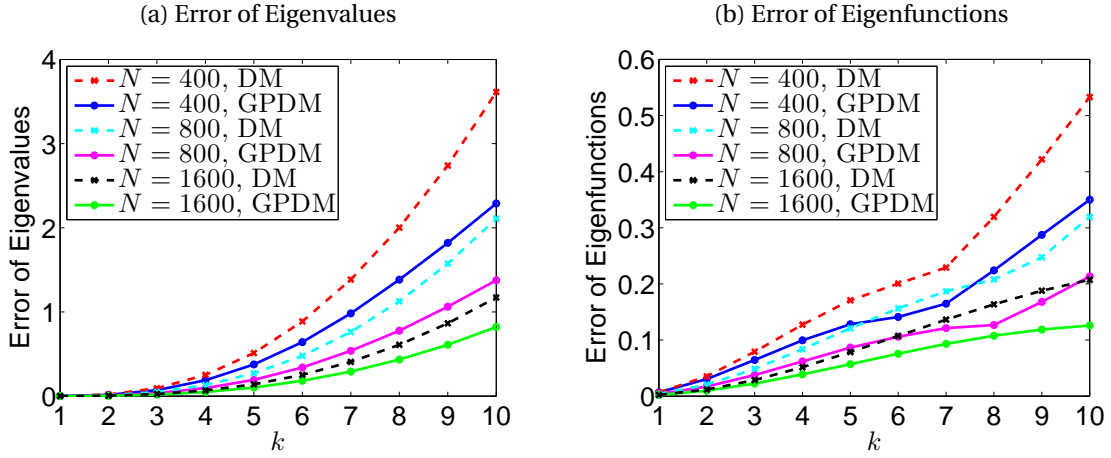


Figure 17: (Color online) Sturm-Liouville problem: Error of (a) eigenvalues and (b) eigenfunctions as functions of k for different number of points N .

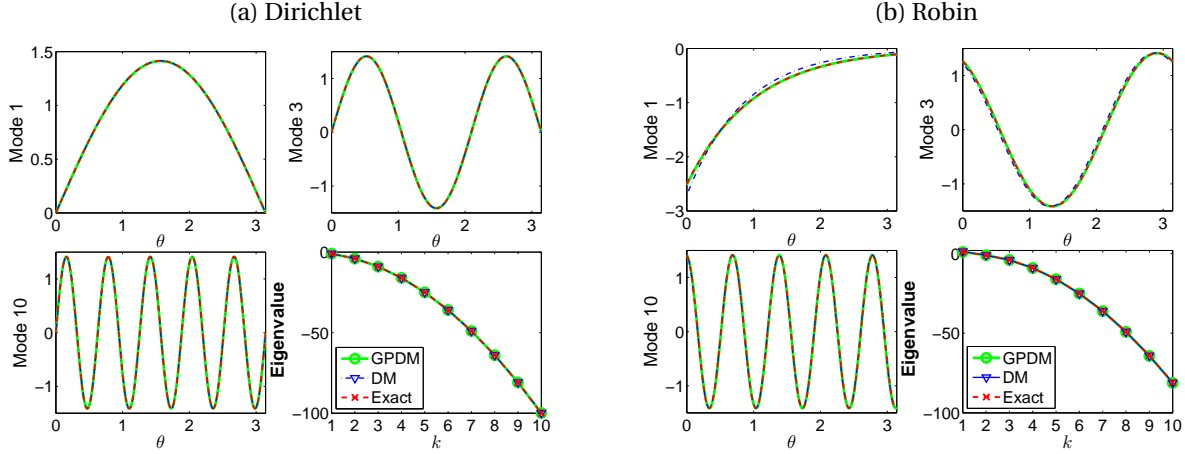


Figure 18: (Color online) Comparisons of eigenvalues and eigenfunctions between DM and GPDM for semi-circle example with (a) Dirichlet and (b) Robin boundary conditions. The Riemannian metric is given by (14) with $a = 1$.

We should point out that the Robin boundary condition at $\theta = 0$ corresponds to unphysical problems.

The Laplace-Beltrami operator \mathcal{L}_1 is numerically estimated using DM and GPDM from formula (3). We construct an $N \times N$ matrix on N equally spaced discrete grids $\{x_i = (\cos((i-1)\pi/(N-1)), \sin((i-1)\pi/(N-1)))\}_{i=1,\dots,N}$. The kernel uses $k = 50$ nearest neighbors and the bandwidth $\epsilon = 2.1 \times 10^{-5}$, that is auto tuned using a fixed (for $N = 400$) grid points for all types of boundary conditions. The numerical results are shown in Fig. 18. In these two problems, the eigenvalues and eigenfunctions can be well approximated by both DM and GPDM, although DM is less accurate for the Robin boundary condition (as seen in the estimation of mode-1).

Figs. 19 (a) and (b) show errors of the eigenvalues and eigenfunctions, respectively, as functions of k for different number of points N on a semi-circle example with Robin boundary condition. Figs. 19(c) and (d) show the errors of the eigenvalues and eigenfunctions as functions of N , respectively. It can be seen that for DM, there is no convergence in the estimation of the leading eigenvalues and eigenfunctions as N increases. In comparison, for GPDM, there is convergence in the estimation of the leading eigenvalues and eigenfunctions.

5.3 Laplace-Beltrami operator on a semi-torus

In this example, we consider solving the eigenvalue problem for $\Delta\psi_k = \lambda_k\psi_k$ on a 2D semi-torus embedded in \mathbb{R}^3 with Dirichlet and Dirichlet-Neumann mixed boundary conditions. Here, the torus is defined with the standard embedding function (19) with the Riemannian metric (20), the parameter $a = 2$, and the intrinsic coordinates (θ, ϕ) on $[0, 2\pi] \times [0, \pi]$. Then, we can check that the Laplace-Beltrami operator in the intrinsic coordinates (θ, ϕ) can be written as:

$$\Delta\psi_k = \frac{1}{(a + \cos\theta)^2} \frac{\partial^2 \psi_k}{\partial \phi^2} + \frac{\partial^2 \psi_k}{\partial \theta^2} - \frac{\sin\theta}{a + \cos\theta} \frac{\partial \psi_k}{\partial \theta} = \lambda_k \psi_k. \quad (53)$$

We can use the method of separation of variables to solve this eigenvalue problem (53), satisfying Dirichlet and the mixed boundary conditions. That is, we set $\psi_k = \Phi_k(\phi)\Theta_k(\theta)$ and substitute ψ_k back into (53) to deduce the eigenvalue problems for Φ_k and Θ_k :

$$\Phi_k'' + m_k^2 \Phi_k = 0, \quad (54)$$

$$\Theta_k'' - \frac{\sin\theta}{a + \cos\theta} \Theta_k' - \frac{m_k^2}{(a + \cos\theta)^2} \Theta_k = \lambda_k \Theta_k, \quad (55)$$

where the derivatives in (54) and (55) are taken with respect to ϕ and θ , respectively. The discrete values of m_k are chosen such that Φ_k satisfies (54) with two types of boundary conditions. In particular, type (a) is the Dirichlet boundary condition at both sides ($\Phi_k(0) = \Phi_k(\pi) = 0$) and type (b) is the Dirichlet-Neumann mixed boundary condition ($\Phi_k(0) = 0$ and $\Phi_k'(\pi) = 0$). Then, the eigenvalue problem (55) can be numerically solved for λ_k with high-order accuracy. The eigenvalue λ_k associated with the eigenfunction ψ_k obtained by the approach above are treated as the exact solutions of the eigenvalue problem (53).

In our numerical implementation, the grid points $\{\theta_i, \phi_j\}$ are uniformly distributed on $[0, 2\pi] \times [0, \pi]$, with $i, j = 1, \dots, 64$ points in each direction resulting in a total of $N = 4096$ grid points. We assume that we do not know the

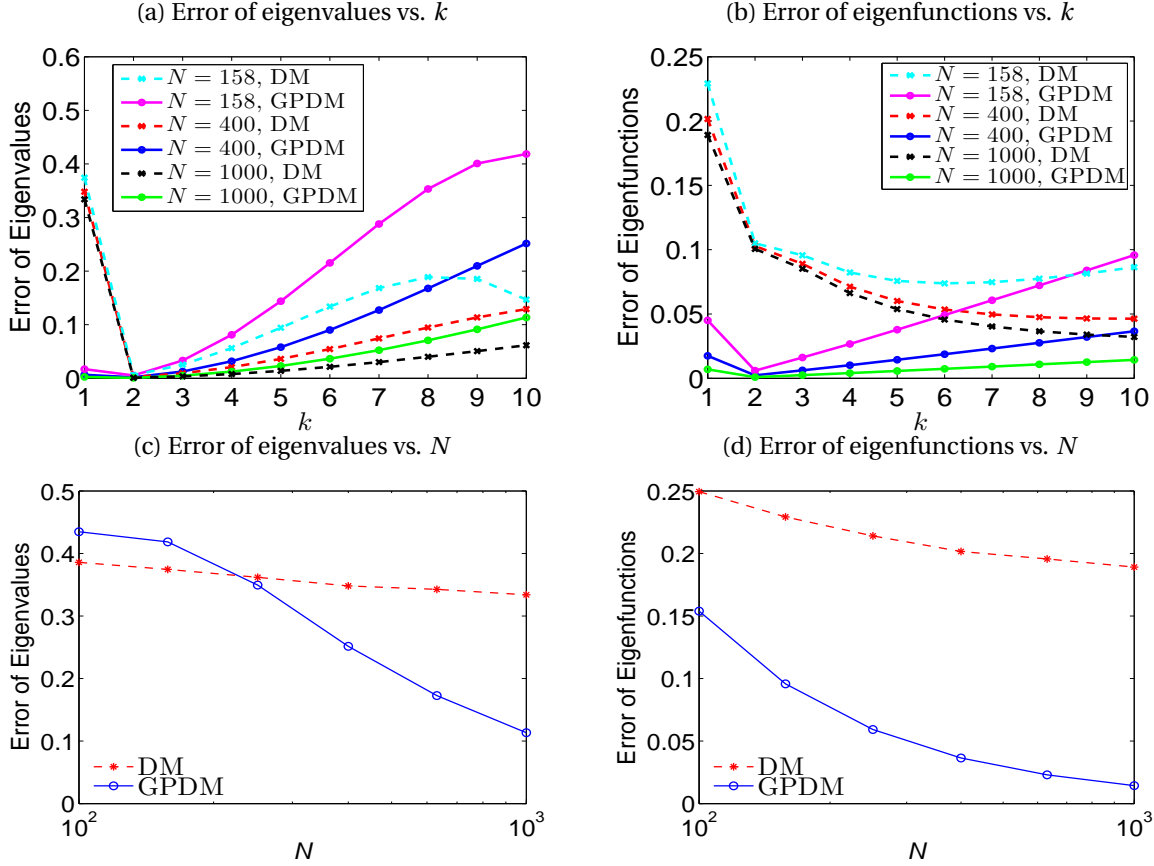


Figure 19: (Color online) Error of (a) eigenvalues and (b) eigenfunctions as functions of k for different number of points N for semi-circle example with Robin boundary condition. (c) Error of eigenvalues and (d) error of eigenfunctions vs. N . Note that for DM, there is no convergence for the leading eigenvalues and eigenfunctions.

embedding function (19) when solving the eigenvalue problem using DM and GPDM. For the GPDM method, we estimate normal direction \mathbf{v} to the boundary, add ghost points along \mathbf{v} , construct an augmented matrix using standard DM, and finally construct the $N \times N$ diffusion matrix based on the extrapolation formula and boundary conditions. We use $k = 400$ nearest neighbors to construct a sparse matrix \mathbf{L} for computational efficiency. The kernel bandwidth $\epsilon = 0.004$ is auto tuned for all types of boundary conditions.

