Ghost Point Diffusion Maps for Solving Elliptic PDEs on Manifolds with Classical Boundary Conditions

SHIXIAO WILLING JIANG
ShanghaiTech University

JOHN HARLIM
The Pennsylvania State University

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 boundaries 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 estimated tangential direction at the sampled points on the boundary. The resulting GPDM estimator restricts the standard DM matrix to a set of extrapolation equations that estimates the function values at the ghost points. This adjustment is analogous to the classical ghost point method in a finite-difference scheme for solving PDEs on flat domains. As opposed to the classical DM, which diverges near the boundary, the proposed GPDM estimator converges pointwise even near the boundary. Applying the consistent GPDM estimator to solve 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 numerically validate the proposed mesh-free PDE solver on various problems defined on simple submanifolds embedded in Euclidean spaces as well as on an unknown manifold. Numerically, we also found that the GPDM is more accurate compared to DM in solving elliptic eigenvalue problems on bounded smooth manifolds.

1 Introduction

Elliptic partial differential equations (PDEs) [27] arise naturally in modeling of physical phenomena, including groundwater flow [39], heat conduction [21], neutron diffusion [53], and probability theory [41]. In the manifold setting solving the PDE formulation arises in modeling of granular flow [45], liquid crystal [52], and biomembranes [20]. In computer graphics [9], PDEs on surfaces have been used to restore damaged patterns on a surface [38] and brain imaging [40], among other applications.

Communications on Pure and Applied Mathematics, 0001–0069 (PREPRINT)

Many numerical methods have been proposed to approximate the solution of PDE on the manifold setting, especially on two-dimensional surfaces. Most of these methods, however, require a parametrization of the surface, which is subsequently used to approximate the tangential derivatives along the surface. For example, the finite element method (FEM) represents the surface [11, 14, 19] using a triangular mesh. Subsequently, the PDE is solved by a Galerkin truncation on the finite-element space of functions defined on a triangular mesh. While this classical approach is popular and has been widely used in applications, it relies on the accuracy of the generated mesh. In addition to the computational task in the mesh generation, given an arbitrary set of point cloud data that lie on the manifold, constructing a regular mesh that avoids inconsistent tangential triangulation [10] can be challenging.

An alternative approach is to embed the surface PDE problem to the ambient space $\mathbb{R}^n$ such that the solution of the embedded PDE problem is consistent with the original problem when restricted to M. One realization of such an approach is to use a level set representation [9] for the surface, and subsequently, solve the embedded PDE equation in $\mathbb{R}^n$ using any standard method that is applicable on the Euclidean domain. The level set representation, unfortunately, can lead to degenerate diffusion equations, in addition to many other limitations pointed out in [46]. To combat the limitations of the level set representation, the authors in [46] introduced the closest-point representation of the surface M. We should point out that it is unclear how this method will perform if we are only given randomly sampled point cloud data since these points may not be the closest point. In their papers [43, 46], they tested their scheme on examples where either the analytical formula for the closest point is given or the surface has a triangular representation. Besides this minor technical issue, a more important problem with this class of approaches is that the computational cost scales with respect to the ambient dimension-n. This is because the embedded PDE is solved in the ambient space $\mathbb{R}^n$, which is at least one dimension more than, for example, the two-dimensional surface M.

Another class of approaches is the mesh-free radial basis function (RBF) method. While several versions of RBF solvers have been proposed [23, 44], they all require one to identify normal vectors at each point cloud and approximate the tangential derivative at each point cloud using the radial basis function interpolation method. In [23], the tangential derivatives are defined by projecting the gradient in $\mathbb{R}^n$ to the tangent space. One of the key issues with this approach is that the shape parameter of the radial basis function can be difficult to tune for high codimensional problems as pointed out in [23]. Another issue that is directly related to the work in the present paper is the erratic behavior near the boundary. As far as we know, the issue near boundary has only been studied on flat domains in $\mathbb{R}^n$ [3]. That work concluded that one can achieve highly accurate solutions by an appropriate choice of radial basis functions with sufficiently large data. However, it is unclear how to extend their approach in the context of unknown manifolds since we cannot sample more data, let alone control the size of the data. In the same paper [3], the authors also numerically demonstrated that their approach can be as effective as using the ghost

points extension. While the ghost point method is computationally straightforward on flat domains, an extension to unknown nonflat geometry is a nontrivial task. The present work will introduce a numerical scheme to realize this nontrivial task and study the convergence of the approximation when it is used with the following PDE solver.

In this paper, we consider approximating the intrinsic 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 by integral operators defined with appropriate Gaussian kernels, which is the theoretical underpinning of a popular nonlinear manifold learning algorithm known as the *diffusion maps* [15] and its local kernel variants [6]. The main advantages of this approach are that it is a consistent estimator of the intrinsic PDE problem even for a submanifold of arbitrary codimension, and it can naturally handle randomly distributed point cloud data. Computationally, this mesh-free algorithm does not require a parametrization of the manifold and/or an estimation of the normal vectors at each point cloud, one of which is essential in the existing approaches discussed in the previous paragraphs. To the authors' knowledge, the idea of using such a kind of integral operator for solving PDEs was first numerically realized by the point integral method (PIM) for solving Poisson problems [35] and isotropic elliptic equations [34]. In separate works, the same idea was realized with the diffusion maps (DM) algorithm [6] for solving elliptic PDEs associated to nonsymmetric advection-diffusion (Kolmogorov) operators associated with Itô diffusion [25] and anisotropic diffusion [29]. We should point out that despite having the same vein, DM and PIM approaches are not identical, as pointed out in [25].

On manifolds with boundaries, however, the homogeneous Neumann problem is the only natural boundary condition for the Gaussian kernel integral approximation, as noted in [15]. Furthermore, as we shall see in this paper, even if the function satisfies the homogeneous 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, 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 [35]. Another approach is to use a volume constraint [47], which is a simple version of the ghost point method that is proposed in the present paper, by setting the function values at the ghost points to be zero. In [49], 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. Recent work in [51] suggests that the diffusion maps asymptotic expansion is a consistent estimator of the Laplacian of a bounded manifold in a weak sense, and the authors devised a boundary integral estimator to specify the desired boundary conditions. All of these approaches, however, do not improve the integral approximation on the interior points near the boundary in

the pointwise sense, and it is unclear whether they can be extended to the Robin boundary condition.

In this paper, we introduce the ghost point diffusion maps (GPDM) as a consistent estimator in the sense of pointwise, complementary to the weak sense result in [51]. The GPDM modifies the DM algorithm by a novel ghost points extension scheme, generalizing the classical ghost point method on flat domains to unknown submanifolds of $\mathbb{R}^n$. For the reader's convenience, let us recall the basic idea of the ghost point method in the finite-difference setting for solving the Neumann boundary value problem: $u''(x) = f(x)$, $x \in (0,1)$, $u'(0) = u'(1) = g$. Suppose the domain is discretized as follows: $\{x_j = jh : j = 0, \ldots, N,\ h = 1/N\}$. Let U_j denotes the finite-difference approximation to the solution, $u(x_j)$. Instead of using the one-side first-order finite difference, consider a center-difference approximation for the boundary condition

$$u'(0) \approx \frac{U_1 - U_{-1}}{h} = g,$$

here we have introduced a new unknown, $U_{-1} \approx u(x_{-1})$ at a ghost point, $x_{-1} := -h \notin [0,1]$. The standard ghost point method (see, e.g., [33]) specifies this function value by an additional equation that effectively imposes the PDE at the boundary point:

$$\frac{1}{h^2}(U_{-1} - 2U_0 + U_1) \approx u''(x_0) = f(x_0). \tag{1.1}$$

Notice that the two key steps in this method, the specification of the ghost point x_{-1} and the extrapolation of the function value U_{-1}, are not immediately trivial when the manifold is not a flat geometry and unknown. In the present work, we devise an algorithm to estimate normal vectors at the boundary, which in turn, allows one to carry the two key steps above along the estimated normal vectors on each point at the boundary. The proposed method uses no information of the geometry other than the available point cloud data that are possibly randomly distributed. We show that the proposed GPDM is a pointwise convergent estimator even for points close to the boundary when the function values at the ghost points are extrapolated with a set of equations that resemble matching the second-order derivatives in addition to an equation that resembles the condition in (1.1). 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 solving eigenvalue problems.

The paper will be organized as follows. In Section 2, we provide a short review of diffusion maps and their 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 with an example, 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 several appendices.

2 Diffusion Maps and Its Extension with Local Kernels

In this section, we provide a short review of the diffusion maps algorithm [15] 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 [29] and the nonsymmetric drifted diffusions via the local kernels [6].

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

$$\begin{aligned} G_\epsilon u(x) &:= \epsilon^{-d/2} \int_M \exp\left(-\frac{|x-y|^2}{4\epsilon}\right) u(y) dV(y) \\ &= \epsilon^{-d/2} \int_{M_{\epsilon,x}} \exp\left(-\frac{|x-y|^2}{4\epsilon}\right) u(y) dV(y) + \mathcal{O}(\epsilon^2) \end{aligned} \tag{2.1}$$

is effectively a local integral operator over the ϵ^r-ball around x,

$$M_{\epsilon,x} := \{y \in M, |x-y| < \epsilon^r\}.$$

In (2.1). 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 $0 < r < 1/2$,

$$G_\epsilon u(x) = m_0 u(x) + \epsilon m_2 \big(\omega(x) u(x) + \Delta_g u(x)\big) + \mathcal{O}(\epsilon^2), \tag{2.2}$$

where m_0 and m_2 are constants that depend on the kernel, 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 $\mathbb{R}^n$. We should point out that with our choice of the exponential kernel, one can verify that $m_0 = m_2$. Based on this asymptotic expansion, one can approximate the Laplace-Beltrami operator as

$$\begin{aligned} L_{1,\epsilon} u(x) &:= \frac{(G_\epsilon 1(x))^{-1} G_\epsilon u(x) - u(x)}{\epsilon} = \Delta_g u(x) + \mathcal{O}(\epsilon) \\ &:= \mathcal{L}_1 u(x) + \mathcal{O}(\epsilon) \end{aligned} \tag{2.3}$$

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 $\kappa : M \to (0, \infty)$, one can also approximate the anisotropic diffusion operator,

$$\begin{aligned} L_{2,\epsilon}u(x) &:= \kappa(x)\frac{\left(G_\epsilon\sqrt{\kappa(x)}\right)^{-1}G_\epsilon(\sqrt{\kappa(x)}u(x)) - u(x)}{\epsilon} \\ &= \mathrm{div}_g\left(\kappa(x)\nabla_g u(x)\right) + \mathcal{O}(\epsilon) := \mathcal{L}_2 u(x) + \mathcal{O}(\epsilon), \end{aligned} \tag{2.4}$$

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 [29].

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

$$\mathcal{L}_3 u := b \cdot \nabla_g u + \frac{1}{2} c^{ij} \nabla_i \nabla_j u, \tag{2.5}$$

where ∇_i is the covariant derivative in the i^{th} direction, and $\nabla_i \nabla_j$ is the component 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 $b : M \to \mathbb{R}^d$ is the drift and the symmetric positive-definite diffusion tensor $c : M \to \mathbb{R}^{d\times d}$ is a $d \times d$ matrix-valued function, where d is the dimension of manifold M.

The operator in (2.5) can be accessed by employing the integral operator in (2.1) with the following prototypical kernel [6]:

$$K(\epsilon, x, y) := \exp\left(-\frac{(x + \epsilon B(x) - y)^\top C(x)^{-1}(x + \epsilon B(x) - y)}{2\epsilon}\right) \tag{2.6}$$

where $B : M \to \mathbb{R}^n$ and $C : M \to \mathbb{R}^{n\times n}$ are related to b and c, respectively, through a local parametrization $\iota : U \subseteq \mathbb{R}^d \to M \subseteq \mathbb{R}^n$ of the manifold M as follows:

$$B(x) = D\iota(x)b(x), \quad C(x)^{-1} = \left(D\iota(x)c(x)D\iota(x)^\top\right)^\dagger. \tag{2.7}$$

Here, the set $U \subseteq \mathbb{R}^d$ denotes a domain that contains $\iota^{-1}(x)$. Here, the notation † denotes the pseudo-inverse and the differential map

$$D\iota(x) : T_{\iota^{-1}(x)}M \subseteq \mathbb{R}^d \to T_x\mathbb{R}^n \subseteq \mathbb{R}^n$$

is an $n \times d$ matrix that is usually known as the Jacobian (or pushforward) corresponding to the map ι. Applying the integral operator in (2.1) with the prototypical kernel $K(\epsilon, x, y)$ on manifold without boundary, we obtain

$$\begin{aligned} G_{K,\epsilon}u(x) &:= \epsilon^{-d/2}\int_M K(\epsilon, x, y)u(y)dV_y \\ &= m(x)u(x) + \epsilon(\omega(x)u(x) + m(x)\mathcal{L}_3 u(x)) + O\left(\epsilon^2\right), \end{aligned} \tag{2.8}$$

where $m(x) = (2\pi)^{d/2} \det(C(x))^{1/2}$ can be approximated by $G_{K,\epsilon}1(x) = m(x) + \mathcal{O}(\epsilon)$. Employing the same algebraic manipulation as in (2.3), we obtain

$$L_{3,\epsilon}u(x) := \frac{(G_{K,\epsilon}1(x))^{-1} G_{K,\epsilon}u(x) - u(x)}{\epsilon} = \mathcal{L}_3 u(x) + \mathcal{O}(\epsilon). \tag{2.9}$$

We note that the evaluation of the prototypical kernel in (2.6) 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 (2.7).

Numerically, given a set of points in ambient coordinate $\{x_i \in M\}_{i=1}^N$, which is also referred to as the point cloud data, one can approximate the integral operator $L_{1,\epsilon}$ (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 are not necessarily uniformly distributed. In particular, the function $G_{\epsilon,q}u := G_\epsilon uq$, where G_ϵ is given in (2.1), can be approximated by the following Monte Carlo average,

$$\begin{aligned} G_{\epsilon,q}u(x_i) &= \epsilon^{-d/2} \int_M \exp\left(-\frac{|x_i - y|^2}{4\epsilon}\right) u(y)q(y)dV(y) \\ &\approx \frac{\epsilon^{-d/2}}{N} \sum_{j=1}^N \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right) u(x_j). \end{aligned}$$

Define also $q_\epsilon = G_{\epsilon,q}1$ as an estimator for the unknown sampling density q. Based on the asymptotic expansion in (2.2), one can deduce

$$\frac{G_{\epsilon,q}(q_\epsilon^{-1}) G_{\epsilon,q}(uq_\epsilon^{-1}) - u}{\epsilon} = \Delta_g u + \mathcal{O}(\epsilon).$$

Compare to (2.3), the algebraic expression above involves a "right normalization" to overcome the bias induced by nonuniform sampling density q (see [6, 15, 28] for the detailed discussion). For the nonsymmetric operator, $\mathcal{L}_3$, one can repeat the same procedure as above using the nonsymmetric kernel in (2.6) but estimate the sampling density q using the symmetric Gaussian kernel to avoid estimating the normalization factor $m(x)$ in (2.8) (see [25] for the detailed discussion).

Now we 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 entries

$$\mathbf{K}_{ij} = \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right).$$

Then, the estimated unnormalized density evaluated at x_i can be estimated by the i^{th} component of vector $\mathbf{q}$, that is, $q(x_i) \approx \mathbf{q}_i = \epsilon^{-d/2} N^{-1} \sum_{j=1}^N \mathbf{K}_{ij}$. Subsequently,

we have

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

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

$$L_{2,\epsilon} \approx \mathbf{L}_2 = \frac{1}{\epsilon}\mathbf{S}(\mathbf{D}^{-1}\mathbf{W} - \mathbf{I}). \tag{2.10}$$

We should point out that the discrete estimator converges pointwise, $\mathbf{L}_j \to \mathcal{L}_j$ (for each $j = 1, 2, 3$) in high probability [4, 25, 48]. For convenience, we state this result in Lemma A.1. For the symmetric cases, $\mathcal{L}_1$ and $\mathcal{L}_2$, the spectral convergence results are also available for closed manifolds [8, 12, 24] in L^2-sense and [13, 18] in L^∞-sense, all of which are valid in high probability.

2.1 Parameter specification

To achieve accurate estimations, one needs to specify the appropriate bandwidth parameter, ϵ. For efficient implementation, we also use k-nearest neighbor algorithm to avoid computing the distances of pair of points that are sufficiently large.

Our choice of ϵ follows the method that was originally proposed in [16]. Basically, the idea relies on the following observation:

$$\begin{aligned}S(\epsilon) &:= \frac{1}{\mathrm{Vol}(M)^2}\int_M \int_{T_xM} \exp\left(-\frac{|x - y|^2}{4\epsilon}\right) dy\, dV(x) \\ &= \frac{1}{\mathrm{Vol}\, M)^2}\int_M (4\pi\epsilon)^{d/2} dV(x) = \frac{(4\pi\epsilon)^{d/2}}{\mathrm{Vol}(M)}.\end{aligned} \tag{2.11}$$

Since S can be approximated by a Monte Carlo integral, for a fixed k, we approximate

$$S(\epsilon) \approx \frac{1}{N\,k}\sum_{i,j=1}^{N,k} \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right),$$

where $\{x_j\}_{j=1}^k$ are the k-nearest neighbors of each x_i. We choose ϵ from a domain (e.g., $[2^{-14}, 10]$ in our numerical implementation) such that $\frac{d\log(S)}{d\log\epsilon} \approx \frac{d}{2}$. Numerically, we found that the maximum slope of $\log(S)$ often coincides with $d/2$, which allows one to use the maximum value as an estimate for the intrinsic dimension d when it is not available and choose the corresponding ϵ.

For well-sampled data, we choose $k < N$ to be large enough (usually between 50 and 200, depending on the size of the data) such that ϵ is smaller than the distance between x_i and its k-neighbor, for all $i = 1, \ldots, N$. With this choice, we numerically obtain $\epsilon = \mathcal{O}(N^{-2/d})$, for $d = 1, 2$, which shows accurate estimates that converge. For randomly distributed data, we set $k = \mathcal{O}(N^{1/2})$ and obtain $\epsilon = \mathcal{O}\big(\frac{k}{N}\big)^{2/d} = \mathcal{O}(N^{-1/d})$, which yields a much larger ϵ compared to the choice in the well-sampled data, that is, $N^{-1/d} > N^{-2/d}$ for $N \gg 1$ and $d \geq 1$. It is worthwhile to point out that the scaling $\epsilon = \mathcal{O}\big(\frac{k}{N}\big)^{2/d}$, which we empirically found to produce convergence solutions in randomly distributed data (as we shall show later), has also been documented as a condition for the pointwise convergence estimate (see theorem 3.6 of [12]).

Now, let us illustrate the problem near the boundary of the asymptotic approximation of the weighted Laplacian in (2.4) with a simple example.

EXAMPLE 2.1. *In this example, we compare the DM and GPDM estimates of the differential operator $\mathcal{L}_2$ on a one-dimensional ellipse $x = (x_1, x_2) = (\cos\theta, a\sin\theta)$, defined with the Riemannian metric*

$$g = \sin^2\theta + a^2\cos^2\theta \quad \textit{for } 0 \leq \theta \leq \pi, \tag{2.12}$$

where $a = 3 > 1$. The diffusion coefficient in the weighted Laplacian (2.4) *is chosen to be $\kappa := 1.1 + x_2/a = 1.1 + \sin\theta$. In local coordinates, the diffusion operator acting on function u is given as*

$$\mathcal{L}_2 u := \operatorname{div}(\kappa\nabla u) = \frac{1}{\sqrt{|g|}}\frac{\partial}{\partial\theta}\left(\sqrt{|g|}\kappa g^{-1}\frac{\partial u}{\partial\theta}\right). \tag{2.13}$$

In Figure 2.1*, we plot the explicit equation in* (2.13) *acting on a test function $u(x) = \cos(3\theta/2 - \pi/4)$, defined on a semi-ellipse with $a = 3$ and $\theta \in [0, \pi]$ being the intrinsic coordinate. The discrete estimator $\mathbf{L}_2$ of $\mathcal{L}_2$ is constructed using $N = 400$ data points distributed at an 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.*

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 (2.3), (2.4), (2.5) defined on one and two-dimensional manifolds with boundaries. To facilitate the discussion, we use

(a) $\mathcal{L}_2 u$ (b) FE of DM=13 (upper), FE of GPDM=0.04 (bottom)

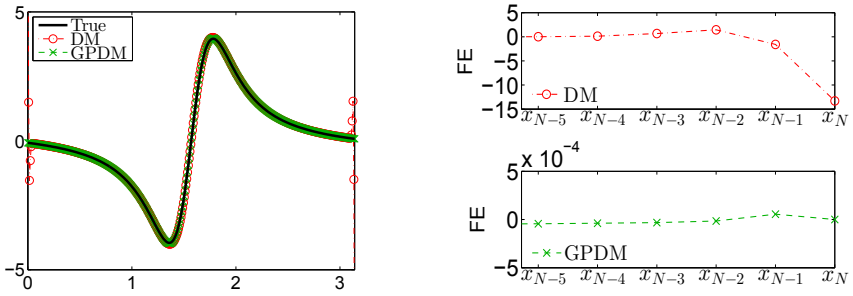


FIGURE 2.1. Numerical approximations of $\mathcal{L}_2 u$ (the weighted Laplacian in (2.4)) on a semi-ellipse example with $N = 400$. (a) Comparison of the true $\mathcal{L}_2 u$ and its DM and GPDM estimates. (b) Absolute error of DM (upper panel) and GPDM (bottom panel) near the boundary. One can see that the forward error (FE), defined as $\|\mathcal{L}_2 u - \mathbf{L}_2 u\|_\infty$ 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. In the bottom of (b), one can see that the FE is very small near the boundary for GPDM.

the conventional notations ∂M and M^o to denote the boundary and interior sets of manifold M, respectively, that satisfy $M = M^o \cup \partial M$ and $M^o \cap \partial M = \varnothing$. We assume that M is a C^∞-smooth, 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 regular curve with additional conditions (which will be clarified in Section 3.3) such that it is extendable along the boundary by a normal collar with radius $R = \mathcal{O}(\epsilon^r)$ for $0 < r < 1/2$.

The basic idea here is to follow the classical ghost point method [33] for solving the Neumann boundary condition with the finite-difference method on flat domain, as reviewed in Section 1. 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 assume that we have sample points at the boundary. For problems with unknown boundary points, one can use the tools developed in [7] to estimate points at the boundary.

We now describe the proposed algorithm, 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 vector $\boldsymbol{\nu}$ to the boundary. In Section 3.2,

we estimate the normal derivative $\partial_{\boldsymbol{\nu}} u$ at $x \in \partial M$, which will be used for specifying the boundary conditions. In Section 3.3, we describe the construction of the ghost points along the normal direction $\boldsymbol{\nu}$ 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 algebraic conditions on the ghost points, which ensures the consistency of the affine estimator of $\mathcal{L}_j$ in the limit of $\epsilon \to 0$ after $N \to \infty$, as reported in Section 3.5. Finally, we show 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 numerical methods to estimate the exterior normal direction using the point cloud data, assuming that the boundary points are given. We split the discussion into two subsections, concerning the well-sampled and randomly sampled data, as they require different algorithms.

Well-sampled data

We start our discussion on 1D manifolds. By well-sampled data, we mean that the data points are well-ordered and all consecutive points have equal (intrinsic) distance. For example, Figure 3.1(a) shows the dataset $\{x_i\}_{i=1,\dots,N}$, well-ordered on a 1D semi-ellipse with x_1 and x_N as the boundary points. Suppose that $\gamma : \mathbb{R} \to M \subseteq \mathbb{R}^n$ is a geodesic parametrization of the one-dimensional manifold M with base point $\gamma(0) = x_1 \in \partial M$ and $\gamma(s) = x_2$ (see Figure 3.1(b)). The arclength parametrization $s = \int_0^s |\gamma'(t)| dt$ is defined such that $|\gamma'(t)| = 1$ for all $t \in [0, s]$. Then, the inward unit normal direction to the boundary is given by the unit tangent vector $-\boldsymbol{\nu}_1 = \gamma'(0) \in \mathbb{R}^n$. When the parametrization γ is unknown, we can use the secant line (see Figure 3.1(c)) to estimate this normal direction $\boldsymbol{\nu}_1$ to the boundary. Specifically, the secant line approximation for $\boldsymbol{\nu}_1$ is given by

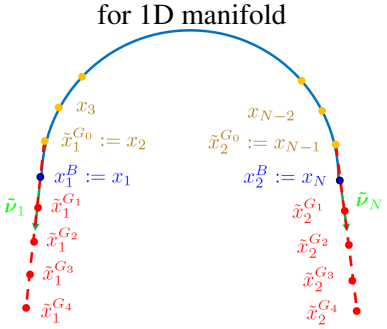
$$\widetilde{\boldsymbol{\nu}}_1 = \frac{x_1 - x_2}{|x_1 - x_2|}. \tag{3.1}$$

Likewise, one can approximate $\boldsymbol{\nu}_N$ at the other boundary point, x_N, with $\widetilde{\boldsymbol{\nu}}_N = \frac{x_N - x_{N-1}}{|x_N - x_{N-1}|}$.

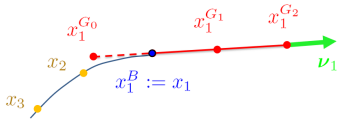
Then, the error estimate for the normal direction $\boldsymbol{\nu}_1$ to the boundary can be formalized as follows. Here, we will focus on $\boldsymbol{\nu}_1$, but this result is also valid for the secant line approximation of the tangent vectors at any $x_i \in M$, including at x_N, with appropriately defined arclength parametrization.

PROPOSITION 3.1. *Let $\gamma(s)$ be a geodesic curve parametrized with the arclength s, connecting discrete points $x_1 \in \partial M$ with $x_2 \in M$ (see Figure 3.1) such that $|x_1 - x_2| = \mathcal{O}(h)$, where $|\cdot|$ denotes the Euclidean $\mathbb{R}^n$-norm. Then, the unit tangent vector $\boldsymbol{\nu}_1 = -\gamma'(0)$ at point $x_1 = \gamma(0)$ can be estimated by $\widetilde{\boldsymbol{\nu}}_1$ in (3.1) with error $|\boldsymbol{\nu}_1 - \widetilde{\boldsymbol{\nu}}_1| = \mathcal{O}(h)$, where the constant in the error bound depends on the local curvature $\omega = |\gamma''(0)|$ of the curve at $x_1 = \gamma(0)$.*

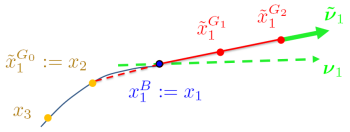
(a) secant line extension for ghost points for 1D manifold



(b) ideal construction: ghost point extension along true $\boldsymbol{\nu}_1$



(c) secant line approximation $\widetilde{\boldsymbol{\nu}}_1$



(d) exterior normal direction $\widetilde{\boldsymbol{\nu}}$ for well-sampled data

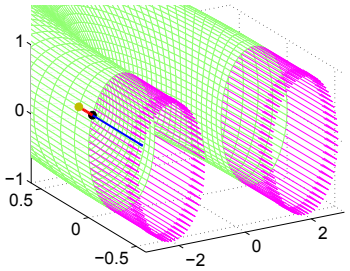


FIGURE 3.1. (a) Sketch of a specification of ghost points $\{x_j^{G_k}\}$ starting from the boundary point x_j^B, along the secant line, on a 1D manifold. Here, $\widetilde{\boldsymbol{\nu}}_1$ and $\widetilde{\boldsymbol{\nu}}_N$ are two estimated exterior normal directions that are along the secant lines connecting the boundary points and their nearest neighbors on the manifold. (b) Ideal construction: ghost point extension for $x_1^{G_0}, x_1^{G_1}, x_1^{G_2}$ along true $\boldsymbol{\nu}_1$. Here, x_2 and x_3 are points on the manifold M, and $x_1^B := x_1$ is a point on the boundary ∂M. (c) Secant line extension for ghost points $\widetilde{x}_1^{G_1}, \widetilde{x}_1^{G_2}$ along the estimated $\widetilde{\boldsymbol{\nu}}_1$. Here, $\widetilde{\boldsymbol{\nu}}_1$ is along the secant line connecting x_2 and x_1. (d) Secant line extension for well-sampled data for the torus example. Blue line is the extension of the secant line, connecting the black boundary point and the yellow manifold point, and similarly for the other magenta lines.

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,

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

Then, we have

$$\begin{aligned}\frac{\gamma(s)-\gamma(0)}{|\gamma(s)-\gamma(0)|} &= (s+\mathcal{O}(s^3))^{-1}(s\gamma'(0)+\frac{s^2}{2}\gamma''(0)+\mathcal{O}(s^3))\\ &= \gamma'(0)+\frac{s}{2}\gamma''(0)+\mathcal{O}(s^2).\end{aligned}$$

By the definitions of $\boldsymbol{\nu}_1$ and $\widetilde{\boldsymbol{\nu}}_1$ and after some algebra, we have

$$|\boldsymbol{\nu}-\widetilde{\boldsymbol{\nu}}| = \left|\gamma'(0)-\frac{\gamma(s)-\gamma(0)}{|\gamma(s)-\gamma(0)|}\right| = \frac{s}{2}|\gamma''(0)|+\mathcal{O}(s^2).$$

Since $s=\mathcal{O}(h)$, it is clear that $|\boldsymbol{\nu}_1-\widetilde{\boldsymbol{\nu}}_1|=\mathcal{O}(h)$ with a constant that depends on the curvature $\omega=|\gamma''(0)|$. □

In higher dimensions, one can use the same approximation method as above for well-sampled data. In the following example, we illustrate the secant line extension on a 2D semitorus, embedded in $\mathbb{R}^3$.

EXAMPLE 3.2. *Figure* 3.1(d) *displays the secant line extension along* $\widetilde{\boldsymbol{\nu}}$ (*magenta lines*) *for the well-sampled data on a semitorus. In this example, the semitorus is defined with the standard parametrization*:

$$x=\iota(\theta,\phi) := \begin{pmatrix}(a+\cos\theta)\cos\phi\\(a+\cos\theta)\sin\phi\\\sin\theta\end{pmatrix} \quad \textit{for} \begin{pmatrix}0\le\theta\le 2\pi\\0\le\phi\le\pi\\a=2\end{pmatrix} \tag{3.2}$$

where (θ,ϕ) *are the two intrinsic coordinates and* a *is the radius of the semitorus. The induced Riemannian metric is given by*

$$g_{(\theta,\phi)}(u,v) = u^\top\begin{pmatrix}1 & 0\\0 & (a+\cos\theta)^2\end{pmatrix}v \quad \forall u,v\in T_{(\theta,\phi)}M. \tag{3.3}$$

For well-sampled data, we notice that the two bases $\frac{\partial x}{\partial\theta}$ *and* $\frac{\partial x}{\partial\phi}$ *are perpendicular to each other. As shown in Figure* 3.1(d)*, we can extend the secant line* (*red*)*, connecting the yellow and black dots to the blue line along this estimated* $\widetilde{\boldsymbol{\nu}}$*. We apply the similar secant line extension to the other magenta lines. Then, we will add ghost points along these magenta secant lines starting from the boundary points, which will be discussed in Section* 3.3.

Unfortunately, this method is not extendable for randomly distributed data on problems of dimension $d\ge 2$ since for each boundary point we do not always sample the corresponding interior point that allows us to construct a secant line perpendicular to the boundary.

Randomly sampled data

For randomly sampled point clouds, $\{x_i\}$, that lie on a d-dimensional manifold, our basic idea here is to estimate the tangent vectors $\widetilde{\boldsymbol{t}}_1,\widetilde{\boldsymbol{t}}_2,\ldots,\widetilde{\boldsymbol{t}}_d$ that span the tangent space at each boundary point, and also estimate the tangent vectors $\widetilde{\boldsymbol{t}}^b_1,\widetilde{\boldsymbol{t}}^b_2,\ldots,\widetilde{\boldsymbol{t}}^b_{d-1}$ along the $(d-1)$-dimensional boundary ∂M. Then, we compute

the normal direction $\widetilde{\boldsymbol{\nu}}$ using the Gram–Schmidt process or QR decomposition from these directions $\{\widetilde{\boldsymbol{t}}_1, \widetilde{\boldsymbol{t}}_2, \ldots, \widetilde{\boldsymbol{t}}_d\}$ and $\{\widetilde{\boldsymbol{t}}^b_1, \ldots, \widetilde{\boldsymbol{t}}^b_{d-1}\}$. Finally, we can determine the sign of $\widetilde{\boldsymbol{\nu}}$ from the orientation of the manifold M.

To estimate these tangent vectors, we used a kernel-based weighted linear regression method as introduced in corollary 3.2 of [5]. Here, we give a quick review of the algorithm for estimating the tangent vectors for an arbitrary point x on a d-dimensional manifold M embedded in $\mathbb{R}^n$. For a point x, one defines $\mathbf{X}$ to be the $n \times K_n$ matrix with columns $\mathbf{X}_j = D(x)^{-1/2} \exp(-|x - x_j|^2/4\epsilon)(x_j - x)$ where $D(x) = \sum_{j=1}^{K_n} \exp(-|x - x_j|^2/2\epsilon)$ with x_j $(j = 1, \ldots, K_n)$ being $K_n > d$ nearest neighbors of the point x. Then, the leading largest d singular values of matrix $\mathbf{X}$ will be of order-$\sqrt{\epsilon}$ with the associated singular vectors parallel to the tangent space of M. The remaining $\min\{n, K_n\} - d$ smaller singular values will be of order-ϵ with the singular vectors orthogonal to the tangent space of M.

To simplify the discussion below, let us focus on 2D problems (while the same algorithm is applicable for any d-dimensional problems with an appropriate choice of ϵ and number of boundary points, which we shall discuss in the Summary section, Section 6). In the 2D case, we first estimate the two tangent vectors $\widetilde{\boldsymbol{t}}_1$ and $\widetilde{\boldsymbol{t}}_2$ for a boundary point $x \in \partial M$ using the kernel-based weighted regression method. We empirically choose $K_{n_1} > d = 2$ and find K_{n_1} nearest neighbors of x from points on the 2D manifold M. Using these K_{n_1} points, we then specify the bandwidth of the kernel ϵ_1 using the auto-tuned method discussed in Section 2.1. The error estimates of the two leading singular vectors $\widetilde{\boldsymbol{t}}_1$ and $\widetilde{\boldsymbol{t}}_2$ for approximating the two tangent vectors are of order-$\sqrt{\epsilon_1}$ (see appendix A in [5] for a detailed discussion). Since there are infinitely many two linearly independent vectors that can span the 2D tangent space of M at x, numerically we can only guarantee that $\mathrm{Span}\{\widetilde{\boldsymbol{t}}_1, \widetilde{\boldsymbol{t}}_2\} = \mathrm{Span}\{\frac{\partial x}{\partial \theta}, \frac{\partial x}{\partial \phi}\}$, where the parametrization $x = \iota(\theta, \phi)$ with θ and ϕ being two intrinsic coordinates. This pair of linearly independent vectors $\widetilde{\boldsymbol{t}}_1$ and $\widetilde{\boldsymbol{t}}_2$ can be different from the local bases $\frac{\partial x}{\partial \theta}$ and $\frac{\partial x}{\partial \phi}$ up to an orthonormal matrix (or a rotation).

Similarly, we apply the weighted regression method to estimate the tangent direction $\widetilde{\boldsymbol{t}} := \widetilde{\boldsymbol{t}}^b_1$ that is parallel to the boundary ∂M for each boundary point $x \in \partial M$. We empirically choose $K_{n_2} > d = 2$ and find K_{n_2} nearest neighbors of x only from boundary points of the one-dimensional ∂M. Using these K_{n_2} points, we auto-tune the bandwidth of the kernel ϵ_2. We can compute $\widetilde{\boldsymbol{t}}$ from the first singular value of this $\mathbf{X}$ and the error estimate of $\widetilde{\boldsymbol{t}}$ is of order-$\sqrt{\epsilon_2}$. Next, the normal direction $\widetilde{\boldsymbol{\nu}}$ can be approximated by subtracting the orthogonal projection of $\widetilde{\boldsymbol{t}}_1$ (or $\widetilde{\boldsymbol{t}}_2$) onto $\widetilde{\boldsymbol{t}}$ from the tangent vector $\widetilde{\boldsymbol{t}}_1$ (or $\widetilde{\boldsymbol{t}}_2$) using the Gram–Schmidt process or QR decomposition,

$$\widetilde{\boldsymbol{\nu}} = \widetilde{\boldsymbol{t}}_1 - \langle \widetilde{\boldsymbol{t}}_1, \widetilde{\boldsymbol{t}} \rangle \widetilde{\boldsymbol{t}},$$

where $\langle \widetilde{\boldsymbol{t}}_1, \widetilde{\boldsymbol{t}} \rangle$ denotes the inner product of vectors $\widetilde{\boldsymbol{t}}_1, \widetilde{\boldsymbol{t}} \in \mathbb{R}^n$, and we notice that $|\widetilde{\boldsymbol{t}}| = 1$ for a singular vector from SVD. Finally, the sign of $\widetilde{\boldsymbol{\nu}}$ can be determined

by comparing with the k-nearest neighbors of x. The error estimate for the normal direction $\widetilde{\boldsymbol{\nu}}$ is thereafter $\mathcal{O}(\sqrt{\epsilon_1}, \sqrt{\epsilon_2})$, that is, $|\boldsymbol{\nu} - \widetilde{\boldsymbol{\nu}}| = \mathcal{O}(\sqrt{\epsilon_1}, \sqrt{\epsilon_2})$.

Applying the ϵ auto-tuning algorithm discussed in Section 2, we obtain an error of order-$N^{-1/d}$, which we have verified for problems of dimensions $d = 1, 2$. For 2D problems, if the number of points at the boundary is $J = \mathcal{O}(N^{1/2})$, then $\epsilon_2 \sim J^{-1} \sim N^{-1/2}$ and this error rate balances with the rate $\epsilon_1 \sim N^{-1/2}$, which is also the rate of the error for the overall GPDM algorithm, as we show in the following example.

EXAMPLE 3.3. *Figure* 3.2(b) *displays a comparison of the true* $\boldsymbol{\nu}$ *and estimated* $\widetilde{\boldsymbol{\nu}}$ *for random data on a semitorus. The embedding function is given by* (3.2) *and the Riemannian metric is given by* (3.3). *It can be seen from Figure* 3.2(c) *that the error rate for* $|\widetilde{\boldsymbol{\nu}} - \boldsymbol{\nu}|$ *is as expected to be* $\mathcal{O}(\epsilon^{1/2})$, *where* $\epsilon = \epsilon_1$ *is chosen to be the same as that in the DM or GPDM method in Example* 4.2.