Fig. 20 shows the numerical estimates of the first 20 eigenvalues and the 8th eigenfunction for (a) Dirichlet and (b) the mixed boundary conditions. One can see from Fig. 20 that the eigenvalues and the 8th eigenfunction can be approximated well by both DM and GPDM. For Dirichlet boundary condition, the largest errors of the first twenty eigenvalues are comparable as 0.08 and 0.12 using the standard DM and GPDM, respectively. The largest ℓ^∞ -norm error of the first twenty eigenfunctions using GPDM ($= 0.01$) is much smaller than that using DM ($= 0.31$). For the mixed boundary condition, the largest errors of the first twenty eigenvalues are comparable as 0.06 and 0.04 using the standard DM and GPDM, respectively. The largest ℓ^∞ -norm error of the first twenty eigenfunctions using GPDM ($= 0.99$) is comparable to that using DM ($= 1.03$). However, a close inspection, e.g. the 8th eigenfunctions, suggests that the GPDM errors occur on smaller regions of the domain compare to those of DM.

6 Summary

In this paper, we introduced the Ghost Points Diffusion Maps (GPDM) to estimate second-order elliptic differential operators defined on smooth manifolds with boundary. The proposed method overcomes the inconsistency of the diffusion maps (DM) algorithm in estimating these differential operators near the boundaries. We provided theoretical convergence study as well as numerical verification on test problems with tractable solutions as well as on the unknown “face” manifold to validate our claim. The key idea of GPDM is motivated by the standard ghost points approach that is used to obtain a higher-order finite-difference approximation of Neumann/Robin type

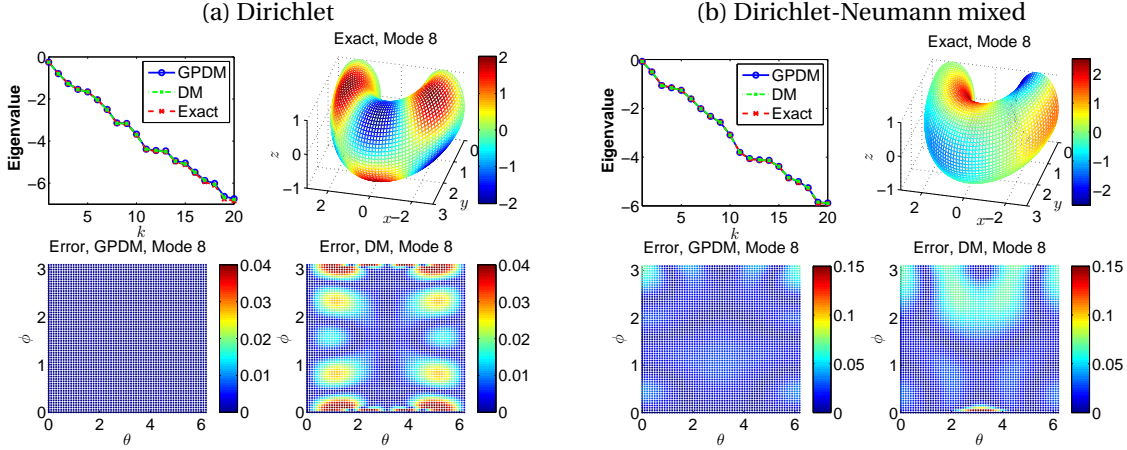


Figure 20: (Color online) Comparisons of eigenvalues and eigenfunctions with (a) Dirichlet, (b) Dirichlet-Neumann mixed boundary conditions on a semi-torus example with Riemannian metric (20) with $a = 2$.

boundary conditions. Our key contribution is to realize this idea with a concrete numerical algorithm for unknown manifolds, identified only by the point clouds, that is guaranteed to be consistent.

We considered solving elliptic PDEs (32) with the GPD operator estimation method. We showed that the PDE solver, which is a mesh-free technique, is a convergent method under the standard assumption of the well-posedness of the PDE problem. Numerically, we validated the solver on a series of 1D and 2D test examples with and without explicit solutions. On problems with unknown manifold where the explicit solution is unknown, we compare the result against the FEM solution. Overall, GPD is much more accurate and robust relative to DM except on the Neumann boundary for which DM is expected to work well as shown in [17]. Numerically, we also found that GPD is more accurate compared to DM in terms of solving eigenvalue problems associated with the operators (3)-(4).

While the proposed approach is encouraging, it also poses many open questions, namely:

- The theoretical analysis in this paper relies on the smoothness of the manifold in Assumption 3.4 and whether such assumption holds with current ghost points construction is unclear.
- Related to this issue is that the ghost points are constructed by extending points along the exterior normal direction from the boundary. Since the errors in estimating the directional derivatives and normal vectors depend on Hessian (second-order derivatives), the error can be large if the curvature is very large at the boundary. In this case, nontrivial higher-order approximation methods are needed to control these errors.
- In this work, we provided concrete algorithms for 1D and 2D manifolds. For higher-dimensional manifolds, the specification of the ghost points is a challenging problem.
- The proposed technique assumed that the boundary points are given. While this approach is suitable in applications where one can place sensors on the boundary of the domain, the minimum number of boundary points that is needed to guarantee a consistent estimator is not addressed in this paper.
- While the numerical demonstration showed convincing results in solving eigenvalue problems, spectral convergence, as well as the error estimate of the eigenfunctions, are not known. One possible avenue is to extend the result in [26] to manifolds with boundary.

Acknowledgment

The research of JH was partially supported under the NSF grant DMS-1854299. This research was supported in part by a Seed Grant award from the Institute for Computational and Data Sciences at the Pennsylvania State University. The authors also thank Faheem Gilani for providing an initial sample code of FELICITY FEM and Tyrus Berry for the helpful discussion on various aspects of differential geometry.

A Proof of Proposition 3.3

The error has two parts, one from the regression coefficients \tilde{a}^L and \tilde{a}^R , and the other from estimation of the directional derivatives $\partial_{\mathbf{v}^L} u$ and $\partial_{\mathbf{v}^R} u$. First, we estimate the error from the regression coefficients \tilde{a}^L and \tilde{a}^R . Let $\gamma_L(\ell)$ be a geodesic parameterized with the arc length ℓ , connecting the points x^B and its "left" nearest neighbor x^L such that $\gamma_L(0) = x^B$ and $\gamma_L(\ell) = x^L$. Define $\mathbf{v}^L := \gamma'_L(0) \in T_{x^B} M$ as a unit tangent vector by noticing that $|\gamma'_L(t)| \equiv 1$ for $0 \leq t \leq \ell$ due to the arc length parametrization. Following the proof in Proposition 3.1, we have the error estimate

$$|\mathbf{v}^L - \tilde{\mathbf{v}}^L| = \mathcal{O}(h).$$

Similarly, we can define the geodesic $\gamma_R(\ell)$ connecting x^B and x^R and the unit tangent vector $\mathbf{v}^R := \gamma'_R(0) \in T_{x^B} M$. Then, we have the similar error estimate

$$|\mathbf{v}^R - \tilde{\mathbf{v}}^R| = \mathcal{O}(h).$$

Since $-\mathbf{v}, \mathbf{v}^L, \mathbf{v}^R \in T_{x^B} M$ and M is a 2D manifold, there exist unique coefficients a^L and a^R such that

$$-\mathbf{v} = a^L \mathbf{v}^L + a^R \mathbf{v}^R.$$

By comparing Eq. (22) and noticing that $|\mathbf{v} - \tilde{\mathbf{v}}| = \mathcal{O}(h, \sqrt{\epsilon})$, we have the estimation for coefficients,

$$|a^L - \tilde{a}^L| = \mathcal{O}(h, \sqrt{\epsilon}) \text{ and } |a^R - \tilde{a}^R| = \mathcal{O}(h, \sqrt{\epsilon}).$$

Next, we estimate the error between the analytic directional derivative $\frac{\partial u}{\partial \mathbf{v}^L}$ and the numerical estimation $\frac{\Delta u}{\Delta \tilde{\mathbf{v}}^L} := \frac{u(x^L) - u(x^B)}{|x^L - x^B|}$. Let $\tilde{\mathbf{z}} = (z_1, \dots, z_d)$ denote the d -dimensional ($d = 2$) geodesic normal coordinate of x_L defined by an exponential map $\exp_{x^B} : T_{x^B} M \rightarrow M$ then $\tilde{\mathbf{z}}$ satisfies

$$\tilde{\mathbf{z}} = \ell \mathbf{v}^L = \ell \gamma'_L(0) \text{ and } \exp_{x^B} \tilde{\mathbf{0}} = x^B, \exp_{x^B} \tilde{\mathbf{z}} = x^L,$$

where $\ell^2 = \ell^2 |\gamma'_L(0)|^2 = |\tilde{\mathbf{z}}|^2 = \sum_{i=1}^d z_i^2$. We also define $\hat{u}(\tilde{\mathbf{z}}) := u(\exp_{x^B} \tilde{\mathbf{z}}) = u(x^L)$, such that, $\hat{u}(\tilde{\mathbf{0}}) = u(x^B)$. With this definition, we have the following Taylor's expansion,

$$\hat{u}(\tilde{\mathbf{z}}) = \hat{u}(\tilde{\mathbf{0}}) + \sum_{i=1}^d z_i \frac{\partial \hat{u}(\tilde{\mathbf{0}})}{\partial z_i} + \frac{1}{2} \sum_{i,j=1}^d z_i z_j \frac{\partial^2 \hat{u}(\tilde{\mathbf{0}})}{\partial z_i \partial z_j} + \mathcal{O}(\ell^3),$$

which is equivalent to,

$$u(x^L) = u(x^B) + \frac{\partial u}{\partial \mathbf{v}^L}(x^B) \ell + \frac{1}{2} (\mathbf{v}^L)^\top H(u(x^B)) \mathbf{v}^L \ell^2 + \mathcal{O}(\ell^3), \quad (56)$$

by noticing that $\tilde{\mathbf{z}} = \ell \mathbf{v}^L$ is the normal coordinate. This is just a Taylor expansion of function u along a geodesic $\gamma_L(\ell)$. Here, H denotes the $d \times d$ -dimensional Hessian matrix whose components are $\nabla_i \nabla_j u(x^B)$, where ∇_i denote the covariant derivative in the i th direction. Following the proof in Proposition 3.1, we have $|x^L - x^B|^{-1} = \ell^{-1} (1 + \mathcal{O}(\ell^2))$. Then, we have the error between the analytic $\frac{\partial u}{\partial \mathbf{v}^L}$ and the numerical $\frac{\Delta u}{\Delta \tilde{\mathbf{v}}^L}$

$$\begin{aligned} \frac{u(x^L) - u(x^B)}{|x^L - x^B|} &= \left(\frac{\partial u}{\partial \mathbf{v}^L}(x^B) \ell + \frac{1}{2} (\mathbf{v}^L)^\top H(u(x^B)) \mathbf{v}^L \ell^2 + \mathcal{O}(\ell^3) \right) \ell^{-1} (1 + \mathcal{O}(\ell^2)) \\ &= \frac{\partial u}{\partial \mathbf{v}^L}(x^B) + \mathcal{O}(\ell). \end{aligned}$$

One can follow the same steps and deduce for the "right" x^R ,

$$\frac{u(x^R) - u(x^B)}{|x^R - x^B|} = \frac{\partial u}{\partial \mathbf{v}^R}(x^B) + \mathcal{O}(\ell), \quad (57)$$

where we have introduced an arc-length ℓ for the geodesic distance between x^R and x^B . Since $\ell = \mathcal{O}(h)$, the remainder is of order- h .

Finally, we obtain the result:

$$\begin{aligned}
\left| \frac{\partial u}{\partial \mathbf{v}}(x^B) - \frac{\Delta u}{\Delta \tilde{\mathbf{v}}}(x^B) \right| &:= \left| \frac{\partial u}{\partial \mathbf{v}}(x^B) + \tilde{a}^L \frac{u(x^L) - u(x^B)}{|x^L - x^B|} + \tilde{a}^R \frac{u(x^R) - u(x^B)}{|x^R - x^B|} \right| \\
&\leq \left| a^L \frac{\partial u}{\partial \mathbf{v}^L}(x^B) - \tilde{a}^L \frac{u(x^L) - u(x^B)}{|x^L - x^B|} \right| + \left| a^R \frac{\partial u}{\partial \mathbf{v}^R}(x^B) - \tilde{a}^R \frac{u(x^R) - u(x^B)}{|x^R - x^B|} \right| \\
&\leq |a^L - \tilde{a}^L| \left| \frac{\partial u}{\partial \mathbf{v}^L}(x^B) \right| + |\tilde{a}^L| \left| \frac{\partial u}{\partial \mathbf{v}^L}(x^B) - \frac{u(x^L) - u(x^B)}{|x^L - x^B|} \right| \\
&\quad + |a^R - \tilde{a}^R| \left| \frac{\partial u}{\partial \mathbf{v}^R}(x^B) \right| + |\tilde{a}^R| \left| \frac{\partial u}{\partial \mathbf{v}^R}(x^B) - \frac{u(x^R) - u(x^B)}{|x^R - x^B|} \right| \\
&= \mathcal{O}(h, \sqrt{\epsilon}).
\end{aligned}$$

B Proof of Proposition 3.5

The proof here is by induction, using the Taylor expansion along the geodesic. For $k = 1$, we define $\gamma_1(s_1)$ to be the geodesic connecting $x_j^{G_1} \in \Delta M$ and $x_j^B \in \partial M$ parameterized with the arc length s_1 . Denote $\mathbf{w}_1 = \gamma_1'(0)$ as the velocity at x_j^B , where we notice that $\mathbf{w}_1 \in T_{x_j^B}(M \cup \Delta M)$ and may be different from the exterior normal direction \mathbf{v}_j that is numerically approximated by $\tilde{\mathbf{v}}_j$ defined in Section 3.1.2. Following the same argument as in the proof of Proposition 3.3, we can expand $\kappa(x_j^{G_1})$ at the boundary point x_j^B ,

$$\kappa(x_j^{G_1}) = \kappa(x_j^B) + \frac{\partial \kappa}{\partial \mathbf{w}_1}(x_j^B) s_1 + \mathcal{O}(s_1^2). \quad (58)$$

By subtracting (58) from the first equation in (26), we obtain

$$\kappa(x_j^{G_1}) - \tilde{\kappa}_{\epsilon,j}^{G_1} = s_1 \frac{\partial \kappa}{\partial \mathbf{w}_1}(x_j^B) - \tilde{d}_j \frac{\Delta \kappa}{\Delta \tilde{\mathbf{v}}_j}(x_j^B) + \mathcal{O}(s_1^2). \quad (59)$$

From Eq. (18), we have $\tilde{d}_j = s_1 + \mathcal{O}(s_1^3)$, where the distance \tilde{d}_j is numerically estimated using Eq. (24) and in this case it is simply the Euclidean distance between $x_j^B = \gamma_1(0)$ and $x_j^{G_1} = \gamma_1(s_1)$, $\tilde{d}_j = |\gamma_1(s_1) - \gamma_1(0)|$. From Proposition 3.1, we have $|\mathbf{w}_1 - \tilde{\mathbf{v}}| = \mathcal{O}(h)$, where \mathbf{w}_1 is the direction of the geodesic and $\tilde{\mathbf{v}}_j$ is the estimated exterior normal direction. Since $|\mathbf{v} - \tilde{\mathbf{v}}| = \mathcal{O}(h)$ (see Section 3.1.2), we obtain $|\mathbf{w}_1 - \mathbf{v}| = \mathcal{O}(h)$. Then, we have

$$\begin{aligned}
\left| \kappa(x_j^{G_1}) - \tilde{\kappa}_{\epsilon,j}^{G_1} \right| &\leq \left| s_1 \frac{\partial \kappa}{\partial \mathbf{w}_1} - s_1 \frac{\partial \kappa}{\partial \mathbf{v}} \right| + \left| s_1 \frac{\partial \kappa}{\partial \mathbf{v}} - \tilde{d}_j \frac{\partial \kappa}{\partial \mathbf{v}} \right| + \left| \tilde{d}_j \frac{\partial \kappa}{\partial \mathbf{v}} - \tilde{d}_j \frac{\Delta \kappa}{\Delta \tilde{\mathbf{v}}} \right| + \mathcal{O}(s_1^2), \\
&\leq Ch^2,
\end{aligned}$$

where the boundedness of the third term follows from Proposition 3.3 and we have used the fact that $s_1 < \epsilon^r = ch$ for some c . Then, we have

$$E^{G_1} := \max_j \left| \kappa(x_j^{G_1}) - \tilde{\kappa}_{\epsilon,j}^{G_1} \right| = Ch^2.$$

For $k = 2$, let us denote $\mathbf{w}_2 = \gamma_1'(s_1)$ as the velocity at $x_j^{G_1}$ of the geodesic $\gamma_1(s_1)$ and we can expand $\kappa(x_j^B)$ at the point $x_j^{G_1}$ along $\gamma_1(s_1)$,

$$\kappa(x_j^B) = \kappa(x_j^{G_1}) - \frac{\partial \kappa}{\partial \mathbf{w}_2}(x_j^{G_1}) s_1 + \mathcal{O}(s_1^2). \quad (60)$$

Let $\gamma_2(s_2)$ be the geodesic starting from $x_j^{G_1}$ to $x_j^{G_2}$ parameterized with the arc length s_2 . The velocity at $x_j^{G_1}$ along $\gamma_2(s_2)$ is denoted $\mathbf{w}_3 = \gamma_2'(0)$. We can expand $\kappa(x_j^{G_2})$ at the point $x_j^{G_1}$ along $\gamma_2(s_2)$,

$$\kappa(x_j^{G_2}) = \kappa(x_j^{G_1}) + \frac{\partial \kappa}{\partial \mathbf{w}_3}(x_j^{G_1}) s_2 + \mathcal{O}(s_2^2). \quad (61)$$

Adding (61) and (60), we obtain

$$\kappa(x_j^{G_2}) - 2\kappa(x_j^{G_1}) + \kappa(x_j^B) = \frac{\partial \kappa}{\partial \mathbf{w}_3}(x_j^{G_1}) s_2 - \frac{\partial \kappa}{\partial \mathbf{w}_2}(x_j^{G_1}) s_1 + \mathcal{O}(h^2). \quad (62)$$

For the right-hand-side of Eq. (62), we notice that $|\mathbf{w}_3 - \tilde{\mathbf{v}}| = \mathcal{O}(h)$ and $|\mathbf{w}_2 - \tilde{\mathbf{v}}| = \mathcal{O}(h)$ from Proposition 3.1, that is, the direction of the geodesic is close to the direction of the straight line, connecting the same two points. Also notice that $|s_1 - \tilde{d}_j| = \mathcal{O}(s_1^3) = \mathcal{O}(h^3)$ and $|s_2 - \tilde{d}_j| = \mathcal{O}(s_2^3) = \mathcal{O}(h^3)$. Thus, the right-hand-side of (62) is on order of h^2 ,

$$\kappa(x_j^{G_2}) - 2\kappa(x_j^{G_1}) + \kappa(x_j^B) = \left(\frac{\partial \kappa}{\partial \mathbf{w}_3}(x_j^{G_1}) s_2 - \frac{\partial \kappa}{\partial \mathbf{w}_3}(x_j^{G_1}) s_1 \right) + \left(\frac{\partial \kappa}{\partial \mathbf{w}_3}(x_j^{G_1}) s_1 - \frac{\partial \kappa}{\partial \mathbf{w}_2}(x_j^{G_1}) s_1 \right) + \mathcal{O}(h^2) = Ch^2, \quad (63)$$

for some constant C . By subtracting $\tilde{\kappa}_{\epsilon,j}^{G_2} - 2\tilde{\kappa}_{\epsilon,j}^{G_1} + \tilde{\kappa}_{\epsilon,j}^B = 0$ in (26) from Eq. (63), we obtain the following equation,

$$E^{G_2} := \max_j \left| \kappa(x_j^{G_2}) - \tilde{\kappa}_{\epsilon,j}^{G_2} \right| = E^B + 2E^{G_1} + Ch^2 = 3Ch^2,$$

where $E^B := \max_j \left| \kappa(x_j^B) - \tilde{\kappa}_{\epsilon,j}^B \right| = 0$.