3.2 Estimation of the normal derivatives on the boundaries and distance h among neighboring ghost points

For each point x^B at the boundary, we denote $\boldsymbol{\nu} := \boldsymbol{\nu}_{x^B} \in \mathbb{R}^n$ as the corresponding normal unit vector that is pointing outward from the manifold M. We approximate the directional derivative of $\partial_{\boldsymbol{\nu}} u(x^B)$ with the following finite-difference method,

$$\frac{\partial u}{\partial \boldsymbol{\nu}}(x^B) \approx \frac{u(x^B) - u(x^{G_0})}{|x^B - x^{G_0}|}, \tag{3.4}$$

where we have defined a ghost point along $-\boldsymbol{\nu}$ (see Figure 3.1(b)) as

$$x^{G_0} := x^B - h\boldsymbol{\nu}, \tag{3.5}$$

where h characterizes the distance between neighboring ghost points as will be specified below after Definition 3.4. Let $\gamma : \mathbb{R} \to M \subseteq \mathbb{R}^n$ be a geodesic, parametrized with arclength h, such that $\gamma(0) = x^B$ and $\gamma'(0) = -\boldsymbol{\nu}$. One can see that $\gamma(h) - x^{G_0} = \gamma(h) - (\gamma(0) + h\gamma'(0)) = \mathcal{O}(h^2)$ (see Figure 3.2(a) for a geometric illustration of the point, $\gamma(h)$). For $u \in C^1$ on a straight line connecting x^{G_0} and $\gamma(h) := \exp_{x^B}(-h\boldsymbol{\nu}) \in \mathbb{R}^n$, we have $u(\gamma(h)) - u(x^{G_0}) = \mathcal{O}(h^2)$. This yields the following error estimate:

$$\begin{aligned} \frac{u(x^B) - u(x^{G_0})}{|x^B - x^{G_0}|} &= \frac{1}{h}\big(u(x^B) - u(\gamma(h))\big) + \frac{1}{h}\big(u(\gamma(h)) - u(x^{G_0})\big) \\ &= -\nabla_g u(x^B) \cdot \gamma'(0) + \mathcal{O}(h) = \nabla_g u(x^B) \cdot \boldsymbol{\nu} + \mathcal{O}(h). \end{aligned} \tag{3.6}$$

Since $\boldsymbol{\nu}$ is numerically estimated by $\widetilde{\boldsymbol{\nu}}$ with error of order-$\sqrt{\epsilon}$ and (3.5) is estimated by

$$\widetilde{x}^{G_0} := x^B - h\widetilde{\boldsymbol{\nu}}, \tag{3.7}$$

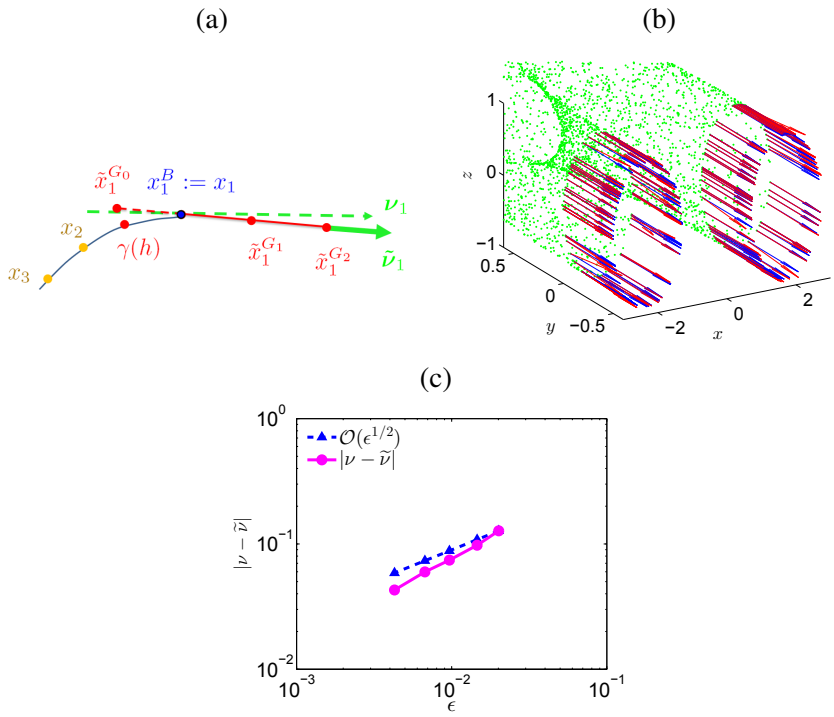


FIGURE 3.2. (a) Sketch for a ghost point extension along the estimated $\tilde{\boldsymbol{\nu}}$ for random data. Here, x_2 and x_3 are points on the manifold M, x_1^B is the point on the boundary ∂M, $\boldsymbol{\nu}_1$ is the true exterior normal direction, $\tilde{\boldsymbol{\nu}}_1$ is estimated normal direction, $\tilde{x}_1^{G_0}$ is the interior ghost point and $\gamma(h) := \exp_{x_1^B}(-h\boldsymbol{\nu}_1)$ is a projected point on the manifold M, and $\tilde{x}_1^{G_1}$ and $\tilde{x}_1^{G_2}$ are ghost points along $\tilde{\boldsymbol{\nu}}_1$. (b) Comparison between exact exterior normal direction $\boldsymbol{\nu}$ (blue arrows) and estimated exterior normal direction $\tilde{\boldsymbol{\nu}}$ (red arrows) for given random points on the semitorus (3.2) with unknown parametrization for one trial when $N = 64^2$. (c) The expectation of the error $|\boldsymbol{\nu} - \tilde{\boldsymbol{\nu}}|$ as a function of ϵ, where $\epsilon = \epsilon_1$ is chosen the same as that in DM and GPDM methods in Example 4.2. The five points correspond to $N = 32^2, 45^2, 64^2, 90^2, 128^2$, and larger N corresponds to smaller auto-tuned ϵ. For each N, we run 16 independent trials and then calculate the mean of $|\boldsymbol{\nu} - \tilde{\boldsymbol{\nu}}|$ versus the mean of auto-tuned ϵ's as one point in panel (c).

then it is immediately clear that $|\tilde{x}^{G_0} - x^{G_0}| = h|\tilde{\boldsymbol{\nu}} - \boldsymbol{\nu}| = \mathcal{O}(h\sqrt{\epsilon})$. If $u \in C^1$ on a straight line connecting $\tilde{x}^{G_0}$ and x^{G_0}, we have $u(x^{G_0}) - u(\tilde{x}^{G_0}) = \mathcal{O}(h\sqrt{\epsilon})$,

and

$$\frac{u(x^B) - u(\widetilde{x}^{G_0})}{\left|x^B - \widetilde{x}^{G_0}\right|} = \frac{1}{h}(u(x^B) - u(x^{G_0})) + \frac{1}{h}\big(u(x^{G_0}) - u(\widetilde{x}^{G_0})\big)$$

$$= \nabla_g u(x^B)\cdot \boldsymbol{\nu} + \mathcal{O}(h, \sqrt{\epsilon}) = \frac{\partial u}{\partial \boldsymbol{\nu}}(x^B) + \mathcal{O}(h, \sqrt{\epsilon}), \tag{3.8}$$

where the first term follows directly from (3.6). Based on this observation, we assume that $u \in C^1$ (in fact, C^3 for the extrapolation scheme in Section 3.4) on the set:

DEFINITION 3.4. $B_{\epsilon^r}(\partial M) := \bigcup_{x\in\partial M} B_{\epsilon^r}(x)$, where $B_{\epsilon^r}(x) = \{y \in \mathbb{R}^n : |x - y| \le \epsilon^r\}$ is an ϵ^r-ball in $\mathbb{R}^n$.

With this assumption, for $h \lesssim \mathcal{O}(\epsilon^r)$, it is clear that $x^{G_0}, \widetilde{x}^{G_0}, \gamma(h) \in B_{\epsilon^r}(x^B) \subset \mathbb{R}^n$, which justifies the use of the Taylor expansions along straight paths between these points.

Well-sampled data: In this case, since $\widetilde{\boldsymbol{\nu}}$ is a secant-line approximation, the estimated point $\widetilde{x}^{G_0}$ coincides with the interior point adjacent to the corresponding boundary point (e.g., in Figure 3.1(c), $\widetilde{x}^{G_0}$ is exactly x_2 when the boundary point $x^B = x_1$). In such a case, one can immediately set $h := |\widetilde{x}^{G_0} - x^B|$, which is also used in the first equality in (3.8). This specification scales as $h = \mathcal{O}(N^{-1/d})$.

Randomly sampled data: In such a case, generally, $\widetilde{x}^{G_0}$ does not coincide with any other randomly sampled data (see Figure 3.2(a)). To use the estimator in (3.7), one has to specify h. In our implementation, h is estimated by the mean distance from x^B to its P (around 10 in our numerical examples) nearest neighbors. Let $x_p^B \in M$ denotes the p^{th} nearest neighbor of x^B for $p = 1, \ldots, P$. Since the distance to the nearest neighbor is a density estimator [36], that is, $|x^B - x_p^B| \propto q(x^B)^{-1/d}$, where q denotes the sampling density and d denotes the dimension of the manifold M, then we specify

$$h = \frac{1}{P}\sum_{p=1}^{P} \left|x^B - x_p^B\right| = \mathcal{O}\big(q(x^B)^{-\frac{1}{2}}\big),$$

for two-dimensional manifolds.

3.3 Ghost points

It is well known that the diffusion operators defined in (2.3)–(2.5) cannot be approximated accurately near the boundary of the manifold using the standard diffusion maps algorithm. This issue is because the asymptotic expansion (2.2) is valid only for points $x \in M$ whose Euclidean 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 (2.2). Geometrically, the local integral is inaccessible if there are no available data beyond M (see Figure 3.3).

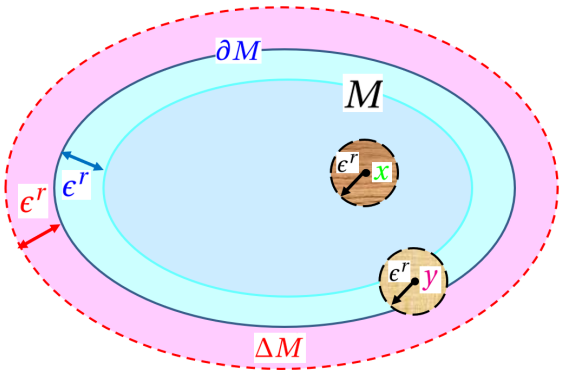


FIGURE 3.3. Sketch of the extended manifold $M \cup \Delta M$.

To address this issue, our 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 (2.1), we will devise a numerical 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 Figure 3.3). Specifically, the ghost points will be sampled from the outer normal collar that can be attached at the boundary such that the extended manifold can be isometrically embedded in $\mathbb{R}^n$ without changing the embedding function of the original M. In the following lemma, we provide the conditions for such a requirement to hold for two-dimensional manifolds.

LEMMA 3.5. *Let M be a two-dimensional Riemannian manifold with nonempty, smooth boundary ∂M, isometrically embedded in $\mathbb{R}^n$. Suppose that ∂M is a regular curve with maximum curvature of $1/R$ and any two points $x, y \in \partial M$, whose geodesic distance $d_g(x, y) > \pi R$, have Euclidean distance $|x - y| > 2R$. Then there exists a submanifold, ΔM (an outer normal collar of radius R), such that the adjunction space, $M \cup_{\mathrm{id}} \Delta M$, defined by attaching M and ΔM along the boundary with an identity "gluing" function $\mathrm{id} : \partial M \to \partial(\Delta M)$, can be isometrically embedded in $\mathbb{R}^n$ with an embedding function that is consistent with the original embedding function when it is restricted to M.*

PROOF. Our construction is to extend M with an exterior collar of radius R along the boundary. By the collar neighborhood theorem (theorem 9.25 in [31]), there exists a normal collar neighbor $W \subseteq M$, which is defined as the range of the following map $\phi : [0, R) \times \partial M \to W \subseteq M$,

$$\phi(t, x) = \exp_x(-t\, \boldsymbol{\nu}_x), \quad t \in [0, R),$$

for some $R > 0$. Here, $\boldsymbol{\nu}_x$ denotes the normal vector at $x \in \partial M$ that is pointing outward from the manifold M, so ϕ maps the points in the inward normal collar to

the collar neighbor W. Define a manifold corresponding to the pre-image of the collar neighbor that points outward as

$$\Delta M = \big\{(-t, x) : t \in [0, R), x \in \partial M, \phi(t, x) \in W\big\}, \tag{3.9}$$

which is the outer normal collar of radius R.

Next, we attach M and ΔM along the boundary by identifying $x \in \partial M$ with the identity map $\mathrm{id}(x) \in \{0\} \times \partial M \subset \Delta M$. Let $M \cup_{\mathrm{id}} \Delta M := (M \sqcup \Delta M)/\sim$ be the adjunction set defined as the quotient space of the disjoint sum induced by attaching ΔM to M along the identity map, id. Since ΔM is the outward normal collar and ∂M is a smooth boundary with bounded curvature, the adjunction set is smooth along the attached boundary ∂M. To finish the proof, we need to show that the adjunction set can be isometrically embedded in $\mathbb{R}^n$ with an embedding function that is consistent when restricted to the original manifold M.

Let $\mathcal{E} \subseteq T\mathbb{R}^n$ denote the domain of the exponential map of $\mathbb{R}^n$ and $N(\partial M)$ denote the normal bundle of ∂M in $\mathbb{R}^n$. Then we can define $E : \mathcal{E} \cap N(\partial M) \to \mathbb{R}^n$ to be the normal exponential map of ∂M in $\mathbb{R}^n$. By the tubular neighborhood theorem (see theorem 5.25 in [32]), ∂M has a uniform tubular neighbor in $\mathbb{R}^n$. Specifically, there exists a normal neighborhood of ∂M, $U \subset \mathbb{R}^n$, that is diffeomorphic under E to an open subset $V \subseteq \mathcal{E} \cap N(\partial M)$. Since $\boldsymbol{\nu}_x \in N_x(\partial M)$ and ∂M has maximum curvature $1/R$ and any two points with geodesic distance larger than one half of the circumference of the osculating circle of radius R, that is, $d_g(x, y) > \pi R$, have Euclidean distance $|x - y| > 2R$, then the open neighbor U is a tubular neighborhood of radius R that is homeomorphic to $\Delta M \subset V$. So, the tubular neighbor theorem ensures that ΔM as defined in (3.9) can be smoothly embedded in $\mathbb{R}^n$. In fact, one can define a Riemannian metric for ΔM to be the pullback of the following embedding function,

$$\widetilde{\iota}(-t, x) := x + t\,\boldsymbol{\nu}_x, \tag{3.10}$$

for any $(-t, x) \in \Delta M$. Since the induced metric of ΔM is consistent with that of M at the attached boundary, that is, $\widetilde{\iota}(0, x) = x \in \mathbb{R}^n$, then the extended manifold $M \cup_{\mathrm{id}} \Delta M$ is isometrically embedded in $\mathbb{R}^n$. □

In the remainder of this paper, we will refer to the extended manifold $M \cup_{\mathrm{id}} \Delta M$ as the set $M \cup \Delta M$ to simplify the notation. We should also point out that for the 1D manifold, since the boundary consists of only two points (e.g., as shown in Figure 3.1), the assumption for ∂M in Lemma 3.5 is slightly different. In this case, the extended manifold can be isometrically embedded as long as the two exterior normal lines of length $R > 0$ from the boundary do not intersect. Next, we will use the embedding function in (3.10) to specify the ghost points with $R = \mathcal{O}(\epsilon^r)$.

Numerically, for each boundary point $x^B \in \partial M \subset \mathbb{R}^n$, let $\widetilde{\boldsymbol{\nu}} \in \mathbb{R}^n$ be the numerical estimate of the corresponding normal vector $\boldsymbol{\nu} \in \mathbb{R}^n$ at x^B. Then, the ghost points,

$$x^{G_k} := x^B + kh\boldsymbol{\nu}, \tag{3.11}$$

are approximated by

$$\widetilde{x}^{G_k} := x^B + kh\widetilde{\boldsymbol{\nu}}, \tag{3.12}$$

for $k = 1, \ldots, K$, where $K = \mathcal{O}(\epsilon^r h^{-1})$. Numerically, however, we specify K empirically (usually $K \leq 10$). See Figures 3.1(b),(c) and 3.2(a) for a geometric illustration.

By the construction above, the Euclidean distance between any point $x \in M^o$ and $\partial(M \cup \Delta M)$ is at least of order ϵ^r. This ensures the validity of the asymptotic expansion in (2.2) for all points on M^o (including points that are close to the boundary ∂M). It is worthwhile to point out that the ghost points $\widetilde{x}^{G_k}$ do not exactly lie on ΔM since the true normal vectors, $\boldsymbol{\nu}$, are not available (ideal case as illustrated in Figure 3.1(b)). In the next section, we will show how this error affects the overall algorithm, especially when the data are randomly distributed.

3.4 Extrapolation of functions on the ghost points

We now address the extrapolation problem on the estimated ghost points. In particular, we need to extrapolate the solution u on the estimated ghost points. Popular extrapolation techniques include the linear and quadratic extrapolation methods, the level set method, and the ghost fluid method [2]. 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 analogue of matching the second-order derivatives of the function evaluated at the ghost points as the extrapolation condition, which mimics the cubic spline condition proposed in [22]. In addition, we also include a condition that mimics the classical finite-difference solution of Neumann (or Robin) boundary value problems with ghost points.

Let $u \in C^3(M \cup B_{\epsilon^r}(\partial M))$, where the set $B_{\epsilon^r}(\partial M) \subset \mathbb{R}^n$ is stated in the Definition 3.4. We note that the numerically estimated ghost points are components of this set,

$$\{\widetilde{x}_j^{G_0}\}_{j=1}^{J} \cup \{\widetilde{x}_j^{G_k}\}_{j,k=1}^{J,K} \subset B_{\epsilon^r}(\partial M).$$

Given the function values $u(x_i)$ at $x_i \in M$ and $u(\widetilde{x}_j^{G_0})$, our goal is to extrapolate u onto the set of ghost points, $\{\widetilde{x}_j^{G_k}\}_{j,k=1}^{J,K}$. In the PDE applications, the function values at $\{\widetilde{x}_j^{G_0}\}$ will be estimated in the same manner as the other data $\{x_i\}$ that lie on the manifold, that is, by inverting the discrete approximation of the diffusion operators. In particular, we define the matrix $\mathbf{L}^h$ as the discrete approximation to one of the diffusion operators in (2.3)–(2.5) with the following important modification. We construct the matrix $\mathbf{L}^h$ by evaluating the kernel on $\{x_i\}_{i=1}^{N} \cup \{\widetilde{x}_j^{G_0}\}_{j=1}^{J} \cup \{\widetilde{x}_j^{G_k}\}_{j,k=1}^{J,K}$. In the case of well-sampled data, the normal vector $\boldsymbol{\nu}$ is estimated by a secant line, and, therefore, some of these ghost points coincide with some interior points, that is, $\{x_i\}_{i=1}^{N} \supset \{\widetilde{x}_j^{G_0}\}_{j=1}^{J}$. In the case of randomly sampled data, we

define $\{x_i\}_{i=1}^{N+J} := \{x_i\}_{i=1}^{N} \cup \{\widetilde{x}_j^{G_0}\}_{j=1}^{J}$ for convenience of notation, even if these interior ghost points do not lie on M.

Since the argument below does not change whether the set $\{x_i\}$ has N or $N+J$ points, instead of using different notations for the well-sampled and randomly sampled cases, we use the same set $\{x_i\}_{i=1}^{N}$ of N-points to denote points on the manifold as well as the interior ghost points $\{\widetilde{x}_j^{G_0}\}_{j=1}^{J}$.

Here, the discrete extrapolation problem is to extend the function u identified by the function values only on x_i, x_j^B to estimate the function values $u(\widetilde{x}_j^{G_k})$ for $k = 1, \ldots, K$ by the estimated quantities $\widetilde{u}_{\epsilon,j}^{G_k}$. With these notations, we define a vector $\vec{u}_\epsilon$ by

$$\vec{u}_\epsilon = (u(x_1), \ldots, u(x_N), \widetilde{u}_{\epsilon,1}^{G_1}, \ldots, \widetilde{u}_{\epsilon,1}^{G_K}, \ldots, \widetilde{u}_{\epsilon,J}^{G_1}, \ldots, \widetilde{u}_{\epsilon,J}^{G_K}) \in \mathbb{R}^{\bar{N}}, \tag{3.13}$$

where $\bar{N} = N + JK$. Since $\{\widetilde{x}_j^{G_0}\}_{j=1}^{J} \subset \{x_i\}_{i=1}^{N}$, the first N-components include the function values $u(\widetilde{x}_j^{G_0})$. Then, we estimate $\{\widetilde{u}_{\epsilon,j}^{G_k}\}_{j,k=1}^{J,K}$, by solving the following JK algebraic equations,

$$\begin{aligned}
\big(\mathbf{L}^h \vec{u}_\epsilon\big)_{B_j} &= f\big(x_j^B\big), \\
\widetilde{u}_{\epsilon,j}^{G_2} - 2\widetilde{u}_{\epsilon,j}^{G_1} + u\big(x_j^B\big) &= \widetilde{u}_{\epsilon,j}^{G_1} - 2u\big(x_j^B\big) + u\big(\widetilde{x}_j^{G_0}\big), \\
\widetilde{u}_{\epsilon,j}^{G_3} - 2\widetilde{u}_{\epsilon,j}^{G_2} + \widetilde{u}_{\epsilon,j}^{G_1} &= \widetilde{u}_{\epsilon,j}^{G_2} - 2\widetilde{u}_{\epsilon,j}^{G_1} + u\big(x_j^B\big), \\
\widetilde{u}_{\epsilon,j}^{G_k} - 2\widetilde{u}_{\epsilon,j}^{G_{k-1}} + \widetilde{u}_{\epsilon,j}^{G_{k-2}} &= \widetilde{u}_{\epsilon,j}^{G_{k-1}} - 2\widetilde{u}_{\epsilon,j}^{G_{k-2}} + \widetilde{u}_{\epsilon,j}^{G_{k-3}}, \quad k = 4, \ldots K,
\end{aligned} \tag{3.14}$$

for $j = 1, \ldots, J$. Here, we have used the subscript B_j to denote the component corresponding to the boundary point x_j^B. The first equation in (3.14) is motivated by the classical finite-difference approach for solving the Neumann problems in (1.1), which imposes the discrete approximation of the elliptic PDE to be consistent at the boundary. The last three equations in (3.14) are the discrete analogue of matching the second-order derivatives along $\widetilde{\boldsymbol{\nu}}_j$ at the ghost points and the corresponding boundary point x_j^B.