By induction argument, we obtain the result $E^{G_k} := \max_j \left| \kappa(x_j^{G_k}) - \tilde{\kappa}_{\epsilon,j}^{G_k} \right| = C(k) h^2$, where the constant $C(k)$ grows exponential with the number of layer k by the discrete Gronwall lemma. We note that since the total number of layers K is always between 3–7 in our algorithm no matter how many dataset N , then $K = \mathcal{O}(1)$ and we can conclude that the overall error is $\mathcal{O}(h^2) = \mathcal{O}(\epsilon^{2r})$.

C Proof of Proposition 3.6

Before proving the main result (Proposition 3.6), we state the forward error estimates that are known from literature. Subsequently, we deduce several lemmas before proving the main result.

From previous results [14, 24, 15, 23, 18], we have the forward error estimation under these three situations: (1) on manifolds without boundary, or (2) for the test function u with Neumann boundary condition on manifold with boundary, or (3) for any u on manifold with boundary but only for points away from the boundary with distance at least $\mathcal{O}(\epsilon^r)$, $0 < r < 1/2$. For reader's convenience, we quote the following forward error estimation based on the third situation.

Lemma C.1. (Pointwise forward error estimate) Let $M \cup \Delta M$ be a smooth d -dimensional manifold embedded in \mathbb{R}^n . Here, ΔM is defined as in Assumption 3.4 and the Euclidean distance between any point in $x \in M$ and the extended boundary $\partial(M \cup \Delta M)$ is at least ϵ^r , where $0 < r < 1/2$. Let $x_i \in M$ for $i = 1, \dots, N$ and $x_j^{G_k} \in \Delta M$ for $j = 1, \dots, J, k = 1, \dots, K$ be i.i.d samples with sampling density $q \in C^3(M \cup \Delta M)$ defined with respect to the volume form inherited by the d -dimensional smooth augmented manifold $M \cup \Delta M$ from the ambient space \mathbb{R}^n . For any $u \in C^3(M \cup \Delta M)$, define a vector $\tilde{u} = (u(x_1), \dots, u(x_N), u(x_1^{G_1}), \dots, u(x_J^{G_K}))^\top \in \mathbb{R}^{\tilde{N}}$. Then for $i = 1, \dots, N$,

$$|(\mathbf{L}_{j'} \tilde{u})_i - \mathcal{L}_{j'} u(x_i)| = O\left(\epsilon, \frac{q(x_i)^{1/2}}{\sqrt{\tilde{N}} \epsilon^{2+d/4}}, \frac{|\nabla u(x_i)| q(x_i)^{-1/2}}{\sqrt{\tilde{N}} \epsilon^{1/2+d/4}}\right), \quad j' = 1, 3, \quad (64)$$

$$|(\mathbf{L}_2 \tilde{u})_i - \mathcal{L}_2 u(x_i)| = O\left(\epsilon, \frac{q(x_i)^{1/2}}{\sqrt{\tilde{N}} \epsilon^{2+d/4}}, \frac{|\nabla(\sqrt{\kappa(x_i)} u(x_i))| q(x_i)^{-1/2}}{\sqrt{\tilde{N}} \epsilon^{1/2+d/4}}\right) \quad (65)$$

in high probability. For \mathcal{L}_1 and \mathcal{L}_2 , the gradient operator is defined with respect to the Riemannian metric $g(u, v)$ for all $u, v \in T_x(M \cup \Delta M)$, inherited by M from the ambient space. For \mathcal{L}_3 , the gradient operator is defined with respect to a new metric, $\tilde{g}(u, v) := g(c^{-1/2}u, c^{-1/2}v)$ for all $u, v \in T_x(M \cup \Delta M)$, where c denotes the symmetric positive definite diffusion tensor.

We should point out that the first error-term is due to the continuous asymptotic expansion in (3), (4), and (9). The second error term is due to the estimation of the sampling density through (66), and the final error term is the bias induced by the discrete estimator.

Proof. The proofs for the cases $j' = 1$ and 3 are readily available in [14, 24, 15, 23, 18]. For $j' = 2$, the proof follows directly the steps in Appendix A of [18] with the following modification. Define a matrix $\mathbf{K}_{ij} = K(\epsilon, x_i, x_j) = \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right)$. Let

$$\hat{q}_\epsilon(x_j) := \frac{\epsilon^{d/2}}{\tilde{N}} \sum_{i=1}^{\tilde{N}} \mathbf{K}_{ji}, \quad (66)$$

as an estimator to the sampling density of the data $q(x_j)$. With this definition, we define $F_i(x_j) := \frac{K(\epsilon, x_i, x_j) \sqrt{\kappa(x_j)} u(x_j)}{\hat{q}_\epsilon(x_j)}$ and $G_i(x_j) := \frac{K(\epsilon, x_i, x_j) \sqrt{\kappa(x_j)}}{\hat{q}_\epsilon(x_j)}$. Following exactly the steps in the proof in [18] with the asymptotic expansion in (2), one obtains the error estimate in (65) for a discrete estimator that converges to $\kappa^{-1} \mathcal{L}_2$. Thus, the error for estimating \mathcal{L}_2 is no difference since the discrete estimator involves only a left multiplication by a diagonal matrix with diagonal components $\kappa(x_j)$ (which we denoted by \mathbf{S} in (13)). \square

Now, we will deduce several intermediate results that will simplify the proof of Proposition 3.6.

Lemma C.2. *Let $u \in C^3(M \cup \Delta M)$, where ΔM satisfies the Assumption 3.4 and the Euclidean distance from any point in $x \in M$ to the boundary of the extended manifold $\partial(M \cup \Delta M)$ is at least ϵ^r , where $0 < r < 1/2$ and suppose that $h := \tilde{d}_j = \mathcal{O}(\epsilon^r)$. Then, the local truncation error for all but the first equation in the extrapolation formula (28) is*

$$\begin{aligned} \left| (u(x^{G_2}) - 3u(x^{G_1})) - (\tilde{u}_\epsilon^{G_2} - 3\tilde{u}_\epsilon^{G_1}) \right| &= \mathcal{O}(\epsilon^{3r}), \\ \left| (u(x^{G_3}) - 3u(x^{G_2}) + 3u(x^{G_1})) - (\tilde{u}_\epsilon^{G_3} - 3\tilde{u}_\epsilon^{G_2} + 3\tilde{u}_\epsilon^{G_1}) \right| &= \mathcal{O}(\epsilon^{3r}), \\ \left| (u(x^{G_k}) - 3u(x^{G_{k-1}}) + 3u(x^{G_{k-2}}) - u(x^{G_{k-3}})) - (\tilde{u}_\epsilon^{G_k} - 3\tilde{u}_\epsilon^{G_{k-1}} + 3\tilde{u}_\epsilon^{G_{k-2}} - \tilde{u}_\epsilon^{G_{k-3}}) \right| &= \mathcal{O}(\epsilon^{3r}), \end{aligned} \quad (67)$$

where the sub-index j has been eliminated from (28).

Proof. Here, we only verify the first equation of (67). We present the proof for 1D case, where we have replaced $\tilde{d}_j \frac{\Delta u}{\Delta \tilde{v}_j}(x_j^B)$ in (28) with $u(x^B) - u(x_2)$. Let $\gamma_0(s_0)$ be the geodesic connecting $\gamma_0(0) = x^B \in \partial M$ and $\gamma_0(s_0) = x_2 \in M$ parameterized with the arc length s_0 . Denote $\mathbf{v} = -\gamma'_0(0)$ the unit velocity at x^B along the tangent space of $M \cup \Delta M$. We can expand $u(x_2)$ at the boundary x^B ,

$$u(x_2) = u(x^B) - s_0 \frac{\partial u}{\partial \mathbf{v}}(x^B) + \frac{1}{2} s_0^2 \mathbf{v}^\top H(u(x^B)) \mathbf{v} + \mathcal{O}(s_0^3), \quad (68)$$

where we used the expansion as in (56) with H denotes the Hessian operator.

Let $\gamma_1(s_1)$ be the geodesic connecting $\gamma_1(0) = x^{G_1} \in G^1$ and $\gamma_1(s_1) = x^B \in \partial M$ parameterized with the arc length s_1 . Notice that $\mathbf{v} = -\gamma'_1(s_1)$ is the unit velocity at x^B . Then, we can expand $u(x^{G_1})$ at the boundary point x^B as,

$$u(x^{G_1}) = u(x^B) + s_1 \frac{\partial u}{\partial \mathbf{v}}(x^B) + \frac{1}{2} s_1^2 \mathbf{v}^\top H(u(x^B)) \mathbf{v} + \mathcal{O}(s_1^3). \quad (69)$$

Adding (68) and (69), we obtain

$$u(x^{G_1}) - 2u(x^B) + u(x_2) = h^2 \mathbf{v}^\top H(u(x^B)) \mathbf{v} + \mathcal{O}(h^3), \quad (70)$$

where $h := \tilde{d}_j = s_0 + \mathcal{O}(h^3) = s_1 + \mathcal{O}(h^3)$ according to Eq. (18).