Now we report the error in approximating the function values $u(x_j^{G_k})$ with $\widetilde{u}_{\epsilon,j}^{G_k}$, obtained from solving the algebraic conditions in (3.14).

PROPOSITION 3.6 (Extrapolation error rate for u). *Let $u \in C^3(M \cup B_{\epsilon^r}(\partial M))$, where the extended manifold $M \cup \Delta M$ is a submanifold of $\mathbb{R}^n$, constructed by Lemma* 3.5 *with $R = \mathcal{O}(\epsilon^r)$, where $0 < r < 1/2$. For each $x_j^{G_k} \in \Delta M$, let $\widetilde{u}_{\epsilon,j}^{G_k}$ be the extrapolated function value at the estimated ghost point $\widetilde{x}_j^{G_k}$, obtained by solving* (3.14). *For any fixed $j = 1, \ldots, J$,*

$$\begin{aligned}
&\big|u\big(x_j^{G_k}\big) - \widetilde{u}_{\epsilon,j}^{G_k}\big| \\
&\quad = \mathcal{O}\big(h^3, h^2\epsilon^{-1/2}, \epsilon^2, \bar{N}^{-1/2}\epsilon^{-(1+d/4)}, \bar{N}^{-1/2}\epsilon^{(1/2-d/4)}\big),
\end{aligned} \tag{3.15}$$

in high probability, as $\epsilon \to 0$ *after* $\bar{N} \to \infty$ *and* $h \to 0$.

PROOF. See Appendix A. □

Throughout the paper, we use the notation $\mathcal{O}(f, g, w)$ as a shorthand for $\mathcal{O}(f) + \mathcal{O}(g) + \mathcal{O}(w)$ as $f, g, w \to 0$. The first and second error bounds in the "big-oh" notation in (3.15) correspond to the extrapolation estimates with ghost points specified with distance h for a fixed $\epsilon > 0$, so they are defined as $h \to 0$. The fourth and fifth error bounds in the "big-oh" notation above in (3.15) correspond to the Monte Carlo estimates of the integral operator for fixed $\epsilon > 0$ so they are defined as $\bar{N} \to \infty$. Therefore, the "big-oh" notation in (3.15) (and in the remainder of this paper) is defined as $\epsilon \to 0$ after $\bar{N} \to \infty$ and $h \to 0$.

Remark 3.7 (Randomly sampled data). In this case, the leading error term in (3.15) is of order-$h^2\epsilon^{-1/2}$. This error rate is contributed by the estimated interior ghost points that do not lie on M and the exterior ghost points that do not lie on ΔM. In Appendix A, we shall see how the distances between the estimated ghost points and the points on the extended manifold,

$$\left|\gamma_j(h) - \tilde{x}_j^{G_0}\right| = \mathcal{O}(h\sqrt{\epsilon}), \quad \left|x_j^{G_k} - \tilde{x}_j^{G_k}\right| = \mathcal{O}(h\sqrt{\epsilon}),$$

where $\gamma_j(h) := \exp_{x_j^B}(-h\boldsymbol{\nu}_{x_j^B}) \in M$ and $\{x_j^{G_k}\}_{j,k=1}^{J,K} \subset \Delta M$, contribute to this error rate.

Remark 3.8 (Well-sampled data). In this case, since the secant line approximation is used to approximate $\boldsymbol{\nu}$, the estimated ghost points $\{\tilde{x}_j^{G_0}\}$ coincide with some components of $\{x_i \in M\}$. Since these interior ghost points lie on the manifold, they do not contribute to the error rate-$h^2\epsilon^{-1/2}$. While the estimated exterior ghost points, $\{\tilde{x}_j^{G_k}\}_{j,k=1}^{J,K}$, do not exactly lie on ΔM, we numerically also found that they do not contribute to the error of order-$h^2\epsilon^{-1/2}$. We suspect that this is because the diffusion maps algorithm, applied on the extended data, $\{x_i \in M\}_{i=1}^{N} \cup \{\tilde{x}_j^{G_k}\}_{j,k=1}^{J,K}$, is approximating the differential operator on a different smooth extended domain $\widetilde{M \cup \Delta M}$ that contains these points, and the error rate in Lemma A.1 is still valid for the matrix $\mathbf{L}^h$ under the assumption that $u \in C^3(M \cup B_{\epsilon^r}(\partial M))$. In light of this, for well-sampled data, the leading error is the first error term of order-h^3 in (3.15).

3.5 The ghost point diffusion maps estimator

Here, we continue using the notation $\vec{u}_\epsilon$ as defined in (3.13), where the first N-components contain the function value at the estimated interior ghost points $\tilde{x}_j^{G_0}$ that may or may not lie exactly on M, depending on the distribution of the data. For the discussion below, we also define the column

vectors:

$$\begin{aligned} \vec{u}_\epsilon^{\,M} &= (u(x_1), \ldots, u\big(\widetilde{x}_1^{G_0}\big), \ldots, u\big(\widetilde{x}_J^{G_0}\big), \ldots, u(x_N)), \\ \vec{u}_\epsilon^{\,G} &= \big(\widetilde{u}_{\epsilon,1}^{G_1}, \ldots, \widetilde{u}_{\epsilon,J}^{G_K}\big), \\ \vec{u}_\epsilon &= \big(\vec{u}_\epsilon^{\,M}, \vec{u}_\epsilon^{\,G}\big), \end{aligned} \tag{3.16}$$

where we emphasized that some of the components of $\vec{u}_\epsilon^{\,M} \in \mathbb{R}^N$ are $u(\widetilde{x}_j^{G_0})$ in the definition above. Similarly, we also define

$$\begin{aligned} \vec{u}^{\,M} &= (u(x_1), \ldots, u(\gamma_1(h)), \ldots, u(\gamma_J(h)), \ldots, u(x_N)), \\ \vec{u}^{\,G} &= \big(u\big(x_1^{G_1}\big), \ldots, u\big(x_J^{G_K}\big)\big), \\ \vec{u} &= (\vec{u}^{\,M}, \vec{u}^{\,G}), \end{aligned} \tag{3.17}$$

where $\vec{u}^{\,M} \in \mathbb{R}^N$ contains $u(\gamma_j(h))$, replacing each component $u(\widetilde{x}_j^{G_0})$ of $\vec{u}_\epsilon^{\,M}$. These definitions imply that

$$\vec{u}_\epsilon^{\,M} = \begin{cases} \vec{u}^{\,M} + \mathcal{O}(h\sqrt{\epsilon}) & \text{if } x_i \text{ are randomly sampled,} \\ \vec{u}^{\,M} & \text{if } x_i \text{ are well-sampled.} \end{cases} \tag{3.18}$$

Recall that equation (3.14) consists of a system of JK equations and it has a unique solution that can be written in compact form as

$$\vec{u}_\epsilon^{\,G} = \mathbf{A}\vec{u}_\epsilon^{\,M} + \vec{b}, \tag{3.19}$$

where one can see the detailed expression of $\mathbf{A} \in \mathbb{R}^{JK\times N}$ and $\vec{b} \in \mathbb{R}^{JK}$ for the 1D case in Appendix B. Here, components of $\vec{b}$ depend on $f(x_j^{\mathcal{B}})$. To this end, we denote the discrete approximation with a nonsquare matrix $\mathbf{L}^h = (\mathbf{L}^{(1)}, \mathbf{L}^{(2)}) \in \mathbb{R}^{N\times\bar{N}}$ that maps vectors $\vec{u}_\epsilon \in \mathbb{R}^{\bar{N}}$ into $\mathbf{L}^h\vec{u}_\epsilon \in \mathbb{R}^N$, where the matrix $\mathbf{L}^h$ is constructed as discussed in Section 3.4. For the discussion below, we define the matrix $\mathbf{L} \in \mathbb{R}^{N\times\bar{N}}$, as a discrete estimator of $\mathcal{L}$ that is constructed in analogous to $\mathbf{L}^h$ except that the kernel is evaluated on $\gamma_j(h) \in M$ (and $\{x_j^{G_k} \in \Delta M\}_{j,k=1}^{J,K}$) in placed of $\widetilde{x}_j^{G_0}$ (and $\{\widetilde{x}_j^{G_k}\}_{jk,=1}^{J,K}$), in addition to the evaluation at all sampled points of M in the construction of $\mathbf{L}^h$. We should point out that each row of the nonsquare matrices $\mathbf{L}$ and $\mathbf{L}^h$ corresponds to the kernel evaluation at the components of $\{x_i\}_{i=1}^N$, where the former includes $\{\gamma_j(h)\}$ and the latter includes $\{\widetilde{x}_j^{G_0}\}$.

Since we are interested in approximating $\vec{u}^{\,M} \in \mathbb{R}^N$ with the constraint that $\vec{u}^{\,G}$ is not available, we define the GPDM estimator $\mathbf{L}^g : \mathbb{R}^N \to \mathbb{R}^N$ as the following affine operator,

$$\mathbf{L}^g(\vec{u}^{\,M}) := \big(\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A}\big)\vec{u}^{\,M} + \mathbf{L}^{(2)}\vec{b}. \tag{3.20}$$

With this definition, we should point out that

$$\mathbf{L}^g\big(\vec{u}_\epsilon^{\,M}\big) = \mathbf{L}^{(1)}\vec{u}_\epsilon^{\,M} + \mathbf{L}^{(2)}\big(\mathbf{A}\vec{u}_\epsilon^{\,M} + \vec{b}\big) = \mathbf{L}^{(1)}\vec{u}_\epsilon^{\,M} + \mathbf{L}^{(2)}\vec{u}_\epsilon^{\,G} = \mathbf{L}^h\vec{u}_\epsilon,$$

where we have used (3.19). We should also point out that clearly $\mathbf{L}^g(\vec{u}^M) \neq \mathbf{L}^h\vec{u}$ since (3.19) is not valid for the pair of $\vec{u}^G$ and $\vec{u}^M$, that is, $\vec{u}^G \neq \mathbf{A}\vec{u}^M + \vec{b}$. With all these definitions, we now state the consistency of the GPDM estimator in (3.20).

THEOREM 3.9 (Consistency of the GPDM). *Let $u \in C^3(M \cup B_{\epsilon^r}(\partial M))$, where the extended manifold $M \cup \Delta M$ is a submanifold of $\mathbb{R}^n$, constructed by Lemma 3.5 with $R = \mathcal{O}(\epsilon^r)$, where $0 < r < 1/2$, such that $\Delta M \subset B_{\epsilon^r}(\partial M)$. For each $x_i \in M$, where $\{x_i\}_{i=1}^N \supset \{\gamma_j(h)\}_{j=1}^J$,*

$$\begin{aligned}&\left|\left(\mathbf{L}^g(\vec{u}^M)\right)_i - \mathcal{L}u(x_i)\right| \\ &\quad = \mathcal{O}\left(h^3\epsilon^{-1}, h^2\epsilon^{-3/2}, h\epsilon^{-1/2}, \epsilon, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}\right),\end{aligned}$$

in high probability as $\epsilon \to 0$ after $\bar{N} \to \infty$ and $h \to 0$.

PROOF. For each $i = 1, \ldots, N$, using the definitions in (3.16)–(3.18),

$$\begin{aligned}&\left|\left(\mathbf{L}^g(\vec{u}^M)\right)_i - \mathcal{L}u(x_i)\right| \\ &\quad = \left|\left(\mathbf{L}^g\left(\vec{u}_\epsilon^M\right)\right)_i - \mathcal{L}u(x_i) + \left(\mathbf{L}^g(\vec{u}^M) - \mathbf{L}^g\left(\vec{u}_\epsilon^M\right)\right)_i\right| \\ &\quad = \left|\left(\mathbf{L}^{(1)}\vec{u}_\epsilon^M + \mathbf{L}^{(2)}\vec{u}_\epsilon^G\right)_i - \mathcal{L}u(x_i) + \left((\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})\left(\vec{u}^M - \vec{u}_\epsilon^M\right)\right)_i\right| \\ &\quad = \left|(\mathbf{L}^{(1)}\vec{u}^M)_i + (\mathbf{L}^{(2)}\vec{u}^G)_i - \mathcal{L}u(x_i)\right. \\ &\qquad \left. + \left(\mathbf{L}^{(2)}\mathbf{A}\left(\vec{u}^M - \vec{u}_\epsilon^M\right)\right)_i + \left(\mathbf{L}^{(2)}\left(\vec{u}_\epsilon^G - \vec{u}^G\right)\right)_i\right| \\ &\quad = \left|(\mathbf{L}^h\vec{u})_i - \mathcal{L}u(x_i) + \left(\mathbf{L}^{(2)}\mathbf{A}\left(\vec{u}^M - \vec{u}_\epsilon^M\right)\right)_i + \left(\mathbf{L}^{(2)}\left(\vec{u}_\epsilon^G - \vec{u}^G\right)\right)_i\right| \\ &\quad \leq \left|(\mathbf{L}^h\vec{u})_i - (\mathbf{L}\vec{u})_i\right| + \left|(\mathbf{L}\vec{u})_i - \mathcal{L}u(x_i)\right| \\ &\qquad + \left|\left(\mathbf{L}^{(2)}\mathbf{A}(\vec{u}_\epsilon^M - \vec{u}^M)\right)_i\right| + \left|\left(\mathbf{L}^{(2)}(\vec{u}_\epsilon^G - \vec{u}^G)\right)_i\right|, \\ &\quad = \mathcal{O}(h^2\epsilon^{-3/2}) + \mathcal{O}(\epsilon, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}) + \mathcal{O}(h\epsilon^{-1/2}) \\ &\qquad + \mathcal{O}\left(h^3\epsilon^{-1}, h^2\epsilon^{-3/2}, \epsilon, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}\right).\end{aligned}$$

The first error term is a consequence of the diffusion matrices from two different sets of points as seen in Lemma A.2, which is equation (A.9). The second error term is the pointwise error bound of the standard diffusion maps in Lemma A.1. The third error term results from the estimated interior ghost points and is bounded by (3.18) multiplied by ϵ^{-1} from the components of $\mathbf{L}^{(2)}\mathbf{A}$. The last error term results from the estimated exterior ghost points and is given by Proposition 3.6 multiplied by ϵ^{-1} from the components of $\mathbf{L}^{(2)}$.

For the well-sampled data, the third error term also vanishes based on (3.18). The first error bound and the second term $h^2\epsilon^{-3/2}$ in the fourth error bound are both not applicable as discussed in Remark 3.8. Thus, the leading error term is $h^3\epsilon^{-1}$ from the fourth error bound for well-sampled data. For the random only sampled data, the leading error terms are $h^2\epsilon^{-3/2}$ and $h\epsilon^{-1/2}$. □

We should point out that for the approximation of the operators $\mathcal{L}_2$ and $\mathcal{L}_3$ in (2.4) and (2.5), respectively, we assume that κ, B, and C are well-defined functions

of the domain $M \cup B_{\epsilon^r}(\partial M)$. This is due to the fact that the associated asymptotic expansion in equation (2.4) or the kernel in equation (2.6) require evaluations of the associated kernel at the estimated ghost points $\{\tilde{x}_j^{G_k}\}_{j=1,k=0}^{J,K} \subset B_{\epsilon^r}(\partial M)$. While one can devise an extrapolation method to determine the function values at these ghost points if the functions were only defined on M, we neglect it in the present work to avoid the extra complication in the analysis above. In our numerics below, we assume that we are given κ, B, C that can be evaluated on any point cloud $x \in M \cup B_{\epsilon^r}(\partial M) \subset \mathbb{R}^n$.

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 (3.20) in estimating $\mathcal{L}_2 u$ for functions u that satisfy various boundary conditions. For the Robin boundary condition, $\beta_1 \partial_{\nu} u + \beta_2 u = g$, 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 $f = \mathcal{L}_2 u$ can be calculated from (2.13). For the Dirichlet and Neumann boundary conditions, we choose the appropriate u that satisfies the boundary conditions and proceed in a similar fashion.

The components of u are evaluated at equally angle distributed points $\{\theta_i = \frac{(i-1)\pi}{N-1}\}_{i=1,\ldots,N}$. In the following numerical experiment, we set $N = 400$ and $k = 50$ nearest neighbors (this is the same configuration that produces Figure 2.1). Figure 3.4 shows the forward error (FE) defined as $\left\| \mathbf{L}^g(\vec{u}^M) - \mathcal{L}_2 u \right\|_\infty$ as a function of the bandwidth parameter ϵ for various boundary conditions. One can see from Figure 3.4 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 Figure 3.4, we also show the results obtained from the auto-tuning algorithm discussed in Section 2. While this automated tuning strategy may not necessarily give the best estimates on the resulting operator estimation (for example, notice that the yellow and blue points in Figure 3.4 do not correspond to the minimum Forward Error), it often gives a starting point for further tuning and is numerically cheap. For a theoretically justifiable method, yet computationally more elaborate, one can also use the local singular value decomposition technique in [5].

Figure 3.5 shows the FE as a function of the number of points N. For comparison, we also show numerical results obtained from the standard DM without adding ghost points. The GPDM FE $\|\mathbf{L}^g(\vec{u}^M) - \mathcal{L}_2 u\|_\infty$ (green curve) is a uniform error computed at all N points on manifold M. The DM FE $\|\mathbf{L}\vec{u}^M - \mathcal{L}_2 u\|_\infty$ 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 the boundary. One can see from Figure 3.5 that the DM FE on the interior of M and the GPDM

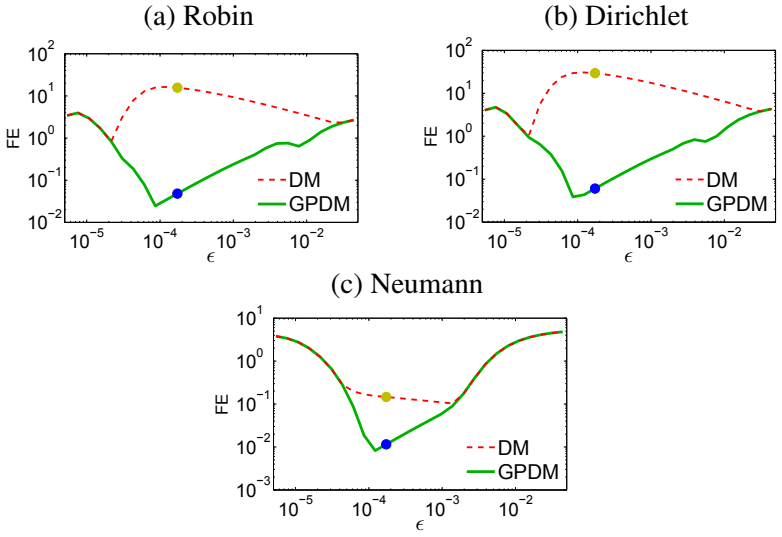


FIGURE 3.4. Forward error (FE) of the operator estimation as a function of the bandwidth ϵ for the semi-ellipse example with fixed $N = 400$ well-sampled data. The operator acts on a test function satisfying homogeneous: (a) Robin, (b) Dirichlet, and (c) Neumann boundary conditions. The yellow point and blue point correspond to the auto-tuned ϵ for DM and GPDM, respectively.

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 [25]). However, in the presence of boundary conditions, only the GPDM FE decays as $\mathcal{O}(N^{-2})$.

4 Applications: Solving Linear Elliptic PDE's

In this section, we consider solving the elliptic PDE's problem on a smooth manifold M,

$$\begin{aligned} \mathcal{L}u &= f, && x \in M^o, \\ \mathcal{B}u &:= (\beta_1 \partial_{\boldsymbol{\nu}} + \beta_2)u = g, && x \in \partial M, \end{aligned} \tag{4.1}$$

where β_1, β_2 are smooth real-valued functions such that $\beta_1\beta_2 > 0$ on ∂M. Here, the differential operator $\mathcal{L}$ is one of (2.3)–(2.5) and is assumed to be uniformly elliptic with smooth coefficients (if any). Here, the smoothness will determine the regularity of the solution. 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 $a(x) \geq a_{\min} > 0 \ \ \forall x \in M$

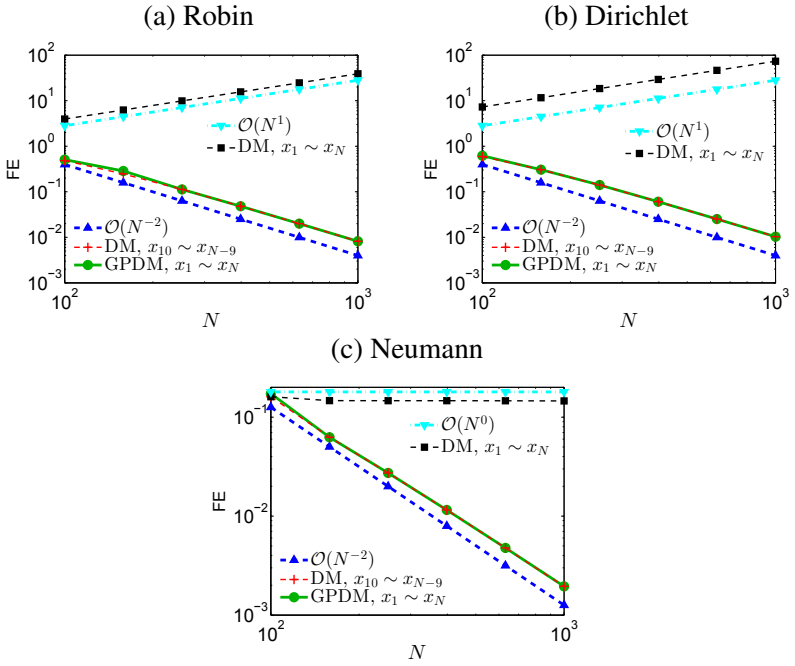


FIGURE 3.5. Comparisons of Forward Errors (FEs) of the estimated operators as functions of the number of points N for the semi-ellipse example. The operator acts on a test function satisfying homogeneous: (a) Robin, (b) Dirichlet, and (c) Neumann boundary conditions. For GPDM, the FE $\left\|\mathbf{L}^g(\vec{u}^M) - \mathcal{L}_2 u\right\|_\infty$ is computed on all points on the manifold, M (green solid line). For DM, The FE $\|\mathbf{L}\vec{u}^M - \mathcal{L}_2 u\|_\infty$ is computed on all points on M (black dashed line) and only on interior points $\{x_i\}_{i=10,\ldots,N-9}$ 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 number of well-sampled data.

for a well-posed problem. For the Robin boundary condition, we also add $-a$ for convenience of the convergence study. For $f \in C^{1,\alpha}(M)$, where $0 < \alpha < 1$, the PDE problem with appropriate smoothness of the coefficients (if any) admits a unique classical solution $u \in C^{3,\alpha}(M)$, when

$$g \in \begin{cases} C^{3,\alpha}(\partial M) & \text{for Dirichlet boundary,} \\ C^{2,\alpha}(\partial M) & \text{for both the Robin and Neumann.} \end{cases} \tag{4.2}$$

We should point out that we impose one-order derivative higher than the usual Schauder estimates ($u \in C^{2,\alpha}$) since the diffusion maps asymptotic expansion (Lemma A.1) requires a C^3-function. For the detailed statement of the Schauder estimates, see theorem 6.11 of [27] or theorem 6.25 of [26] for the Dirichlet problem, theorem 6.31 of [26] for the Robin problem, and [42] for the Neumann problem.

We should also point out that since the convergence analysis will rely on the consistency of the GPDM estimator, we require that u be C^3 not only on M but on the extended domain $M \cup B_{\epsilon^r}(\partial M)$, which is assumed in Theorem 3.9. The appropriate regularity for κ, B, C on the extended domain is also implicitly assumed for the consistent GPDM estimators to both differential operators $\mathcal{L}_2$ and $\mathcal{L}_3$. Likewise, the function values of f on the estimated interior ghost points $\tilde{x}_j^{G_0}$ are also assumed to be well defined and that

$$f \in C^1(M \cup B_{\epsilon^r}(\partial M)). \tag{4.3}$$

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 (4.1) with the affine operator in (3.20) for our GPDM method. To be concise, we define $\hat{u}^M = (\hat{u}_1, \ldots, \hat{u}_N)$, whose components are the numerical solution of the elliptic problem at $\{x_i\}_{i=1}^N$ that also include solutions at the estimated ghost points, $\{\tilde{x}_j^{G_0}\}_{j=1}^J$. Then, the PDE is discretized as

$$\mathbf{L}^g(\hat{u}^M) = (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})\hat{u}^M + \mathbf{L}^{(2)}\vec{b} = \vec{f}, \tag{4.4}$$

where $\vec{f} \in \mathbb{R}^N$, with components $f_i = f(x_i)$, $x_i \in M$, and $x_i = \tilde{x}_j^{G_0}$ for some i, j. In the analysis below, we will establish the convergence of the solution $\hat{u}^M$ of the linear problem in (4.4) to the true solution, $\vec{u}^M$, as defined in (3.17), subjected to boundary conditions.

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

$$\beta_1(x_i)\partial_\nu + \beta_2(x_i) \approx \beta_1(x_i)\left(\frac{\delta(x_i) - \delta(\tilde{x}_i^{G_0})}{h}\right) + \beta_2(x_i)\delta(x_i) := \mathbf{B}_i,$$

following equation (3.8). 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 $\tilde{x}_i^{G_0}$ and $x_i \in \partial M$. With this notation, the estimated boundary condition can be written in a compact form as

$$\mathbf{B}\hat{u}^M = \vec{g}, \tag{4.5}$$

where $\vec{g} \in \mathbb{R}^J$, with components $g_i = g(x_i)$ for all $x_i \in \partial M$. Then we have

$$
\begin{aligned}
&(\mathbf{B}\widehat{u}^M - \mathbf{B}\vec{u}^M)_j \\
&\quad = g\big(x_j^B\big) - (\mathbf{B}\vec{u}^M)_j = \mathcal{B}u\big(x_j^B\big) - (\mathbf{B}\vec{u}^M)_j \\
&\quad = \mathcal{B}u\big(x_j^B\big) - \big(\mathbf{B}\big(\vec{u}_\epsilon^M + \mathcal{O}(h\epsilon^{1/2})\big)\big)_j \\
&\quad = \beta_1\big(x_j^B\big)\bigg(\partial_{\boldsymbol{\nu}} u\big(x_j^B\big) - \frac{u\big(x_j^B\big) - u\big(\widetilde{x}_j^{G_0}\big)}{h} + \mathcal{O}(\epsilon^{1/2})\bigg) \\
&\quad = \mathcal{O}(h, \epsilon^{1/2}),
\end{aligned}
\tag{4.6}
$$

for $j = 1, \ldots, J$. For the equality in the third line, we have used (3.18) such that the error of order $h\epsilon^{1/2}$ only occurs for the randomly sampled data. As for the equality in the fifth line, for well-sampled data, $\widetilde{x}_j^{G_0}$ coincides with one of the interior points (due to the secant line approximation), and the error bound in the approximation of the directional derivative is of order-h. For the randomly sampled data, the error bound in the approximation of the directional derivative is given by (3.8).

Dirichlet Problem: Numerically, we consider solving an $(N-J)\times(N-J)$ linear problem that is obtained by asserting (4.5) to the first $N-J$ row of (4.4). 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,

$$
\big(\mathbf{L}^I \mid \mathbf{L}^B\big) = \begin{pmatrix} (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})_1 \\ \vdots \\ (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})_{N-J} \end{pmatrix} \in \mathbb{R}^{(N-J)\times N}.
\tag{4.7}
$$

and decompose the estimated solution $\widehat{u}^M = (\widehat{u}^I, \widehat{u}^B)$ to

$$
\begin{aligned}
\widehat{u}^I &= (\widehat{u}_1, \ldots, \widehat{u}_{N-J}) && \text{for interior components,} \\
\widehat{u}^B &= (\widehat{u}_{N-J+1}, \ldots, \widehat{u}_N) && \text{for the boundary components.}
\end{aligned}
$$

Similarly, we will decompose the true solution as $\vec{u}^M = (\vec{u}^I, \vec{u}^B)$ with

$$
\begin{aligned}
\vec{u}^I &= (u(x_1), \ldots, u(\gamma_j(h)), \ldots, u(x_{N-J})) && \text{for the interior components,} \\
\vec{u}^B &= (u_{N-J+1}, \ldots, u_N) = (u(x_1^B), \ldots, u(x_J^B)) && \text{for the boundary components.}
\end{aligned}
$$

For the Dirichlet boundary condition, $u(x_j^B) = g(x_j^B)$ for $j = 1, \ldots, J$, then one can directly replace $\widehat{u}_{N-J+j} = u(x_j^B) = g(x_j^B)$, applying the decomposition in (4.7) on the first $N-J$ rows of (4.4), we arrive at the following reduced system,

$$
\mathbf{L}^I \widehat{u}^I = \vec{f}^I - \mathbf{L}^B \vec{g}
\tag{4.8}
$$

where we have also defined

$$
\vec{f}^I = (f(x_1) - (\mathbf{L}^{(2)}\vec{b})_1, f(x_2) - (\mathbf{L}^{(2)}\vec{b})_2, \ldots, f(x_{N-J}) - (\mathbf{L}^{(2)}\vec{b})_{N-J}).
$$

We now show that the solution of (4.8) converges to the solution of the PDE in (4.1) with Dirichlet boundary condition.

THEOREM 4.1 (Convergence of the Dirichlet Problem). *Let u be the solution of the PDE in* (4.1) *with Dirichlet boundary condition, $u(x_j^B) = g(x_j^B)$ for $j = 1, \ldots, J$. Assuming the regularity in* (4.2) *and* (4.3) *for g and f, let $\widehat{u}_i$ be the solution of the linear system in* (4.8)*, where the diffusion operator $\mathcal{L}$ is approximated by the GPDM affine estimator in* (3.20)*, constructed with N grid points on the manifold and the estimated ghost points* (3.12)*, whose consecutive distance is $h > 0$ such that the consistency in Theorem* 3.9 *is valid. Assume that the differential operator $\mathcal{L}$ satisfies the maximum principle; then for any $x_i \in M^o$, $\widehat{u}_i$ converges to $u(x_i)$ with an error bound given as*

$$|\widehat{u}_i - u(x_i)| = \mathcal{O}\big(h^3\epsilon^{-1}, h^2\epsilon^{-3/2}, h\epsilon^{-1/2}, \epsilon, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}\big),$$

in high probability, as $\epsilon \to 0$ after $\bar{N} \to \infty$ and $h \to 0$.

PROOF. See Appendix B. □

Recall that some components of $\{\widehat{u}_i\}_{i=1}^{N-J}$ correspond to the numerical solutions at the ghost points $\{\tilde{x}_j^{G_0}\}_{j=1}^{J}$. For these components, $\widehat{u}_i$ converges to the true solution u, evaluated at the corresponding point $\gamma_j(h) \in M$. We will elaborate this case in the proof of the next Theorem 4.2 (see the discussion after (4.12)).

Robin and Neumann problems: Here, we consider

$$\begin{aligned} (-a + \mathcal{L})u &= f, && x \in M^o, \\ \mathcal{B}u := (\beta_1 \partial_{\boldsymbol{\nu}} + \beta_2)u &= g, && x \in \partial M, \end{aligned} \tag{4.9}$$

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 (4.9) is given by the following $N \times N$ system,

$$\mathbf{N}\widehat{u}^M := \begin{pmatrix} -\mathbf{a} + \mathbf{L}^I & \mathbf{L}^B \\ \mathbf{B}^I & \mathbf{B}^B \end{pmatrix} \begin{pmatrix} \widehat{u}^I \\ \widehat{u}^B \end{pmatrix} = \begin{pmatrix} \vec{f}^I \\ \vec{g} \end{pmatrix}, \tag{4.10}$$

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,

$$\widehat{u}^B = (\mathbf{B}^B)^{-1}(\vec{g} - \mathbf{B}^I\widehat{u}^I), \tag{4.11}$$

and insert this solution to the first $(N - J)$ rows in problem (4.10) 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* (4.9) *with Robin boundary condition and $\beta_1, \beta_2 > 0$. Let the corresponding GPDM estimator be constructed as in Theorem* 4.1 *and assume that $a \in C^1(M \cup B_\epsilon(\partial M))$. Assuming the regularity in* (4.2) *and* (4.3) *for g and f, for any $x_i \in M^o$, the solution $\widehat{u}_i$ of the linear system in* (4.10) *converges to $u(x_i)$ with error bound given as*

$$|\widehat{u}_i - u(x_i)| = \mathcal{O}\big(h^3\epsilon^{-1}, h^2\epsilon^{-3/2}, h\epsilon^{-1/2}, \epsilon^{1/2}, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}\big),$$

in high probability, as $\epsilon \to 0$ after $\bar{N} \to \infty$ and $h \to 0$.

PROOF. Using the definition of $\vec{f}^I$ and the decomposition in (4.7), one can immediately see the consistency. Multiplying the matrix $\mathbf{N}$ in (4.10) with a vector consists of the difference between the estimated and the true solutions, we obtain

$$\begin{aligned}
&\big((-\mathbf{a} + \mathbf{L}^I)(\widehat{u}^I - \vec{u}^I) + \mathbf{L}^B(\widehat{u}^B - \vec{u}^B)\big)_i \\
&\quad = \big(\vec{f}^I - (-\mathbf{a} + \mathbf{L}^I)\vec{u}_I - \mathbf{L}^B\vec{u}_B\big)_i \\
&\quad = f(x_i) + a(x_i)u(x_i) - \big(\mathbf{L}^{(2)}\vec{b} + \mathbf{L}^I\vec{u}_I + \mathbf{L}^B\vec{u}_B\big)_i \\
&\quad = \mathcal{L}u(x_i) - (\mathbf{L}^g(\vec{u}^M))_i
\end{aligned} \tag{4.12}$$

for $i = 1, \ldots, N - J$. For the randomly sampled case, some of the elements of $\{x_i\}$ are $\tilde{x}_j^{G_0}$ that do not lie on M. For such components, we have

$$\begin{aligned}
&\big((-\mathbf{a} + \mathbf{L}^I)(\widehat{u}^I - \vec{u}^I) + \mathbf{L}^B(\widehat{u}^B - \vec{u}^B)\big)_i \\
&\quad = f\big(\tilde{x}_j^{G_0}\big) + a\big(\tilde{x}_j^{G_0}\big)u(\tilde{x}_j^{G_0}) - \big(\mathbf{L}^g(\vec{u}^M)\big)_i \\
&\quad = \mathcal{L}u(\gamma_j(h)) - \big(\mathbf{L}^g(\vec{u}^M)\big)_i \\
&\qquad + \big(f\big(\tilde{x}_j^{G_0}\big) - f(\gamma_j(h)) + a\big(\tilde{x}_j^{G_0}\big)u\big(\tilde{x}_j^{G_0}\big) - a(\gamma_j(h))u(\gamma_j(h))\big) \\
&\quad = \mathcal{L}u(\gamma_j(h)) - \big(\mathbf{L}^g(\vec{u}^M)\big)_i + \mathcal{O}(h\epsilon^{1/2}),
\end{aligned}$$

where the last term is valid under the assumption that $a, f, u \in C^1(B_{\epsilon^r}(\partial M))$. The last J rows corresponding to the boundary points are nothing but (4.6).

From equation (B.6) in Appendix B, 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

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

for $i = 1, \ldots, N - J$. 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 (4.10) is given as

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

In this case, $|\mathbf{N}_{i,i}| - \sum_{j=1, j\neq i}^{N} |\mathbf{N}_{i,j}| = \beta_2(x_i) > 0$ for $i > N - J$. Therefore, the matrix $\mathbf{N}$ is strictly diagonal dominant and nonsingular. By the Ahlberg-Nilson-Varah bound [1, 50], the inverse matrix is uniformly bounded,

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

Thus, multiplying $\mathbf{N}^{-1}$ to a vector where the first $N - J$ components consist of (4.12) and the last J components consist of (4.6), we have

$$\begin{aligned}
|\widehat{u}_i - u(x_i)| &\leq \|\mathbf{N}^{-1}\|_\infty \Big(\max_{i=1,\ldots,N-J} \big| \big((-\mathbf{a} + \mathbf{L}^I)(\widehat{u}^I - \vec{u}^I) + \mathbf{L}^B(\widehat{u}^B - \vec{u}^B) \big)_i \big|, \\
&\qquad \max_{j=1,\ldots,J} \big| \big(\mathbf{B}(\widehat{u}^M - \vec{u}^M)_j \big| \big) \Big)_j \\
&= \|\mathbf{N}^{-1}\|_\infty \Big(\max_{i=1,\ldots,N-J} \big| \mathcal{L}u(x_i) - \big(\mathbf{L}^g(\vec{u}^M) \big)_i \big|, \\
&\qquad \max_{j=1,\ldots,J} \big| \big(\mathbf{B}\big(\widehat{u}^M - \vec{u}^M\big)_j \big| \Big),
\end{aligned}$$

for all $x_i \in M$. Since the GPDM is consistent, $\big|\mathcal{L}u(x_i) - ((\mathbf{L}^g(\vec{u}^M))_i\big| \to 0$ as $\epsilon \to 0$ after $N \to \infty$ and $h \to 0$ with error rate given in Theorem 3.9. Together with the error bound in (4.6), the proof is completed. □

For the Neumann problem, the last J components of $\mathbf{N}$ are 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 (4.11) simplifies to $\widehat{u}_B = -(\mathbf{B}^B)^{-1}\mathbf{B}^I\widehat{u}^I$. For example, in a well-sampled case, the discrete approximation in (3.4) yields $\widehat{u}_N = \widehat{u}_{N-1}$ and $\widehat{u}_1 = \widehat{u}_2$. Substituting these solutions (J equations in general) to the first $N - J$ rows of (4.10), one can verify that the reduced $N - J$ problem is nonsingular and 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.

4.2 Numerical examples on simple manifolds

In this section, we discuss three examples of 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 semitorus 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 the randomly sampled data for the semitorus PDE problem.

Numerically, we will compare GPDM with the standard DM. To account for other than homogeneous Neumann boundary conditions, we modify the standard DM as follows. We consider the $N - J$ rows corresponding to the interior points $x_i \in M^o$ of the equation, $\mathbf{L}_{\mathrm{DM}}\widehat{u}^M = \vec{f}$, where $\mathbf{L}_{\mathrm{DM}}$ is the standard diffusion maps operator. To approximate boundary conditions that involve normal derivatives, we use the algorithm in Appendix C that requires no interior ghost points. Then the inverse problem consists of solving the reduced linear system (arising from imposing the appropriate boundary conditions), analogous to the reduced linear problem with GPDM.

Anisotropic diffusion on a semi-ellipse with well-sampled data

First, let us present the results of the 1D problem in Example 2.1 in solving

$$\mathcal{L}_2 u = f, \tag{4.13}$$

with the three boundary conditions. In this numerical experiment, the configuration is the same as in Section 3.6. In particular, Figure 4.1 demonstrates the error of the solutions $\|\widehat{u}^M - \vec{u}^M\|_\infty$, which we refer to as the inverse error (IE) as a function of ϵ for fixed $N = 400, k = 50$. 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 Figure 4.2. Particularly, for the Robin BC, one can see that the GPDM IE decays on $\mathcal{O}(N^{-1})$, whereas the DM IE does not decay 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.