Denote $\mathbf{w}_1 = -\gamma'_1(0)$ as the unit velocity at x^{G_1} along the tangent space $\mathbf{w}_1 \in T_{x^{G_1}}(M \cup \Delta M)$. We can expand $u(x^B)$ at $x^{G_1} \in G^1$,

$$u(x^B) = u(x^{G_1}) - s_1 \frac{\partial u}{\partial \mathbf{w}_1}(x^{G_1}) + \frac{1}{2} s_1^2 \mathbf{w}_1^\top H(u(x^{G_1})) \mathbf{w}_1 + \mathcal{O}(s_1^3). \quad (71)$$

Let $\gamma_2(s_2)$ be the geodesic connecting $\gamma_2(0) = x^{G_2} \in G^2$ and $\gamma_2(s_2) = x^{G_1} \in G^1$ parameterized with the arc length s_2 . Notice that $\mathbf{w}_1 = -\gamma'_2(s_2)$ is the unit velocity at x^{G_1} . Then, we can expand $u(x^{G_2})$ at the ghost point x^{G_1} ,

$$u(x^{G_2}) = u(x^{G_1}) + s_2 \frac{\partial u}{\partial \mathbf{w}_1}(x^{G_1}) + \frac{1}{2} s_2^2 \mathbf{w}_1^\top H(u(x^{G_1})) \mathbf{w}_1 + \mathcal{O}(s_2^3). \quad (72)$$

Adding (71) and (72), we obtain

$$u(x^{G_2}) - 2u(x^{G_1}) + u(x^B) = h^2 \mathbf{w}_1^\top H(u(x^{G_1})) \mathbf{w}_1 + \mathcal{O}(h^3), \quad (73)$$

where $h := \tilde{d}_j = s_1 + \mathcal{O}(h^3) = s_2 + \mathcal{O}(h^3)$ according to Eq. (18).

Since $M \cup \Delta M$ is smooth, then any curve on the manifold is smooth, especially, we can expand the derivatives of the geodesic and obtain,

$$|\mathbf{v} - \mathbf{w}_1| = |\gamma'_1(s_1) - \gamma'_1(0)| = |\gamma''(0)| s_1 + \mathcal{O}(s_1^2) = \mathcal{O}(h). \quad (74)$$

Since $u \in C^3(M \cup \Delta M)$, then each component in the Hessian operator is $C^1(M)$ such that one can expand each component of the Hessian,

$$H(u(x^{G_1}))_{i,j} - H(u(x^B))_{i,j} = \mathcal{O}(s_1) = \mathcal{O}(h). \quad (75)$$

Subtracting (70) from (73), we obtain

$$\begin{aligned} u(x^{G_2}) - 3u(x^{G_1}) + 3u(x^B) - u(x_2) &= h^2 (\mathbf{v}^\top H(u(x^B)) \mathbf{v} - \mathbf{w}_1^\top H(u(x^{G_1})) \mathbf{w}_1) + \mathcal{O}(h^3) \\ &= h^2 \mathbf{v}^\top (H(u(x^B)) - H(u(x^{G_1}))) \mathbf{v} + \mathcal{O}(h^3) = \mathcal{O}(h^3), \end{aligned} \quad (76)$$

where we have used (74) and (75). Subtracting (28) from the above Eq. (76), we prove the first equation of (67).

For 2D case, one can follow the same argument as above, but replace $\tilde{d}_j \frac{\Delta u}{\Delta \mathbf{v}_j}(x_j^B)$ in (28) with the expression in (23) and replace the Taylor expansion in (68) with those on x_L and x_R . We can prove the other equations in (67) in a similar way. \square

Lemma C.3. *Let $u \in C^3(M \cup \Delta M)$, where ΔM satisfies the Assumption 3.4 and the Euclidean distance from any point in $x \in M$ to the extended boundary $\partial(M \cup \Delta M)$ is at least ϵ^r , where $0 < r < 1/2$. Let $\mathbf{L} = (\tilde{\mathbf{D}} - \mathbf{I})/\epsilon := (\mathbf{D}^{-1}\mathbf{W} - \mathbf{I})/\epsilon$ be the $\tilde{N} \times \tilde{N}$ matrix, where $\tilde{N} = N + JK$, obtained from the standard diffusion maps as a discrete approximation to one of the diffusion operators in (15) with the entries of $\tilde{\mathbf{D}}$ are denoted by $\tilde{D}_{i,j}$. Let $\tilde{\mathbf{u}}_\epsilon = (u_\epsilon(x_1), \dots, u_\epsilon(x_N), \tilde{u}_{\epsilon,1}^{G_1}, \dots, \tilde{u}_{\epsilon,1}^{G_K}, \dots, \tilde{u}_{\epsilon,j}^{G_1}, \dots, \tilde{u}_{\epsilon,j}^{G_K})^\top$ and $\tilde{\mathbf{u}} = (u(x_1), \dots, u(x_N), u(x_1^{G_1}), \dots, u(x_j^{G_1}), \dots, u(x_j^{G_K}))^\top$. Then, for each boundary point $x_j^B \in \partial M$, the local truncation error for the first equation in the extrapolation formula (28) is given by,*

$$\left| \sum_{j',k=1}^{J,K} \tilde{D}_{B_j, (N+(j'-1)K+k)} (u(x_{j'}^{G_k}) - \tilde{u}_{\epsilon,j'}^{G_k}) \right| = \mathcal{O} \left(\epsilon \left(\epsilon, \tilde{N}^{-1/2} \epsilon^{-(2+d/4)}, \tilde{N}^{-1/2} \epsilon^{-(1/2+d/4)} \right) \right), \quad (77)$$

where the sub-index B_j corresponds to the boundary point x_j^B .

Proof. We only consider B_j th row corresponding to the boundary point x_j^B . According to Lemma C.1, we have

$$\left| (\mathbf{L}\tilde{\mathbf{u}})_{B_j} - \mathcal{L}u(x_j^B) \right| = \mathcal{O} \left(\epsilon, \tilde{N}^{-1/2} \epsilon^{-(2+d/4)}, \tilde{N}^{-1/2} \epsilon^{-(1/2+d/4)} \right),$$

where we have suppressed the dependence on $q(x_j^B), \nabla u(x_j^B), \nabla(\kappa^{1/2}(x_j^B)u(x_j^B))$ in (64) and (65) to simplify the discussion. We also notice that the PDE equation $\mathcal{L}u(x_j^B) = f(x_j^B)$ and $(\mathbf{L}\tilde{\mathbf{u}}_\epsilon)_{B_j} = f(x_j^B)$, where the latter is the first equation of the extrapolation formula in (28). Thus, we have

$$\left| (\mathbf{L}\tilde{\mathbf{u}})_{B_j} - (\mathbf{L}\tilde{\mathbf{u}}_\epsilon)_{B_j} \right| = \mathcal{O} \left(\epsilon, \tilde{N}^{-1/2} \epsilon^{-(2+d/4)}, \tilde{N}^{-1/2} \epsilon^{-(1/2+d/4)} \right). \quad (78)$$

Since $u(x_i) = u_\epsilon(x_i)$ for $i = 1, \dots, N$ and the identity \mathbf{I} only contributes to the coefficient of $u(x_j^B) = \tilde{u}_{\epsilon,j}^{B_j}$ for the boundary point x_j^B , we can simplify the left-hand-side of (78) as,

$$\left| (\mathbf{L}\tilde{\mathbf{u}})_{B_j} - (\mathbf{L}\tilde{\mathbf{u}}_\epsilon)_{B_j} \right| = \frac{1}{\epsilon} \left| (\tilde{\mathbf{D}}\tilde{\mathbf{u}})_{B_j} - (\tilde{\mathbf{D}}\tilde{\mathbf{u}}_\epsilon)_{B_j} \right| = \frac{1}{\epsilon} \left| \sum_{j',k=1}^{J,K} \tilde{D}_{B_j, (N+(j'-1)K+k)} (u(x_{j'}^{G_k}) - \tilde{u}_{\epsilon,j'}^{G_k}) \right|. \quad (79)$$

Thus, from (78) and (79), we obtain the result in (77). \square

Remark C.4. *We should point out that for the case of approximation of \mathcal{L}_2 , it involves the integrals $G_\epsilon(\sqrt{\kappa}(x_j^B))$ and $G_\epsilon(\sqrt{\kappa}(x_j^B)u(x_j^B))$ which are numerically approximated by Monte-Carlo averages, each of which involves an averaging over the extrapolated function values, $\tilde{\kappa}_{\epsilon,j}^{G_k}$. Here,*

$$\epsilon^{d/2} G_\epsilon \sqrt{\kappa(x_i)} \approx \frac{1}{N} \sum_{j=1}^{N+JK} \mathbf{K}_{i,j} \frac{\sqrt{\kappa(x_j)}}{\mathbf{q}_j} = \frac{1}{N} \sum_{j=1}^N \mathbf{K}_{i,j} \frac{\sqrt{\kappa(x_j)}}{\mathbf{q}_j} + \frac{1}{N} \sum_{j,k=1}^{J,K} \mathbf{K}_{i, N+(j-1)K+k} \frac{\sqrt{\kappa_{\epsilon,j}^{G_k}}}{\mathbf{q}_j} + \mathcal{O}(\epsilon^{2r}),$$

where the error constant involves J, K where K is of order-1 and $J \ll N$. In this case, the error rate in (78) has an extra term of order ϵ^{2r} . If we include this error term in (78), we obtain ϵ^{2r+1} in the error bound. But since the error bound in Lemma C.2 will dominate, we ignore this term.

As for the approximation of the diffusion operator \mathcal{L}_3 , since our interest is to evaluate $\mathcal{L}_3 u(x_i)$ at the interior points x_i , numerically we only need to evaluate the kernel in (6) as,

$$\tilde{K}(\epsilon, x_i, y) := \exp\left(-\frac{(x + \epsilon B(x_i) - y)^\top C(x)^{-1} (x + \epsilon B(x_i) - y)}{2\epsilon}\right)$$

for all points $x_i \in M$, $y \in M \cup \Delta M$, and thus we don't need to extrapolate B and C on the ghost points. Thus, the error bound in (64) (and (78) above) holds.

Now, we are ready to prove Proposition 3.6. Consider (67) in Proposition C.2 and (77) in Proposition C.3, we write these equations in a matrix form,

$$\mathbf{E} \delta \tilde{u}_\epsilon^G = \mathcal{O}\left(\epsilon^{3r}, \epsilon^2, \bar{N}^{-1/2} \epsilon^{-(1+d/4)}, \bar{N}^{-1/2} \epsilon^{(1/2-d/4)}\right), \quad (80)$$

where $\delta \tilde{u}_\epsilon^G = (|\tilde{u}_{\epsilon,1}^{G_1} - u(x_1^{G_1})|, \dots, |\tilde{u}_{\epsilon,J}^{G_K} - u(x_J^{G_K})|)^\top$ and the matrix \mathbf{E} is of size $JK \times JK$. For simplicity of the discussion, we set $J = 1$ corresponds to a boundary point. One can use the same idea for the case of $J > 1$.