Nonsymmetric backward Kolmogorov elliptic PDE on a semitorus with well-sampled data

In the next example, we consider solving $\mathcal{L}_3 u = f$, with a mixed Dirichlet-Neumann boundary condition on a semitorus $M \subset \mathbb{R}^3$. The parametrization of the torus is given in (3.2) and the corresponding Riemannian metric is defined in (3.3) with (θ, ϕ) being the two intrinsic coordinates. The differential operator $\mathcal{L}_3$ is defined as in (2.5) with

$$\begin{pmatrix} b^1(x) \\ b^2(x) \end{pmatrix} := \begin{pmatrix} 2 + x_3 \\ (x_1^2 + x_2^2)^{1/2} \end{pmatrix} = \begin{pmatrix} 2 + \sin\theta \\ 2 + \cos\theta \end{pmatrix}$$

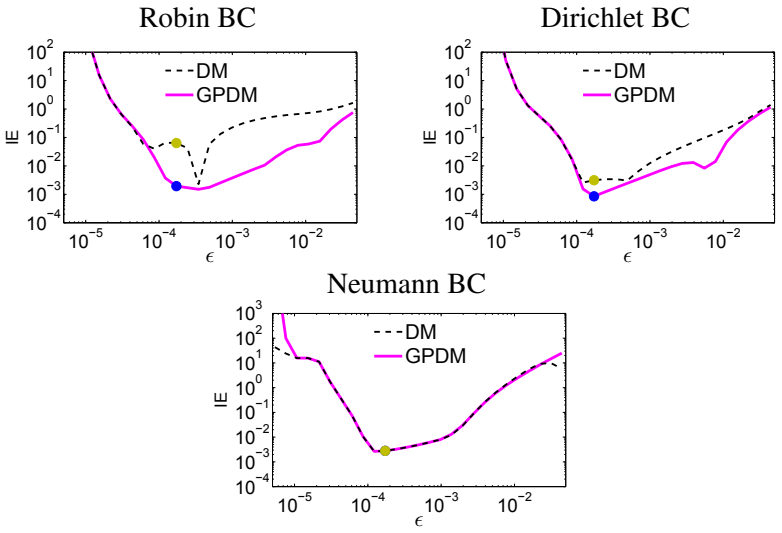


FIGURE 4.1. Pointwise inverse error (IE) of the solution of (4.13) as a function of the bandwidth ϵ for the semi-ellipse example with fixed $N = 400$ well-sampled data. The yellow point and blue point correspond to the auto-tuned ϵ for DM and GPDM, respectively.

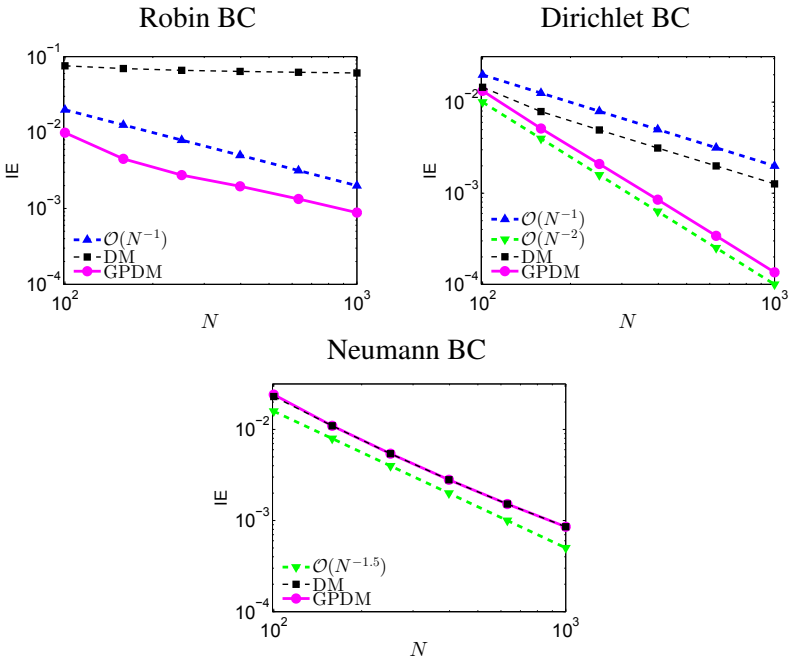


FIGURE 4.2. Comparisons of the inverse errors (IEs) of the solutions of (4.13) as functions of N for the semi-ellipse example. The bandwidth ϵ is auto-tuned for each N number of well-sampled data.

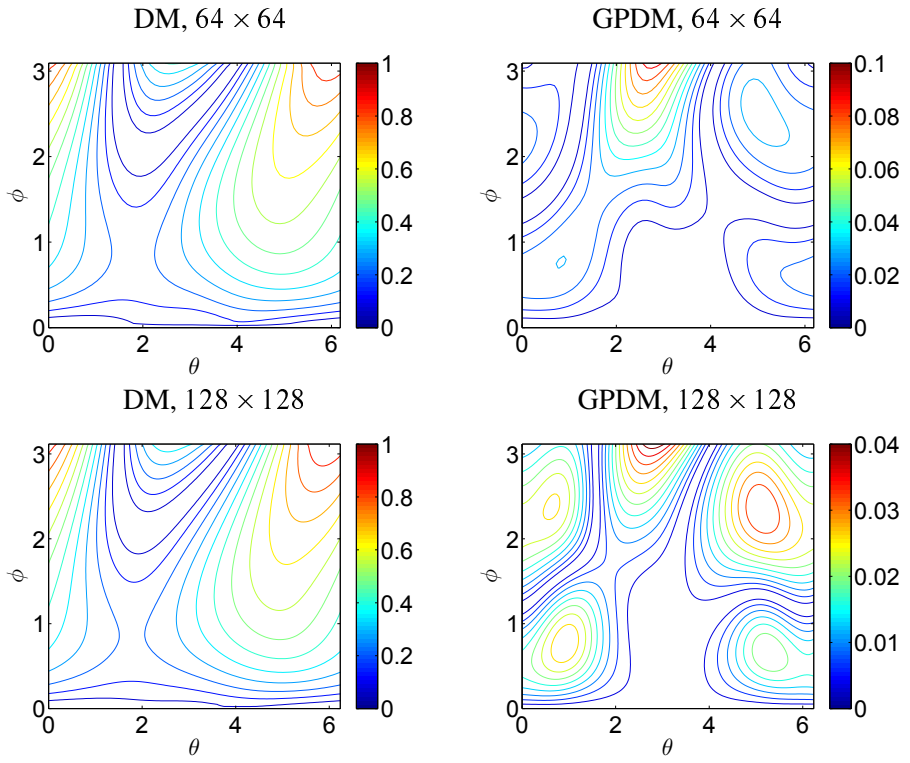


FIGURE 4.3. Absolute errors in the estimated solutions for the semitorus example with well-sampled data: (a) DM for $N = 64 \times 64$ with $\|\hat{u}^M - \vec{u}^M\|_\infty = 0.88$, (b) GPDM for $N = 64 \times 64$ with $\|\hat{u}^M - \vec{u}^M\|_\infty = 0.095$, (c) DM for $N = 128 \times 128$ with $\|\hat{u}^M - \vec{u}^M\|_\infty = 0.91$, (d) GPDM for $N = 128 \times 128$ with $\|\hat{u}^M - \vec{u}^M\|_\infty = 0.042$.

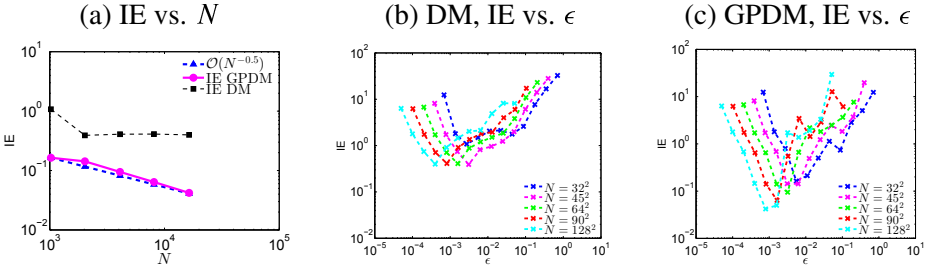


FIGURE 4.4. The semitorus example with the well-sampled data. (a) IEs of DM and GPDM methods as functions of N. For each N, the IE is obtained from the minimal inverse error for different ϵ. IEs of (b) DM and (c) GPDM methods as functions of bandwidth ϵ for different N.

$$\begin{pmatrix} c^{11}(x) & c^{12}(x) \\ c^{21}(x) & c^{22}(x) \end{pmatrix} := \begin{pmatrix} 3 + x_1/(x_1^2 + x_2^2)^{1/2} & 1/10 \\ 1/10 & 2 \end{pmatrix} = \begin{pmatrix} 3 + \cos\phi & 1/10 \\ 1/10 & 2 \end{pmatrix}$$

The semitorus is defined with the standard parametrization function as in (3.2) so that the induced Riemannian metric is given in (3.3). We set an analytic solution of this problem to be

$$u(x) = \left(\sin 2\phi - \frac{2\cos 2\phi}{2 + \cos\theta}\right)\cos\theta, \tag{4.14}$$

where

$$\begin{aligned} \cos\theta &= (x_1^2 + x_2^2)^{1/2} - 2, & \sin\theta &= x_3, \\ \cos\phi &= \frac{x_1}{(x_1^2 + x_2^2)^{1/2}}, & \sin\phi &= \frac{x_2}{(x_1^2 + x_2^2)^{1/2}}, \end{aligned} \tag{4.15}$$

with $\sin 2\phi = 2\sin\phi\cos\phi$ and $\cos 2\phi = 2\cos^2\phi - 1$. Next, we calculate $f := \mathcal{L}_3 u$ and $g := \beta_1 \partial_{\boldsymbol{\nu}} u + \beta_2 u$ at $\phi = 0$ and $\phi = \pi$. In this semitorus example, the explicit expression for f is given by

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

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

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

with the trigonometric functions defined in (4.15). 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 (4.14). 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

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

where the analytic u in (4.14) and $\phi = \pi$ have been used. Then, we approximate the solution in (4.14) for the PDE problem in (4.1), subjected to the manufactured f and g. Numerically, the grid points $\{\theta_i, \phi_j\}$ are uniformly distributed on $[0, 2\pi] \times [0, \pi]$, with $i, j = 1, \ldots, 64$ or $i, j = 1, \ldots, 128$ points in each direction, resulting in a total of $N = 4096$ or $N = 16384$ grid points. To apply the local kernel in (2.6), we use $k = 200$ nearest neighbors for all N and manually tune the kernel bandwidth as $\epsilon = 0.0032$ for $N = 4096$ and $\epsilon = 8 \times 10^{-4}$ for $N = 16384$. We found that the auto-tuned method discussed in Section 2 is not so robust for the estimation of $\mathcal{L}_3$, and we suspect that this is because the covariance in the Gaussian kernel is not constant such that the scaling used in (2.11) may not be appropriate.

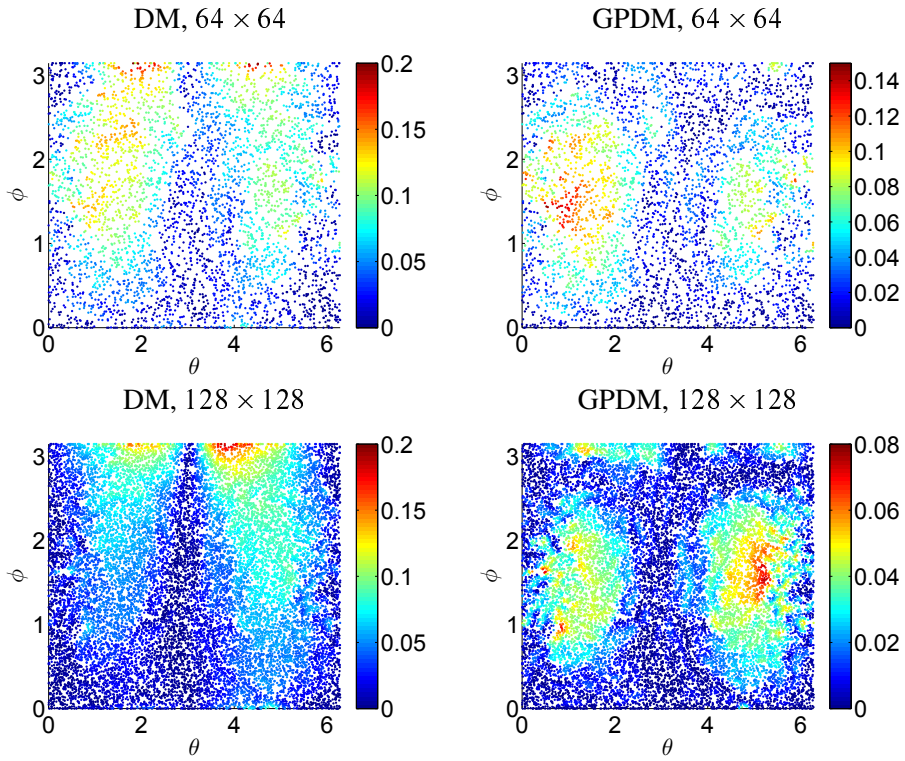


FIGURE 4.5. Absolute errors in the estimated solutions for the semi-torus example with random data: (a) DM for $N = 64 \times 64$ with $\|\vec{u}^M - \hat{u}^M\|_\infty = 0.203$, (b) GPDM for $N = 64 \times 64$ with $\|\vec{u}^M - \hat{u}^M\|_\infty = 0.146$, (c) DM for $N = 128 \times 128$ with $\|\vec{u}^M - \hat{u}^M\|_\infty = 0.186$, (d) GPDM for $N = 128 \times 128$ with $\|\vec{u}^M - \hat{u}^M\|_\infty = 0.074$.

In Figure 4.3, we show the absolute errors 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 $\|\vec{u}^M - \hat{u}^M\|_\infty = 0.9$ 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 0.095 to 0.042 as N is increased from 64×64 to 128×128.

Figure 4.4(a) shows the IEs as functions of N for DM and GPDM methods. One can see that GPDM solutions converge whereas DM solutions do not converge. Figure 4.4(b) and (c) show IEs of DM and GPDM methods, respectively, as functions of bandwidth ϵ for different N. One can see that as N increases, IE of GPDM decreases (at the rate of $\mathcal{O}(N^{-1/2})$) whereas IE of DM does not decrease.

Anisotropic diffusion on a semitorus with random data

In this example, we consider solving $\mathcal{L}_2 u = f$, with a mixed Dirichlet-Neumann boundary conditions on a semitorus $M \subset \mathbb{R}^3$. The differential operator $\mathcal{L}_2$ is

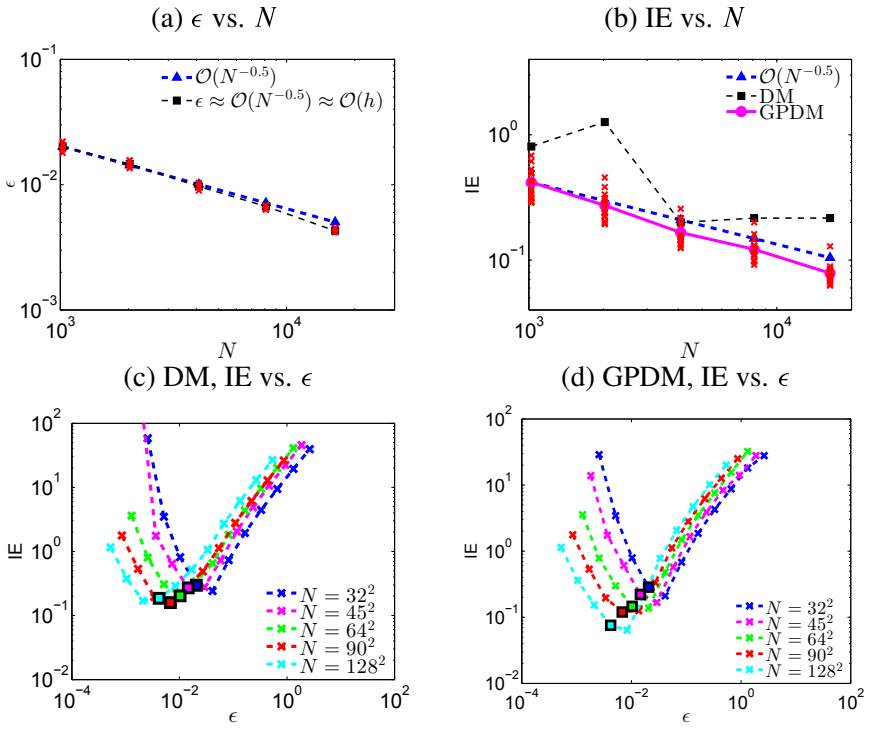


FIGURE 4.6. The semitorus example with random data in Section 4.2. A total of 16 independent trials are run. (a) The auto-tuned bandwidth ϵ as a function of the number of points N. Each red cross corresponds to an auto-tuned ϵ of one trial, and each black square corresponds to the mean of these auto-tuned ϵ. (b) IEs of DM and GPDM methods as functions of N. Each red cross is the IE for one trial. For one independent trial, plotted are (c) IEs of DM and (d) IEs of GPDM as functions of bandwidth ϵ for different N. Squares correspond to the auto-tuned ϵ.

defined as in (2.5) with

$$\kappa(x) = 1.1 + \sin^2\theta \cos^2\phi,$$

where the trigonometric functions for (θ, ϕ) as functions x are still given in (4.15). The semitorus is still defined with the embedding function as in (3.2). We set the analytic solution of this problem to be

$$u(x) = \sin\phi \sin\theta,$$

and calculate

$$f := \mathcal{L}_2 u = \frac{1}{\sqrt{|g|}} \partial_i \left(\kappa \sqrt{|g|} g^{ij} \partial_j u \right).$$

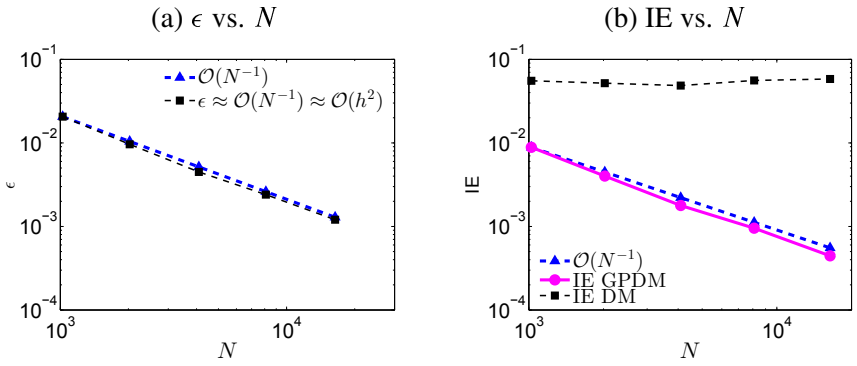


FIGURE 4.7. The semitorus example with the well-sampled data in Section 4.2. (a) The auto-tuned bandwidth ϵ as a function of number of points N. (b) IEs of DM and GPDM methods as functions of N.

The boundary conditions $g := \beta_1 \partial_{\boldsymbol{\nu}} u + \beta_2 u$ at $\phi = 0$ and $\phi = \pi$ are given the same as those in Section 4.2. Then, we approximate the solution for the PDE problem, subjected to the manufactured f and g.

Randomly sampled data: Numerically, the grid points $\{\theta_i, \phi_j\}$ are randomly uniformly distributed on $[0, 2\pi] \times [0, \pi]$. For $N = 32^2, 45^2, 64^2, 90^2, 128^2$ grid points, we set $k \sim \sqrt{N}$ and apply the ϵ-auto-tuning method discussed in Section 2.1. For each N, we show results for 16 independent trials. In Figure 4.5, we show the absolute errors in θ and ϕ between the true and the estimated solutions obtained for DM and GPDM methods for $N = 64^2$ and $N = 128^2$. For DM, the IEs are relatively large and do not follow a clear decreasing pattern as N increases ($\|\vec{u}^{\boldsymbol{M}} - \widehat{u}^{\boldsymbol{M}}\|_\infty = 0.203$ for $N = 64^2$ and $\|\vec{u}^{\boldsymbol{M}} - \widehat{u}^{\boldsymbol{M}}\|_\infty = 0.186$ for $N = 128^2$). On the other hand, the inverse error (IE) of GPDM is smaller than that of DM and decreases from 0.146 to 0.074 as N is increased from 64^2 to 128^2.