In this case, the matrix \mathbf{E} in (80) is given by

$$\mathbf{E} = \begin{pmatrix} \tilde{D}_{B_1,(N+1)} & \tilde{D}_{B_1,(N+2)} & \tilde{D}_{B_1,(N+3)} & \tilde{D}_{B_1,(N+4)} & \cdots & \tilde{D}_{B_1,(N+K)} \\ -3 & 1 & & & & 0 \\ 3 & -3 & 1 & & & \vdots \\ -1 & 3 & -3 & 1 & & \vdots \\ \vdots & & & & \ddots & 0 \\ 0 & \cdots & -1 & 3 & -3 & 1 \end{pmatrix}.$$

We can obtain the uniform error between \tilde{u} and \tilde{u}_ϵ in (80) once showing that $\|\mathbf{E}^{-1}\|_\infty < C$. We have the following decomposition for matrix \mathbf{E} , $\mathbf{E} = \mathbf{E}_0 + \mathbf{v}_1 \mathbf{v}_2^\top$,

$$\mathbf{E}_0 = \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ -3 & 1 & & & & 0 \\ 3 & -3 & 1 & & & \vdots \\ -1 & 3 & -3 & 1 & & \vdots \\ \vdots & & & & \ddots & 0 \\ 0 & \cdots & -1 & 3 & -3 & 1 \end{pmatrix}, \quad \mathbf{v}_1 = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ \vdots \\ \vdots \\ 0 \end{pmatrix}, \quad \mathbf{v}_2 = \begin{pmatrix} \tilde{D}_{B_1,(N+1)} - 1 \\ \tilde{D}_{B_1,(N+2)} \\ \tilde{D}_{B_1,(N+3)} \\ \vdots \\ \tilde{D}_{B_1,(N+K)} \end{pmatrix}.$$

By induction, one can show that

$$\mathbf{E}_0^{-1} = \begin{pmatrix} 1 & 0 & 0 & \cdots & \cdots & 0 \\ 3 & 1 & & & & 0 \\ 6 & 3 & 1 & & & \vdots \\ 10 & 6 & 3 & 1 & & \vdots \\ \vdots & & & & \ddots & 0 \\ K(K+1)/2 & \cdots & 10 & 6 & 3 & 1 \end{pmatrix},$$

so that $\|\mathbf{E}_0^{-1}\|_\infty = K(K+1)(K+2)/6 < C$ by noticing that K is always fixed to be about $3 \sim 7$ even when $N \rightarrow \infty$. One can calculate that $\mathbf{E}_0^{-1} \mathbf{v}_1 = (1, 3, 6, \dots, K(K+1)/2)^\top$ and $1 + \mathbf{v}_2^\top \mathbf{E}_0^{-1} \mathbf{v}_1 = \sum_{k=1}^K k(k+1) \tilde{D}_{B_1,(N+k)}/2$ which is nonzero. Thus, according to Sherman-Morrison formula, we have

$$\|\mathbf{E}^{-1}\|_\infty = \|(\mathbf{E}_0 + \mathbf{v}_1 \mathbf{v}_2^\top)^{-1}\|_\infty = \left\| \left(\mathbf{I} - \frac{\mathbf{E}_0^{-1} \mathbf{v}_1 \mathbf{v}_2^\top}{1 + \mathbf{v}_2^\top \mathbf{E}_0^{-1} \mathbf{v}_1} \right) \mathbf{E}_0^{-1} \right\|_\infty < C,$$

and the proof is complete.

D Proof of Theorem 4.1.

The proof here follows the standard approach for proving the convergence of the finite-difference method presented in many numerical PDE texts (e.g., see [37]). That is, we will show that \mathbf{L}^I in (38) satisfies a discrete maximum principle. Subsequently, a comparison function is chosen using the maximum principle of the Dirichlet PDE problem to establish the stability condition. The convergence is achieved with the consistency of the GPDM estimator in Theorem (3.7). Before we proceed with these steps, let us first analyze the resulting GPDM estimator, $(\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})\tilde{\mathbf{u}}^M + \mathbf{L}^{(2)}\tilde{\mathbf{b}}$.

To simplify the discussion, we present the 1D case with $J = 2$ boundary points, denoted by $x_1^B = x_1$ and $x_2^B = x_N$ (see Fig. 2). Replacing $-\tilde{d}_1 \frac{\Delta u}{\Delta \tilde{\mathbf{v}}_1}(x_1^B) = u(x_2) - u(x_1)$ and $-\tilde{d}_2 \frac{\Delta u}{\Delta \tilde{\mathbf{v}}_N}(x_2^B) = u(x_{N-1}) - u(x_N)$ in the second equation of (28) for $j = 1, 2$, we can write the last three equations in (28) as,

$$\begin{aligned} \tilde{u}_{\epsilon,1}^{G_k} &= \frac{k(k+1)}{2} \tilde{u}_{\epsilon,1}^{G_1} - (k^2 - 1)u_1 + \frac{k(k-1)}{2} u_2, \\ \tilde{u}_{\epsilon,2}^{G_k} &= \frac{k(k+1)}{2} \tilde{u}_{\epsilon,2}^{G_1} - (k^2 - 1)u_N + \frac{k(k-1)}{2} u_{N-1}, \end{aligned} \quad k = 2, \dots, K. \quad (81)$$

Using the same notation as in Lemma C.3, we let $\mathbf{L} = (\tilde{\mathbf{D}} - \mathbf{I})/\epsilon := (\mathbf{D}^{-1}\mathbf{W} - \mathbf{I})/\epsilon$ be the $\tilde{N} \times \tilde{N}$ matrix, where $\tilde{N} = N + 2K$, obtained from the standard diffusion maps as an discrete approximation to one of the diffusion operators in (3)-(5) with the entries of $\tilde{\mathbf{D}}$ denoted by $\tilde{D}_{i,j}$. Then, the i th component of $(\tilde{\mathbf{D}} - \mathbf{I})\tilde{\mathbf{u}}_\epsilon$ is given by,

$$\begin{aligned} \sum_{j=1}^{N+2K} \tilde{D}_{i,j} u_j - u_i &= \sum_{j=3}^{N-2} \tilde{D}_{i,j} u_j - u_i + \tilde{D}_{i,1} u_1 + \tilde{D}_{i,2} u_2 + \tilde{D}_{i,N-1} u_{N-1} + \tilde{D}_{i,N} u_N + \sum_{j,k=1}^{2,K} \tilde{D}_{i,N+(j-1)K+k} \tilde{u}_{\epsilon,j}^{G_k} \\ &= \sum_{j=3}^{N-2} \tilde{D}_{i,j} u_j - u_i + \left(\tilde{D}_{i,1} - \sum_{k=2}^K (k^2 - 1) \tilde{D}_{i,N+k} \right) u_1 + \left(\tilde{D}_{i,2} + \sum_{k=2}^K \frac{k(k-1)}{2} \tilde{D}_{i,N+k} \right) u_2 \\ &\quad + \left(\tilde{D}_{i,N-1} + \sum_{k=2}^K \frac{k(k-1)}{2} \tilde{D}_{i,N+K+k} \right) u_{N-1} + \left(\tilde{D}_{i,N} - \sum_{k=2}^K (k^2 - 1) \tilde{D}_{i,N+K+k} \right) u_N \\ &\quad + \left(\sum_{k=1}^K \frac{k(k+1)}{2} \tilde{D}_{i,N+k} \right) \tilde{u}_{\epsilon,1}^{G_1} + \left(\sum_{k=1}^K \frac{k(k+1)}{2} \tilde{D}_{i,N+K+k} \right) \tilde{u}_{\epsilon,2}^{G_1} \\ &= \sum_{j=3}^{N-2} \tilde{D}_{i,j} u_j - u_i + c_{i,1} u_1 + c_{i,2} u_2 + c_{i,N-1} u_{N-1} + c_{i,N} u_N + c_{i,0} \tilde{u}_{\epsilon,1}^{G_1} + c_{i,N+1} \tilde{u}_{\epsilon,2}^{G_1}, \end{aligned} \quad (82)$$

where we have defined,

$$\begin{aligned} c_{i,1} &= \tilde{D}_{i,1} - \sum_{k=2}^K (k^2 - 1) \tilde{D}_{i,N+k}, & c_{i,2} &= \tilde{D}_{i,2} + \sum_{k=2}^K \frac{k(k-1)}{2} \tilde{D}_{i,N+k}, \\ c_{i,N-1} &= \tilde{D}_{i,N-1} + \sum_{k=2}^K \frac{k(k-1)}{2} \tilde{D}_{i,N+K+k}, & c_{i,N} &= \tilde{D}_{i,N} - \sum_{k=2}^K (k^2 - 1) \tilde{D}_{i,N+K+k}, \\ c_{i,0} &= \sum_{k=1}^K \frac{k(k+1)}{2} \tilde{D}_{i,N+k}, & c_{i,N+1} &= \sum_{k=1}^K \frac{k(k+1)}{2} \tilde{D}_{i,N+K+k}, \end{aligned} \quad (83)$$

for convenience. From the first equation in (28), we have,

$$\begin{aligned} \sum_{j=3}^{N-2} \tilde{D}_{1,j} u_j - u_1 + c_{1,1} u_1 + c_{1,2} u_2 + c_{1,N-1} u_{N-1} + c_{1,N} u_N + c_{1,0} \tilde{u}_{\epsilon,1}^{G_1} + c_{1,N+1} \tilde{u}_{\epsilon,2}^{G_1} &= \epsilon f(x_1), \\ \sum_{j=3}^{N-2} \tilde{D}_{N,j} u_j - u_N + c_{N,1} u_1 + c_{N,2} u_2 + c_{N,N-1} u_{N-1} + c_{N,N} u_N + c_{N,0} \tilde{u}_{\epsilon,1}^{G_1} + c_{N,N+1} \tilde{u}_{\epsilon,2}^{G_1} &= \epsilon f(x_N). \end{aligned} \quad (84)$$

Since $c_{1,N+1} = c_{N,0} \approx 0$, we obtain,

$$\begin{aligned} \tilde{u}_{\epsilon,1}^{G_1} &= \frac{1}{c_{1,0}} (\epsilon f(x_1) - \sum_{j=3}^{N-2} \tilde{D}_{1,j} u_j + (1 - c_{1,1}) u_1 - c_{1,2} u_2 - c_{1,N-1} u_{N-1} - c_{1,N} u_N), \\ \tilde{u}_{\epsilon,2}^{G_1} &= \frac{1}{c_{N,N+1}} (\epsilon f(x_N) - \sum_{j=3}^{N-2} \tilde{D}_{N,j} u_j - c_{N,1} u_1 - c_{N,2} u_2 - c_{N,N-1} u_{N-1} + (1 - c_{N,N}) u_N). \end{aligned} \quad (85)$$