Figure 4.6(a) shows the auto-tuned bandwidth ϵ as a function of N. Figure 4.6(b) shows IEs as functions of N for both DM and GPDM. One can see that GPDM solutions converge, whereas DM solutions do not converge. Figure 4.6(c) and (d) show IEs of DM and GPDM methods, respectively, as functions of bandwidth ϵ for different N for one independent trial. One can see that as N increases, IE of GPDM decreases, whereas IE of DM does not decrease. For completeness, we depict the results of auto-tuned epsilon in squared symbols as shown in Figure 4.6(c)(d). Note that for the GPDM, the auto-tuned ϵ seems to correspond to the lowest IE.

For comparison, we also show numerical results with well-sampled data.

Well-sampled data: The grid points $\{\theta_i, \phi_j\}$ are well uniformly distributed on $[0, 2\pi] \times [0, \pi]$, with i, j, both equal to $32, 45, 64, 90, 128$ points in each direction, which is the same as those in Section 4.2. For different N grid points, we fix $k = 121$ nearest neighbors and then apply the ϵ-auto tuning method discussed in Section 2.1. One can see from Figure 4.7(a) that the auto-tuned bandwidth ϵ is on

order of N^{-1}. This rate for the well-sampled data is faster than that for random data, as shown in Figure 4.6(a). Figure 4.7(b) shows that IE of GPDM decays on the order of N^{-1}, whereas IE of DM does not decay for different N.

Other choices of k nearest neighbors and auto-tuned ϵ for GPDM: For well-sampled data, we also examined the auto-tuned ϵ under variable k nearest neighbors, that is, we choose $k \sim \sqrt{N}$. We found that for well-sampled data, the rates preserve as in Figure 4.7, that is, the bandwidth $\epsilon = \mathcal{O}(N^{-1})$ and IE is $\mathcal{O}(N^{-1})$ as well (not shown here).

For random data, we also examined the auto-tuned ϵ under fixed $k = 200$ nearest neighbors. However, we found that the results are different between using fixed k and variable k. For variable k, the bandwidth $\epsilon = \mathcal{O}(N^{-1/2})$ and IE is $\mathcal{O}(N^{-1/2})$ as shown in Figure 4.6. For fixed k, the bandwidth $\epsilon = \mathcal{O}(N^{-1})$ and IE is $\mathcal{O}(N^{-1/4})$ (not shown here).

4.3 Anisotropic diffusion on an unknown "face" manifold

In this section, we consider solving the boundary value problem in (4.13) 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 [17]. Notice that we have no access to the analytic solution since we do not know the embedding of the face surface. For comparisons, we numerically solve the problem with the finite element method (FEM) using the FELICITY FEM Matlab toolbox [54].

Figure 4.8 shows the comparison of the solutions among FEM, DM, and GPDM methods corresponding to the Robin boundary condition ($\partial_{\boldsymbol{\nu}} u + 10u = 0$ 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 Figure 4.8, 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}); here the scaling of the true solution is on the order of 10^{-3}. 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 better performance than the standard DM.

5 Applications: Solving Elliptic Eigenvalue Problems

In this section, we apply the GPDM algorithm for solving the eigenvalue problem $\mathcal{L}\psi_k = \lambda_k \psi_k$ on a manifold with boundary, where $\mathcal{L}$ is either the Laplace-Beltrami

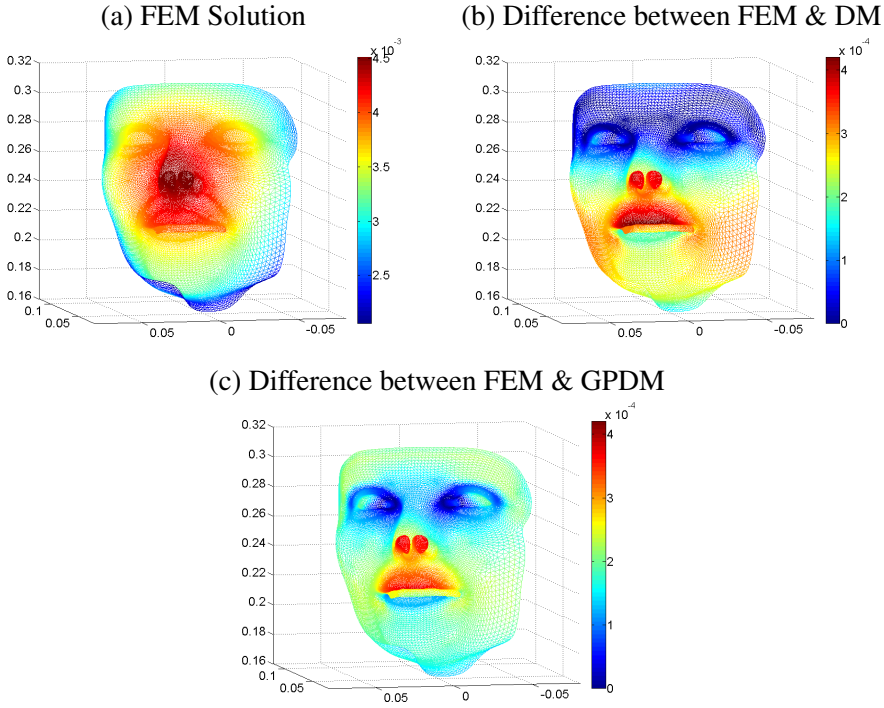


FIGURE 4.8. Comparison of the PDE solutions among FEM, DM, and GPDM on the "face" example with the Robin boundary condition. (a) FEM solution. (b) Absolute difference between FEM and DM solutions. (c) Absolute difference between FEM and GPDM solutions.

in (2.3) or the weighted Laplacian operator in (2.4). Since there is no f in this problem, we cannot use the quadratic extrapolation formula in (3.14). Instead, we extrapolate u using the linear extrapolation formula defined as follows:

$$\begin{aligned} \tilde{u}_{\epsilon,j}^{G_1} - 2u\left(x_j^B\right) + u\left(\tilde{x}_j^{G_0}\right) &= 0, \\ \tilde{u}_{\epsilon,j}^{G_2} - 2\tilde{u}_{\epsilon,j}^{G_1} + u\left(x_j^B\right) &= 0, \\ \tilde{u}_{\epsilon,j}^{G_k} - 2\tilde{u}_{\epsilon,j}^{G_{k-1}} + \tilde{u}_{\epsilon,j}^{G_{k-2}} &= 0, \quad k = 3, \ldots, K, \end{aligned} \tag{5.1}$$

where $\tilde{u}_{\epsilon,j}^{G_k}$ are the function values to be specified. It is worth noting that if we replace the quadratic extrapolation in (3.14) with (5.1), one can deduce the error rate analogous to Proposition 3.6 except that the first error bound h^3 is replaced with h^2. With this linear extrapolation formula, 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* $\bar{N} \times \bar{N}$ *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* (5.1) *for* u *at the ghost points and the homogeneous boundary condition* (4.5)*. Here,* J *is the number of boundary points. The homogeneous equations* (5.1) *and* (4.5) *have a unique solution that can be written in a compact form as a the column vector,*

$$\big(\widehat{u}^B, \vec{u}_\epsilon^G\big) = \mathbf{C}\widehat{u}^I,$$

where $\vec{u}_\epsilon^G$, $\widehat{u}^B$, *and* $\widehat{u}^I$ *are column vectors with components consisting of the estimated function values of* u *at the estimated ghost points, boundary points, and interior points, respectively, as defined in* (3.16)–(3.17)*. Here,* $\mathbf{C}$ *is a* $(JK+J)\times(N-J)$ *matrix. Denoting the column vector* $\widehat{u} := (\widehat{u}^I, \widehat{u}^B, \vec{u}_\epsilon^G) \in \mathbb{R}^{\bar{N}}$*, the diffusion operator* $\mathcal{L}$ *is approximated with the following matrix*:

$$\begin{aligned}\mathbf{L}^h\widehat{u} := \mathbf{L}^{(1)}\widehat{u}^I + \mathbf{L}^{(2)}\big(\widehat{u}^B, \vec{u}_\epsilon^G\big) &= \mathbf{L}^{(1)}\widehat{u}^I + \mathbf{L}^{(2)}\mathbf{C}\widehat{u}^I \\ &= \big(\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{C}\big)\widehat{u}^I.\end{aligned}$$

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* $(N-J)\times\bar{N}$ *matrix* $\mathbf{L}^h \equiv (\mathbf{L}^{(1)}, \mathbf{L}^{(2)})$*, and we should point out that these submatrices are different than those defined in* (3.20)*.*

(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 the eigenvalue problem $\mathcal{L}\psi_k = \lambda_k\psi_k$ with the following modification to incorporate boundary conditions other than homogeneous Neumann.

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* (4.5)*. Here,* J *is the number of boundary points. The homogeneous boundary condition* (4.5) *has a unique solution that can be written in a compact form as*

$$\widehat{u}^B = \mathbf{C}_{\mathrm{DM}}\widehat{u}^I,$$

where $\widehat{u}^B$ *and* $\widehat{u}^I$ *are vectors with components consisting of the estimated function values of* u *evaluated at the boundary points and interior points, respectively. For boundary conditions that involve normal derivatives, we used the algorithm in Appendix* C *to approximate the normal derivatives without adding ghost points. Here,* $\mathbf{C}_{\mathrm{DM}}$ *is a* $(J)\times(N-J)$ *matrix. For the formula below, we define a column vector* $\widehat{u} = (\widehat{u}^I, \widehat{u}^B)$*. Then, the diffusion operator* $\mathcal{L}$ *can be approximated with the following matrix*:

$$\begin{aligned}\mathbf{L}_{\mathrm{DM}}\widehat{u} = \mathbf{L}^{(1)}_{\mathrm{DM}}\widehat{u}^I + \mathbf{L}^{(2)}_{\mathrm{DM}}\widehat{u}^B &= \mathbf{L}^{(1)}_{\mathrm{DM}}\widehat{u}^I + \mathbf{L}^{(2)}_{\mathrm{DM}}\mathbf{C}_{\mathrm{DM}}\widehat{u}^I \\ &\equiv \big(\mathbf{L}^{(1)}_{\mathrm{DM}} + \mathbf{L}^{(2)}_{\mathrm{DM}}\mathbf{C}_{\mathrm{DM}}\big)\widehat{u}^I.\end{aligned}$$

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

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

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 manifolds with boundary for 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 semicircle in $\mathbb{R}^2$ with Dirichlet and Robin boundary conditions and a 2D semitorus in $\mathbb{R}^2$ with mixed boundary conditions.

5.1 A singular Sturm-Liouville problem

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

$$\mathcal{L}\psi_k := \frac{d}{dx}\left[\left(1-x^2\right)\frac{d\psi_k}{dx}\right] = -k(k+1)\psi_k, \tag{5.2}$$

where the eigenvalues are $\lambda_k = -k(k+1)$ with $k = 0, 1, 2, \ldots$, and the eigenfunctions ψ_k are Legendre polynomials. The Legendre polynomials are orthogonal with respect to a uniformly distributed 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 (5.2) 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 (5.2) is estimated by choosing $\kappa = 1 - x^2$ in the weighted Laplacian operator $\mathcal{L}_2$ in (2.4) using the GPDM method. At the boundaries $x = \pm 1$, $\mathcal{L}$ 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 that is estimated using a finite-difference method. In particular, we construct an $N \times N$ diffusion matrix on N equally spaced discrete grids $\{x_i = 2(i-1)/(N-1) - 1\}_{i=1,\ldots,N}$ 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 [28]. The bandwidth $\epsilon = 1.5 \times 10^{-5}$ is chosen for $N = 400$ by the auto-tuning algorithm discussed in Section 2.

Figure 5.1 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 Figure 5.1 that both eigenvalues and eigenfunctions can be well approximated within numerical accuracy. For a detailed inspection, we show the errors of the eigenvalues and the eigenfunctions as functions of the mode-k, respectively, for the different number of points N in Figure 5.2. 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.

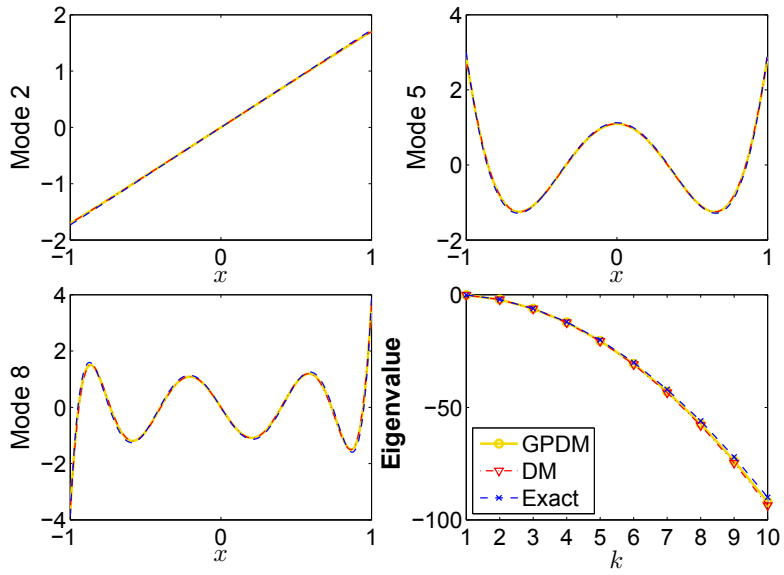


FIGURE 5.1. DM and GPDM estimation of eigenvalues and eigenfunctions for the Legendre polynomials on flat domain $[-1, 1]$.

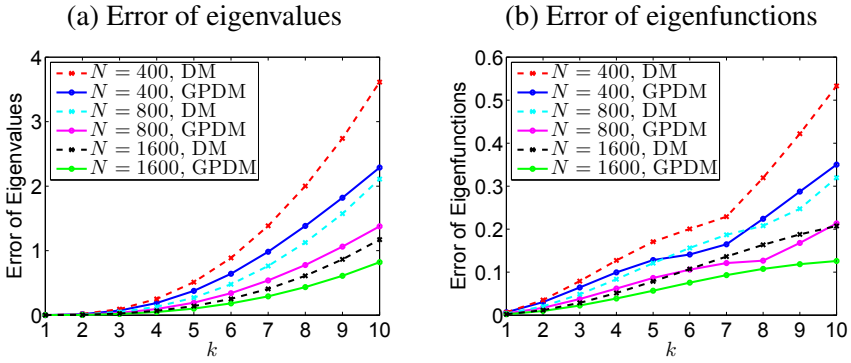


FIGURE 5.2. Sturm-Liouville problem: Error of (a) eigenvalues and (b) eigenfunctions as functions of the mode-k for different numbers of points.

5.2 Laplace-Beltrami operator on a semicircle

In this example, we consider solving the eigenvalue problem $\Delta\psi_k = \lambda_k\psi_k$ on a 1D semicircle with Dirichlet and Robin boundary conditions. We neglect to show results with the Neumann boundary condition since the performances of GPDM and DM are identical. The Riemannian metric of the semicircle is given by (2.12) 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

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

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,

$$\lambda_k = \begin{cases} 1 & \text{for } k = 1, \\ -(k-1)^2 & \text{for } k = 2, 3, \ldots, \end{cases}$$

$$\psi_k = \begin{cases} \exp(-x) & \text{for } k = 1, \\ \sin((k-1)x) - (k-1)\cos((k-1)x) & \text{for } k = 2, 3, \ldots. \end{cases}$$

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 (2.3). We construct an $N \times N$ matrix on N equally spaced discrete grids $\{x_i = (\cos((i-1)\pi/(N-1)), \sin((i-1)\pi/(N-1)))\}_{i=1,\ldots,N}$. 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 Figure 5.3. 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).

Figures 5.4(a),(b) show errors of the eigenvalues and eigenfunctions, respectively, as functions of mode-k for a different number of points N on a semicircle example with the Robin boundary condition. Figures 5.4(c),(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.

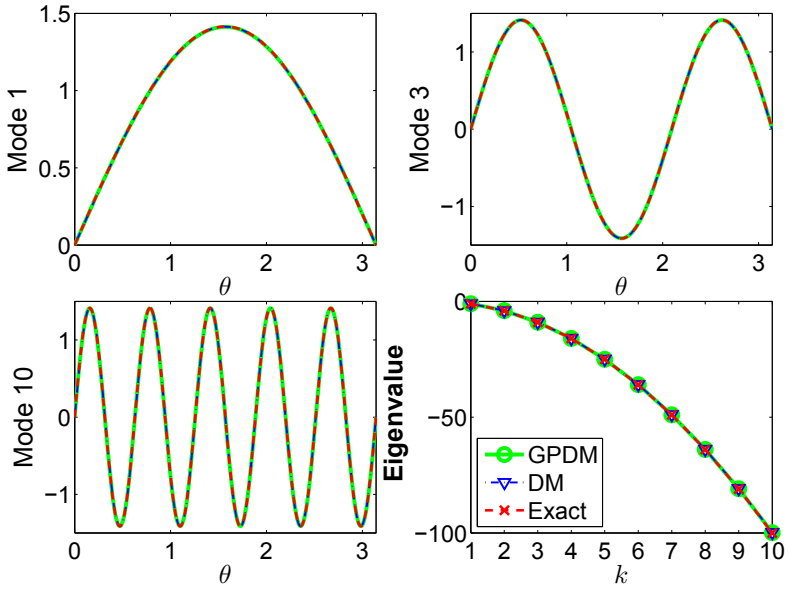
5.3 Laplace-Beltrami operator on a semitorus

In this example, we consider solving the eigenvalue problems $\Delta\psi_k = \lambda_k \psi_k$ on a 2D semitorus embedded in $\mathbb{R}^3$ with Dirichlet and Dirichlet-Neumann mixed boundary conditions. Here, the torus is defined with the standard embedding function (3.2) with the Riemannian metric (3.3), 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:

$$\Delta\psi_k = \frac{1}{(a + \cos\theta)^2} \frac{\partial^2 \psi_k}{\partial \phi^2} + \frac{\partial^2 \psi_k}{\partial \theta^2} - \frac{\sin\theta}{a + \cos\theta} \frac{\partial \psi_k}{\partial \theta} = \lambda_k \psi_k. \tag{5.3}$$

We can use the method of separation of variables to solve this eigenvalue problem (5.3), satisfying Dirichlet and the mixed boundary conditions. That is, we set

(a) Dirichlet



(b) Robin

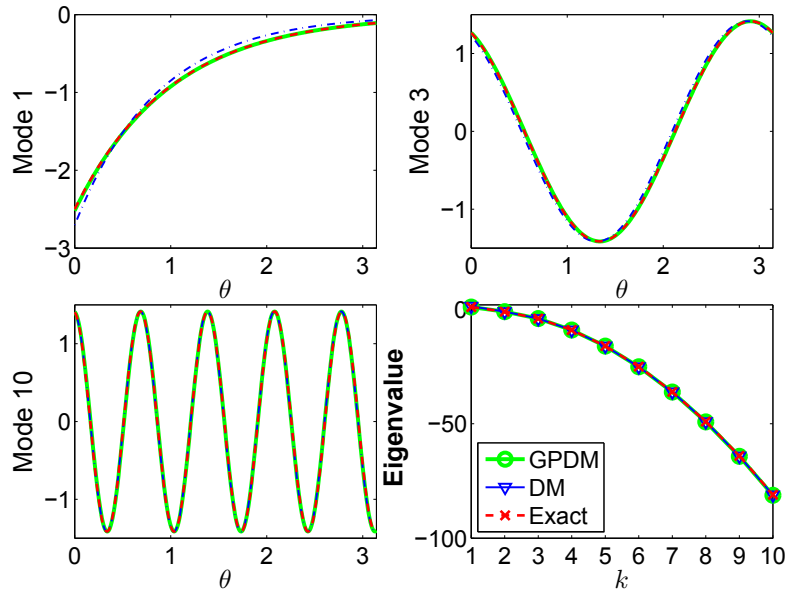


FIGURE 5.3. Comparisons of eigenvalues and eigenfunctions between DM and GPDM for semicircle example with (a) Dirichlet and (b) Robin boundary conditions. The Riemannian metric is given by (2.12) with $a = 1$.