We should point out that Eqs. (85) and (81) are components of (29). Therefore, the i th row in (82) becomes,

$$\begin{aligned}
\sum_{j=1}^{N+2K} \tilde{D}_{i,j} u_j - u_i &= \sum_{j=3}^{N-2} \tilde{D}_{i,j} u_j - u_i + c_{i,1} u_1 + c_{i,2} u_2 + c_{i,N-1} u_{N-1} + c_{i,N} u_N \\
&\quad + \frac{c_{i,0}}{c_{1,0}} (\epsilon f(x_1) - \sum_{j=3}^{N-2} \tilde{D}_{1,j} u_j + (1 - c_{1,1}) u_1 - c_{1,2} u_2 - c_{1,N-1} u_{N-1} - c_{1,N} u_N) \\
&\quad + \frac{c_{i,N+1}}{c_{N,N+1}} (\epsilon f(x_N) - \sum_{j=3}^{N-2} \tilde{D}_{N,j} u_j - c_{N,1} u_1 - c_{N,2} u_2 - c_{N,N-1} u_{N-1} + (1 - c_{N,N}) u_N) \\
&= \sum_{j=3}^{N-2} \left(\tilde{D}_{i,j} - \frac{c_{i,0}}{c_{1,0}} \tilde{D}_{1,j} - \frac{c_{i,N+1}}{c_{N,N+1}} \tilde{D}_{N,j} \right) u_j - u_i + \left(c_{i,1} + \frac{c_{i,0}}{c_{1,0}} (1 - c_{1,1}) - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,1} \right) u_1 \\
&\quad + \left(c_{i,2} - \frac{c_{i,0}}{c_{1,0}} c_{1,2} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,2} \right) u_2 + \left(c_{i,N-1} - \frac{c_{i,0}}{c_{1,0}} c_{1,N-1} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,N-1} \right) u_{N-1} \\
&\quad + \left(c_{i,N} - \frac{c_{i,0}}{c_{1,0}} c_{1,N} + \frac{c_{i,N+1}}{c_{N,N+1}} (1 - c_{N,N}) \right) u_N + \epsilon \left(\frac{c_{i,0}}{c_{1,0}} f(x_1) + \frac{c_{i,N+1}}{c_{N,N+1}} f(x_N) \right). \tag{86}
\end{aligned}$$

It is clear that $0 < \frac{c_{i,0}}{c_{1,0}}, \frac{c_{i,N+1}}{c_{N,N+1}} < 1$, for all, $i = 2, \dots, N-1$. Also, $\tilde{D}_{i,j} > \tilde{D}_{1,j}$ and $\tilde{D}_{i,j} > \tilde{D}_{N,j}$ for $i = 2, \dots, N-1$ and $j = 3, \dots, N-2$. This implies,

$$\tilde{D}_{i,j} - \frac{c_{i,0}}{c_{1,0}} \tilde{D}_{1,j} - \frac{c_{i,N+1}}{c_{N,N+1}} \tilde{D}_{N,j} > \tilde{D}_{i,j} \left(1 - \frac{c_{i,0}}{c_{1,0}} - \frac{c_{i,N+1}}{c_{N,N+1}} \right) > 0.$$

In fact, since $c_{i,2} > c_{1,2}$ and $c_{i,2} > c_{N,2}$ for $i = 2, \dots, N-1$, it is clear that

$$c_{i,2} - \frac{c_{i,0}}{c_{1,0}} c_{1,2} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,2} > c_{i,2} \left(1 - \frac{c_{i,0}}{c_{1,0}} - \frac{c_{i,N+1}}{c_{N,N+1}} \right) > 0.$$

Likewise, we have

$$c_{i,N-1} - \frac{c_{i,0}}{c_{1,0}} c_{1,N-1} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,N-1} > c_{i,N-1} \left(1 - \frac{c_{i,0}}{c_{1,0}} - \frac{c_{i,N+1}}{c_{N,N+1}} \right) > 0.$$

The coefficients on the boundary points,

$$\begin{aligned}
c_{i,1} + \frac{c_{i,0}}{c_{1,0}} (1 - c_{1,1}) - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,1} &> c_{i,1} \left(1 - \frac{c_{i,0}}{c_{1,0}} - \frac{c_{i,N+1}}{c_{N,N+1}} \right) > 0. \\
c_{i,N} - \frac{c_{i,0}}{c_{1,0}} c_{1,N} + \frac{c_{i,N+1}}{c_{N,N+1}} (1 - c_{N,N}) &> c_{i,N} \left(1 - \frac{c_{i,0}}{c_{1,0}} - \frac{c_{i,N+1}}{c_{N,N+1}} \right) > 0
\end{aligned}$$

are also strictly positive. Thus, all of the nondiagonal coefficients of (86) are strictly positive.

We should point out that the expression on the right-hand-side of (86) is nothing but the i th row of the affine operator in (30), that is,

$$\sum_{j=1}^{N+2K} \tilde{D}_{i,j} u_j - u_i = \epsilon \left((\mathbf{L}^{(1)} + \mathbf{L}^{(2)} \mathbf{A}) \tilde{\mathbf{u}}^M + \mathbf{L}^{(2)} \tilde{\mathbf{b}} \right)_i. \tag{87}$$

Let us denote $\mathbf{M} = \epsilon(\mathbf{L}^{(1)} + \mathbf{L}^{(2)} \mathbf{A})$. Notice that if $u_i = 1$ for all $i = 1, \dots, N+2K$, then from (84) and the fact that $\sum_{j=1}^{N+2K} \tilde{D}_{i,j} = 1$, one can verify that $f(x_1) = f(x_N) = 0$, which means $(\mathbf{L}^{(2)} \tilde{\mathbf{b}})_i = 0$. Evaluating (86) at $u_i = 1$, one can see that,

$$\begin{aligned}
0 &= \sum_{j=1}^{N+2K} \tilde{D}_{i,j} - 1 = \sum_{\substack{j=3 \\ j \neq i}}^{N-2} \left(\tilde{D}_{i,j} - \frac{c_{i,0}}{c_{1,0}} \tilde{D}_{1,j} - \frac{c_{i,N+1}}{c_{N,N+1}} \tilde{D}_{N,j} \right) + \left(\tilde{D}_{i,i} - \frac{c_{i,0}}{c_{1,0}} \tilde{D}_{1,i} - \frac{c_{i,N+1}}{c_{N,N+1}} \tilde{D}_{N,i} - 1 \right) \\
&\quad + \left(c_{i,1} + \frac{c_{i,0}}{c_{1,0}} (1 - c_{1,1}) - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,1} \right) + \left(c_{i,2} - \frac{c_{i,0}}{c_{1,0}} c_{1,2} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,2} \right) \\
&\quad + \left(c_{i,N-1} - \frac{c_{i,0}}{c_{1,0}} c_{1,N-1} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,N-1} \right) + \left(c_{i,N} - \frac{c_{i,0}}{c_{1,0}} c_{1,N} + \frac{c_{i,N+1}}{c_{N,N+1}} (1 - c_{N,N}) \right) \\
&= \sum_{\substack{j=3 \\ j \neq i}}^{N-2} \mathbf{M}_{i,j} + \mathbf{M}_{i,i} + \mathbf{M}_{i,1} + \mathbf{M}_{i,2} + \mathbf{M}_{i,N-1} + \mathbf{M}_{i,N}, \tag{88}
\end{aligned}$$

where $\mathbf{M}_{i,i} < 0$ and $\mathbf{M}_{i,j} > 0$ for all $j \neq i$ are defined as in the brackets in the previous equality, respectively.

Discrete Maximum Principle: Suppose $\vec{v} = (v(x_2), \dots, v(x_{N-1}))$ is such that $\mathbf{L}^I \vec{v} > 0$. Suppose the maximum occurs at the interior point x_i , that is $v(x_i) \geq v(x_j)$ for all $j \neq i$. Then,

$$-\mathbf{M}_{i,i} v(x_i) = \sum_{\substack{j=2 \\ j \neq i}}^{N-1} \mathbf{M}_{i,j} v(x_j) - \epsilon (\mathbf{L}^I \vec{v})_i \leq \sum_{\substack{j=2 \\ j \neq i}}^{N-1} \mathbf{M}_{i,j} v(x_j) \leq \left(\sum_{\substack{j=2 \\ j \neq i}}^{N-1} \mathbf{M}_{i,j} \right) v(x_i). \quad (89)$$

Here, we use the fact that the matrix $\epsilon \mathbf{L}^I$ (as defined in (37)) is nothing but the submatrix of \mathbf{M} , ignoring the first and N th columns. From (88), $-\mathbf{M}_{i,i} = \sum_{j=1}^N \mathbf{M}_{i,j} > \sum_{j=2}^{N-1} \mathbf{M}_{i,j}$, which contradicts (89) so v cannot attain the maximum at x_i . Repeating the same argument on all interior points, it is clear that the maximum has to occur at the boundary. That is,

$$\max_{1 \leq j \leq N} v(x_j) = \{v(x_1), v(x_N)\}. \quad (90)$$

Using the same argument, one can also show that the minimum occurs at the boundaries.

Stability: By assumption, the PDE satisfies a maximum principle. Consider $v \in C^2(M)$ that solves $\mathcal{L}v(x) = C$ for all $x \in M^o$, $v(x)|_{x \in \partial M} = 0$, and a constant $C > 0$ to be determined. Here, the existence of the unique solution v follows from the well-posedness assumption of the Dirichlet problem. By the maximum principle, it is clear that $v(x) \leq 0$. Also, since M is compact, it attains the global minima on M . Define $v_s(x) := v(x) - v_{\min}$, where $v_{\min} = \min_{x \in M} v(x) \leq 0$. Thus it is clear that $0 \leq v_s(x) \leq C_2 = |v_{\min}|$ solves $\mathcal{L}v_s = C$ and $v_s(x)|_{x \in \partial M} = C_2$. In this case, since GPDM is consistent (see Theorem 3.7), it is clear that $|\mathbf{L} \vec{v}_{s,\epsilon})_i - \mathcal{L}v_s(x_i)| \leq c_1 \epsilon^{3r-1}$. Notice that,

$$|\mathcal{L}v_s(x_i) - (\mathbf{L} \vec{v}_{s,\epsilon})_i| = \left| \mathcal{L}v_s(x_i) - \left((\mathbf{L}^{(1)} + \mathbf{L}^{(2)} \mathbf{A}) \vec{v}_s^M + \mathbf{L}^{(2)} \vec{b} \right)_i \right| = \left| \mathcal{L}v_s(x_i) - (\mathbf{L}^{(2)} \vec{b})_i - (\mathbf{L}^B \vec{g} + \mathbf{L}^I \vec{v}_s^I)_i \right|, \quad (91)$$

where we have used the decomposition in (37) and the affine estimator (30). This means,

$$(\mathbf{L}^I \vec{v}_s^I)_i \geq C - c_1 \epsilon^{3r-1} - (\mathbf{L}^{(2)} \vec{b})_i - (\mathbf{L}^B \vec{g})_i.$$

Choose $C = 2 + \|\mathbf{L}^{(2)} \vec{b}\|_\infty + \|\mathbf{L}^B \vec{g}\|_\infty$, we obtain

$$(\mathbf{L}^I \vec{v}_s^I)_i \geq 2 - c_1 \epsilon^{3r-1} + \left(\|\mathbf{L}^{(2)} \vec{b}\|_\infty - (\mathbf{L}^{(2)} \vec{b})_i \right) + \left(\|\mathbf{L}^B \vec{g}\|_\infty - (\mathbf{L}^B \vec{g})_i \right) \geq 2 - c_1 \epsilon^{3r-1} \geq 0.$$

Basically $0 \leq v_s(x_i) \leq C_2$ is a comparison function that we have identified for proving the stability of the solution. Let $M = \|\vec{f}^I - \mathbf{L}^B \vec{g}\|_\infty$ be the maximum of the right-hand-side in (38), then for \hat{u}^I that solves (38), we have,

$$\mathbf{L}^I (\hat{u}^I + M \vec{v}_s^I) \geq \vec{f}^I - \mathbf{L}^B \vec{g} + (2 - c_1 \epsilon^{3r-1}) M \geq 0,$$

as $r \rightarrow 1/2$ and fixed $0 < \epsilon \ll 1$. By the discrete maximum principle in (90), it is clear that,

$$\max_{x_i \in M} \hat{u}^I \leq \max_{x_i \in M} (\hat{u}^I + M \vec{v}_s^I) \leq \max_{x_i \in \partial M} \hat{u}^B + \max_{x_i \in \partial M} M \vec{v}_s^I \leq \|\hat{u}^B\|_\infty + C_2 \|\vec{f}^I - \mathbf{L}^B \vec{g}\|_\infty.$$

Using similar argument on $-\hat{u}^I$, we obtain the stability of the approximate solution

$$\|\hat{u}^I\|_\infty \leq \|\hat{u}^B\|_\infty + C_2 \|\vec{f}^I - \mathbf{L}^B \vec{g}\|_\infty. \quad (92)$$

Convergence: Applying (92) on $\hat{u}^I - \vec{u}^I$, where components of \vec{u}^I are the true solution of the PDE in (32) with Dirichlet boundary condition, we obtain,

$$\|\hat{u}^I - \vec{u}^I\|_\infty \leq \|\hat{u}^B - \vec{u}^B\|_\infty + C_2 \|\vec{f}^I - \mathbf{L}^B \vec{g} - \mathbf{L}^I \vec{u}^B\|_\infty. \quad (93)$$

Using the same argument as in (91) and the error bound in Theorem 3.7, we immediately see the consistency of the estimator, that is,

$$\left| f(x_i) - (\mathbf{L}^{(2)} \vec{b})_i - (\mathbf{L}^B \vec{g} + \mathbf{L}^I \vec{u}^I)_i \right| = |\mathcal{L}u(x_i) - (\mathbf{L} \vec{u}_\epsilon)_i| = \mathcal{O} \left(\epsilon^{3r-1}, \bar{N}^{-1/2} \epsilon^{-(2+d/4)}, \bar{N}^{-1/2} \epsilon^{-(1/2+d/4)} \right). \quad (94)$$

Since $\vec{u}^B = \hat{u}^B = \vec{g}$, combining (93) and (94), the proof is completed.

References

- [1] Qing Han and Fanghua Lin. Elliptic partial differential equations, volume 1. American Mathematical Soc., 2011.
- [2] Dennis McLaughlin and Lloyd R Townley. A reassessment of the groundwater inverse problem. Water Resources Research, 32(5):1131–1161, 1996.
- [3] Richard P Feynman, Robert B Leighton, and Matthew Sands. The Feynman lectures on physics, Vol. I: The new millennium edition: mainly mechanics, radiation, and heat, volume 1. Basic books, 2011.
- [4] Eugene L Wachspress. Iterative solution of elliptic systems,; And applications to the neutron diffusion equations of reactor physics. Prentice-Hall, 1966.
- [5] Peter Mörters and Yuval Peres. Brownian Motion. Cambridge University Press, 2010.
- [6] Gerhard Dziuk and Charles M Elliott. Finite element methods for surface pdes. Acta Numerica, 22:289–396, 2013.
- [7] Fernando Camacho and Alan Demlow. L2 and pointwise a posteriori error estimates for fem for elliptic pdes on surfaces. IMA Journal of Numerical Analysis, 35(3):1199–1227, 2015.
- [8] Andrea Bonito, J Manuel Cascón, Khamron Mekchay, Pedro Morin, and Ricardo H Nochetto. High-order afem for the laplace–beltrami operator: Convergence rates. Foundations of Computational Mathematics, 16(6):1473–1539, 2016.
- [9] Marcelo Bertalmio, Li-Tien Cheng, Stanley Osher, and Guillermo Sapiro. Variational problems and partial differential equations on implicit surfaces. Journal of Computational Physics, 174(2):759–780, 2001.
- [10] Facundo Mémoli, Guillermo Sapiro, and Paul Thompson. Implicit brain imaging. NeuroImage, 23:S179–S188, 2004.
- [11] Steven J Ruuth and Barry Merriman. A simple embedding method for solving partial differential equations on surfaces. Journal of Computational Physics, 227(3):1943–1961, 2008.
- [12] Cécile Piret. The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces. Journal of Computational Physics, 231(14):4662–4675, 2012.
- [13] Zhen Li and Zuoqiang Shi. A convergent point integral method for isotropic elliptic equations on a point cloud. Multiscale Modeling & Simulation, 14(2):874–905, 2016.
- [14] Ronald R Coifman and Stéphane Lafon. Diffusion maps. Applied and computational harmonic analysis, 21(1):5–30, 2006.
- [15] Tyrus Berry and Timothy Sauer. Local kernels and the geometric structure of data. Applied and Computational Harmonic Analysis, 40(3):439–469, 2016.
- [16] Zhen Li, Zuoqiang Shi, and Jian Sun. Point integral method for solving poisson-type equations on manifolds from point clouds with convergence guarantees. Communications in Computational Physics, 22(1):228–258, 2017.
- [17] Faheem Gilani and John Harlim. Approximating solutions of linear elliptic pde’s on a smooth manifold using local kernel. Journal of Computational Physics, 395:563 – 582, 2019.
- [18] John Harlim, Daniel Sanz-Alonso, and Ruiyi Yang. Kernel methods for bayesian elliptic inverse problems on manifolds. arXiv preprint arXiv:1910.10669, 2019.
- [19] Zuoqiang Shi. Enforce the dirichlet boundary condition by volume constraint in point integral method. arXiv preprint arXiv:1506.02343, 2015.
- [20] Erik H Thiede, Dimitrios Giannakis, Aaron R Dinner, and Jonathan Weare. Galerkin approximation of dynamical quantities using trajectory data. The Journal of chemical physics, 150(24):244111, 2019.
- [21] Ryan Vaughn, Tyrus Berry, and Harbir Antil. Diffusion maps for embedded manifolds with boundary with applications to pdes. arXiv preprint arXiv:1912.01391, 2019.
- [22] Randall J LeVeque. Finite difference methods for ordinary and partial differential equations: steady-state and time-dependent problems, volume 98. Siam, 2007.
- [23] John Harlim. Data-driven computational methods: Parameter and Operator Estimations. Cambridge University Press, Cambridge, 2018.
- [24] Amit Singer. From graph to manifold laplacian: The convergence rate. Appl. Comp. Harmonic Anal., 21:128–134, 2006.

- [25] Tyrus Berry and John Harlim. Variable bandwidth diffusion kernels. Appl. Comput. Harmon. Anal., 40:68–96, 2016.
- [26] Nicolás García Trillos, Moritz Gerlach, Matthias Hein, and Dejan Slepčev. Error estimates for spectral convergence of the graph laplacian on random geometric graphs toward the laplace–beltrami operator. Foundations of Computational Mathematics, pages 1–61, 2019.
- [27] Tyrus Berry and Timothy Sauer. Consistent manifold representation for topological data analysis. Foundations of Data Science, 1:1, 2019.
- [28] Tyrus Berry and John Harlim. Iterated diffusion maps for feature identification. Appl. Comput. Harmon. Anal., 45(1):84–119, 2018.
- [29] Avidesh Zakhor and Gary Alvstad. Two-dimensional polynomial interpolation from nonuniform samples. IEEE Transactions on signal processing, 40(1):169–180, 1992.
- [30] Don O Loftsgaarden, Charles P Quesenberry, et al. A nonparametric estimate of a multivariate density function. The Annals of Mathematical Statistics, 36(3):1049–1051, 1965.
- [31] Tariq Aslam, Songting Luo, and Hongkai Zhao. A static pde approach for multidimensional extrapolation using fast sweeping methods. SIAM Journal on Scientific Computing, 36(6):A2907–A2928, 2014.
- [32] Bengt Fornberg, Tobin A Driscoll, Grady Wright, and Richard Charles. Observations on the behavior of radial basis function approximations near boundaries. Computers & Mathematics with Applications, 43(3-5):473–490, 2002.
- [33] J Harold Ahlberg and Edwin N Nilson. Convergence properties of the spline fit. Journal of the Society for Industrial and Applied Mathematics, 11(1):95–104, 1963.
- [34] James M Varah. A lower bound for the smallest singular value of a matrix. Linear Algebra and its Applications, 11(1):3–5, 1975.
- [35] Keenan Crane. Keenan’s 3d model repository. <http://www.cs.cmu.edu/~kmc Crane/Projects/ModelRepository>.
- [36] Shawn W Walker. Felicity: A matlab/c++ toolbox for developing finite element methods and simulation modeling. SIAM Journal on Scientific Computing, 40(2):C234–C257, 2018.
- [37] Stig Larsson and Vidar Thomée. Partial differential equations with numerical methods, volume 45. Springer Science & Business Media, 2008.