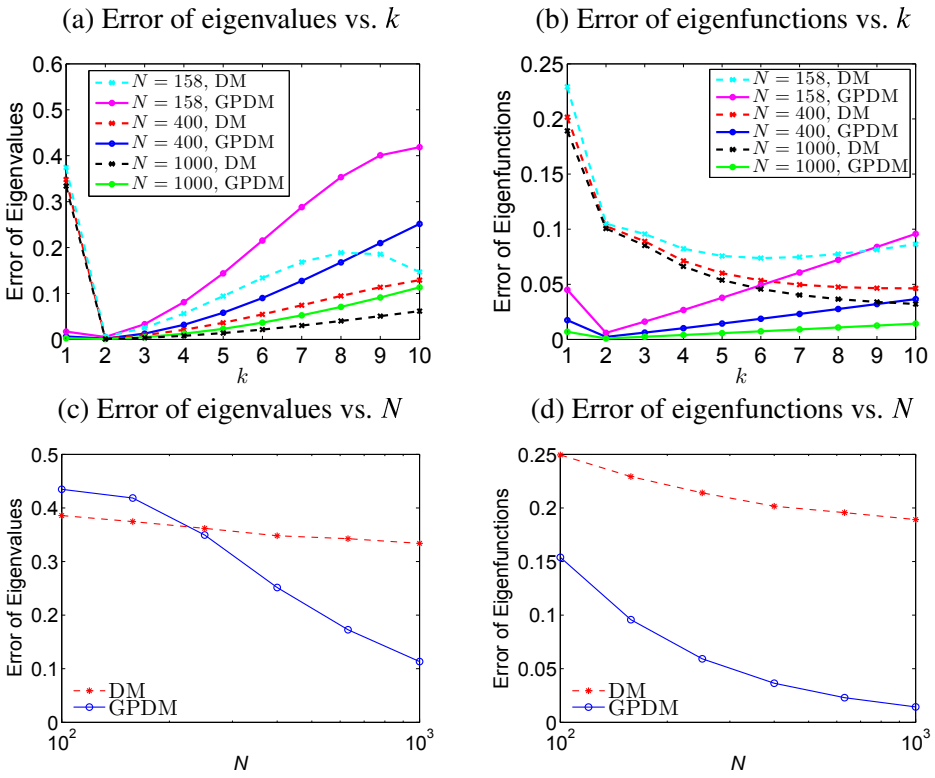


FIGURE 5.4. Error of (a) eigenvalues and (b) eigenfunctions as functions of k for different number of points N for the semicircle example with the Robin boundary condition.Error of (c) eigenvalues and (d) eigenfunctions vs. N. Note that for DM, there is no convergence for the leading eigenvalues and eigenfunctions.

$\psi_k = \Phi_k(\phi)\Theta_k(\theta)$ and substitute ψ_k back into (5.3) to deduce the eigenvalue problems for Φ_k and Θ_k:

$$\Phi_k'' + m_k^2 \Phi_k = 0, \tag{5.4}$$

$$\Theta_k'' - \frac{\sin\theta}{a+\cos\theta}\Theta_k' - \frac{m_k^2}{(a+\cos\theta)^2}\Theta_k = \lambda_k \Theta_k, \tag{5.5}$$

where the derivatives in (5.4) and (5.5) are taken with respect to ϕ and θ, respectively. The discrete values of m_k are chosen such that Φ_k satisfies (5.4) 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 (5.5) 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 (5.3).

In our numerical implementation, the grid points $\{\theta_i, \phi_j\}$ are uniformly distributed on $[0, 2\pi] \times [0, \pi]$ with $i, j = 1, \ldots, 64$ points in each direction resulting in a total of $N = 4096$ grid points. We assume that we do not know the embedding function (3.2) when solving the eigenvalue problem using DM and GPDM. For the GPDM method, we estimate normal direction $\boldsymbol{\nu}$ to the boundary, add ghost points along $\boldsymbol{\nu}$, 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 $k = 200$ nearest neighbors to construct a sparse matrix $\mathbf{L}^h$ for computational efficiency. The kernel bandwidth $\epsilon = 0.004$ is auto-tuned for all types of boundary conditions.

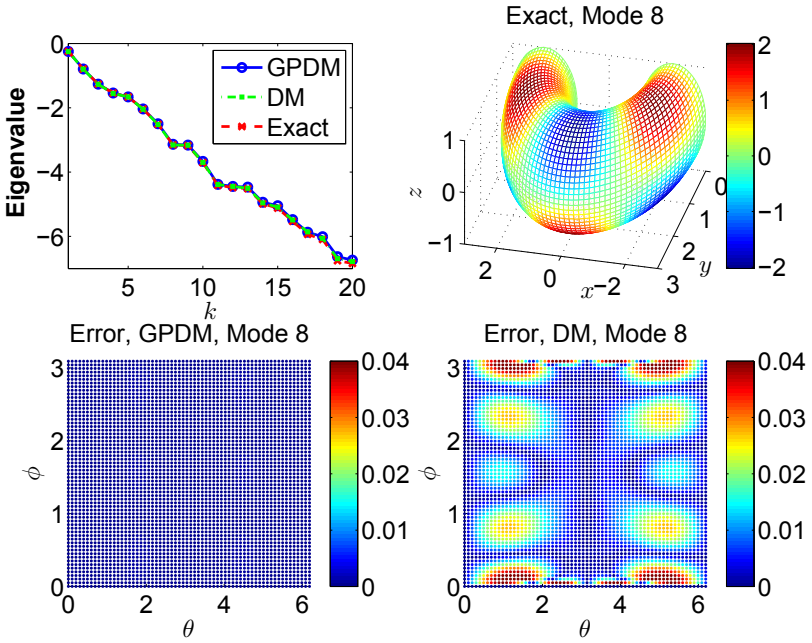
Figure 5.5 shows the numerical estimates of the first 20 eigenvalues and the eighth eigenfunction for (a) Dirichlet and (b) the mixed boundary conditions. One can see from Figure 5.5 that the eigenvalues and the eighth eigenfunction can be approximated well by both DM and GPDM. For the Dirichlet boundary condition, the largest errors of the first 20 eigenvalues are comparable as 0.08 and 0.12 using the standard DM and GPDM, respectively. The largest ℓ^∞-norm error of the first 20 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 20 eigenvalues are comparable as 0.06 and 0.04 using the standard DM and GPDM, respectively. The largest ℓ^∞-norm error of the first 20 eigenfunctions using GPDM ($= 0.99$) is comparable to that using DM ($= 1.03$). However, a close inspection, e.g., the eighth eigenfunctions, suggests that the GPDM errors occur on smaller regions of the domain compared 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 boundaries. The proposed method overcomes the inconsistency of the diffusion maps (DM) algorithm in estimating these differential operators near the boundaries. We provided a theoretical convergence study as well as numerical verification on test problems with tractable solutions and 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 boundary conditions on the flat domain. Our key contribution is to realize this idea with a concrete numerical algorithm on unknown manifolds, identified only by the point clouds, that is guaranteed to be consistent.

We considered solving elliptic PDEs (4.1) with the GPDM 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 a problem with an unknown

(a) Dirichlet



(b) Dirichlet-Neumann mixed

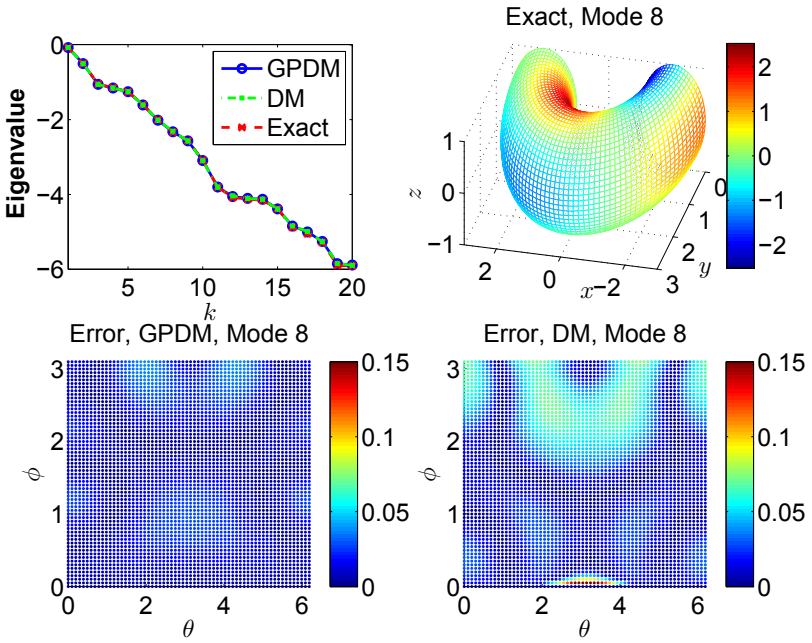


FIGURE 5.5. Comparisons of eigenvalues and eigenfunctions with (a) Dirichlet and (b) Dirichlet-Neumann mixed boundary conditions on the semitorus example with Riemannian metric (3.3) with $a = 2$.

manifold where the explicit solution is unknown, we compare the result against the FEM solution. Overall, GPDM 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 [25]. Numerically, we also found that GPDM is more accurate compared to DM in solving eigenvalue problems associated to the operators (2.3)–(2.4).

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

- 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 significant if the curvature is very large at the boundary. This suggests that the method can be improved by specifying ghost points that account for the curvature at the boundary. In our context, where the manifold is unknown, this requires an estimation of the boundary curvature from the point clouds, which is a problem that we are not currently familiar with.
- In this work, we have verified the proposed method on 1D and 2D manifolds. For higher-dimensional manifolds, while the numerical method can be used, the conditions to achieve the conclusion in the Lemma 3.5 require further studies.
- The proposed technique assumed that the boundary points are given. In the case of well-sampled data, the number of points at the boundary is specified explicitly, $J = N^{1/d}$. For the randomly sampled data, when we employ the local kernel, the auto-tuned ϵ yields error rates of $\epsilon_1 \sim N^{-1/2}$ and $\epsilon_2 \sim J^{-1}$. To have a balanced error, $\epsilon_1 \sim \epsilon_2$, we require $J = N^{1/2}$, as we numerically verified on 2D examples. If this scaling is valid for arbitrary dimensions, that is, $\epsilon_1 \sim N^{-1/d}$ and $\epsilon_2 \sim J^{-1/(d-1)}$, then the number of points at the boundary required to achieve balanced error rates of order $N^{-1/d}$ is $J = N^{(d-1)/d}$. While this estimate seems to indicate a severe limitation of this method, intuitively this is consistent with the well-known fact that the distribution of high-dimensional random variables on a bounded domain tends to lie near the boundary. Further investigation is required to understand this thoroughly.
- While the numerical demonstration showed convincing results in solving eigenvalue problems, spectral convergence and the error estimate of the eigenfunctions are not known. One possible avenue is to extend the result in [8, 12, 13, 18, 24] to manifolds with general boundary conditions. In this direction, a result for the Neumann boundary condition was recently reported in [37].

Appendix A Proof of Proposition 3.6

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

From previous results [6, 15, 28, 29, 48], we have the pointwise error estimation under these three situations: (1) on manifolds without boundary, (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 the reader's convenience, we quote the following error estimation based on the third situation.

LEMMA A.1 (Pointwise forward error estimate). *Let $M \cup \Delta M$ be a smooth d-dimensional manifold embedded in $\mathbb{R}^n$. Let the assumptions in Proposition* 3.6 *for the extended manifold $M \cup \Delta M$ and $x \in M$ hold. 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*

$$\vec{u} = (u(x_1), \dots, u(x_N), u(x_1^{G_1}), \dots, u\big(x_J^{G_K}\big))^\top \in \mathbb{R}^{\bar{N}}.$$

Then for $i = 1, \dots, N$ and $j' = 1, 3$,

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

(A.2)
$$\left|(\mathbf{L}_2\vec{u})_i - \mathcal{L}_2u(x_i)\right| = \mathcal{O}\left(\epsilon, \frac{q(x_i)^{1/2}}{\sqrt{\bar{N}}\epsilon^{2+d/4}}, \frac{\left|\nabla(\sqrt{\kappa(x_i)}u(x_i))\right|q(x_i)^{-1/2}}{\sqrt{\bar{N}}\epsilon^{1/2+d/4}}\right)$$

in high probability as $\epsilon \to 0$ after $\bar{N} \to \infty$. 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, $\widetilde{g}(u, v) := g(c^{-1/2}u, c^{-1/2}v)$ for all $u, v \in T_x(M \cup \Delta M)$, where c denotes the symmetric positive definite diffusion tensor.

In (A.1)–(A.2), the first error term, which is valid as $\epsilon \to 0$, is due to the continuous asymptotic expansion in (2.3), (2.4), and (2.9). The second error term is due to the estimation of the sampling density through (A.3), and the final error term is the bias induced by the discrete estimator; both of these are valid as $\bar{N} \to \infty$ and fixed $\epsilon > 0$.

PROOF. The proofs for the cases $j' = 1$ and 3 are readily available in [6, 15, 28, 29, 48]. For $j' = 2$, the proof follows directly the steps in appendix A of [29] 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

$$\widehat{q}_\epsilon(x_j) := \frac{\epsilon^{d/2}}{\bar{N}} \sum_{i=1}^{\bar{N}} \mathbf{K}_{ji}, \tag{A.3}$$

as an estimator to the sampling density of the data $q(x_j)$. With this definition, we define

$$F_i(x_j) := \frac{K(\epsilon, x_i, x_j)\sqrt{\kappa(x_j)}u(x_j)}{\widehat{q}_\epsilon(x_j)} \quad \text{and} \quad G_i(x_j) := \frac{K(\epsilon, x_i, x_j)\sqrt{\kappa(x_j)}}{\widehat{q}_\epsilon(x_j)}.$$

Following exactly the steps in the proof in [29] with the asymptotic expansion in (2.2), one obtains the error estimate in (A.2) for a discrete estimator that converges to $\kappa^{-1}\mathcal{L}_2$. Thus, the error for estimating $\mathcal{L}_2$ is similar to that of $\kappa^{-1}\mathcal{L}_2$ 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 (2.10)). □

Now, we will deduce several intermediate results that will simplify the proof of Proposition 3.6. For the discussion below, we define the matrix,

$$\mathbf{L}^h = \frac{1}{\epsilon}(\widetilde{\mathbf{D}} - \mathbf{I}) := \frac{1}{\epsilon}((\mathbf{D}^h)^{-1}\mathbf{K}^h - \mathbf{I}), \tag{A.4}$$

obtained from the standard diffusion maps as a discrete approximation to one of the diffusion operators in (2.13) with the entries of $\widetilde{\mathbf{D}}$ denoted by $\widetilde{D}_{i,j}$. As we discussed in Section 3.4, the matrix $\mathbf{L}^h$ is a discrete approximation to one of the diffusion operators in (2.3)–(2.5) with the following important modification. We construct the matrix $\mathbf{L}^h$ by evaluating the kernel on

$$\{x_i\}_{i=1}^N \cup \{\widetilde{x}_j^{G_k}\}_{j,k=1}^{J,K},$$

where the interior ghost points are denoted as components of $\{x_i\}$, that is, $\{\widetilde{x}_j^{G_0}\} \subset \{x_i\}$. To be consistent with the notation in Section 3.4, we emphasize that $\mathbf{L}^h \in \mathbb{R}^{N\times\bar{N}}$ is a nonsquare matrix with $\bar{N} = N + JK$, where the N rows correspond to the kernel evaluation at $\{x_i\}_{i=1}^N$. Based on the discussion in Remark 3.8, we will only prove the next lemma for randomly sampled data, of which the error is mainly due to

$$\begin{aligned} \left|\gamma_j(h) - \widetilde{x}_j^{G_0}\right| &\le \left|\gamma_j(h) - x_j^{G_0}\right| + \left|x_j^{G_0} - \widetilde{x}_j^{G_0}\right| \\ &= \mathcal{O}(h^2) + h|\boldsymbol{\nu} - \widetilde{\boldsymbol{\nu}}| = \mathcal{O}(h\sqrt{\epsilon}). \\ \left|x_j^{G_k}, \widetilde{x}_j^{G_k}\right| &= kh|\boldsymbol{\nu} - \widetilde{\boldsymbol{\nu}}| = \mathcal{O}(h\sqrt{\epsilon}), \end{aligned} \tag{A.5}$$

for $j = 1, \ldots, J$ and $k = 1, \ldots, K$, as pointed out in Remark 3.7. In (A.5), we have used the fact that $|\boldsymbol{\nu} - \widetilde{\boldsymbol{\nu}}| = \mathcal{O}(\sqrt{\epsilon})$ for randomly sampled data. For convenience, we recall that $\gamma_j(h) := \exp_{\boldsymbol{x}_j^B}(-h\boldsymbol{\nu}_{x_j^B}) \in M$ (see Figure 3.2(a) for a geometric illustration).

Let $\mathbf{L}$ be defined as in (A.4) such that the matrix is constructed by evaluating the kernel on $\{x_i\} \cup \{x_j^{G_k}\}$, where we replace the interior ghost points $\{\widetilde{x}_j^{G_0}\}$ in the construction of $\mathbf{L}^h$ with the corresponding $\{\gamma_j(h)\} \in M$. Basically $\mathbf{L}$ is constructed based on points that lie on the extended manifold, $M \cup \Delta M$. With this construction, we have:

LEMMA A.2. *Let $\mathbf{L}^h$ and $\mathbf{L}$ be constructed as in the discussion above. Suppose that $x_j := \gamma_j(h)$ such that $|x_i - x_j| = \mathcal{O}(h)$ for all $i \neq j$. Then for $u \in C(B_{\epsilon^r}(\partial M))$,*

$$\sum_{j=1}^{\bar{N}} \mathbf{L}_{ij}^h \widetilde{u}_j = \sum_{j=1}^{\bar{N}} \mathbf{L}_{ij} u_j + \mathcal{O}(h^2\epsilon^{-3/2}, h\epsilon^{-1/2}), \tag{A.6}$$

as $h \to 0$ and fixed $\epsilon > 0$. Here, $\widetilde{u}_j$ and u_j are different only on the ghost points, particularly, when $\widetilde{u}_j = u(\widetilde{x}_j^{G_0})$, we have $u_j = u(\gamma_j(h)) = u(x_j)$. Also, when $\widetilde{u}_j = u(\widetilde{x}_j^{G_k})$, then $u_j = u(x_j^{G_k})$ for all $k = 1, \ldots, K$.

PROOF. Suppose we consider $x_j = \gamma_j(h)$ that satisfies (A.5). For $|x_i - x_j| = \mathcal{O}(h)$, one can show that

$$\begin{aligned}
\mathbf{K}_{ij}^h &:= \exp\left(-\frac{|x_i - \widetilde{x}_j^{G_0}|^2}{4\epsilon}\right) \\
&= \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right) \exp\left(-\frac{c|x_i - x_j|h\sqrt{\epsilon} + \mathcal{O}(h^2\epsilon)}{4\epsilon}\right) \\
&= \exp\left(-\frac{|x_i - x_j|^2}{4\epsilon}\right)(1 + \mathcal{O}(h^2\epsilon^{-1/2})).
\end{aligned}$$

Therefore,

$$\mathbf{D}_i^h := \sum_{j=1}^{\bar{N}} \mathbf{K}_{ij}^h = \sum_{j=1}^{\bar{N}} \mathbf{K}_{ij} + \mathcal{O}(h^2\epsilon^{-\frac{1}{2}}) := \mathbf{D}_i + \mathcal{O}(h^2\epsilon^{-\frac{1}{2}}),$$

where the constant in the big-oh notation absorbs the number of perturbed points, which is much smaller than k when the k-nearest neighbor summand is used. This means

$$\left(\mathbf{D}_i^h\right)^{-1} := \mathbf{D}_i^{-1}\left(1 - \mathbf{D}_i^{-1}\mathcal{O}(h^2\epsilon^{-1/2})\right).$$

and

$$
\begin{aligned}
&\sum_{j=1}^{\bar{N}} \mathbf{L}^h_{ij}\tilde{u}_j \\
&\quad := \epsilon^{-1}\Big(\big(\mathbf{D}^h_i\big)^{-1} \sum_{j=1}^{\bar{N}} \mathbf{K}^h_{ij}\tilde{u}_j - \tilde{u}_i \Big) \\
&\quad = \epsilon^{-1}\Big(\mathbf{D}_i^{-1}(1-\mathbf{D}_i^{-1}\mathcal{O}(h^2\epsilon^{-1/2})) \sum_{j=1}^{\bar{N}} \mathbf{K}^h_{ij}\tilde{u}_j - \tilde{u}_i \Big) \\
&\quad = \epsilon^{-1}\Big(\mathbf{D}_i^{-1}(1-\mathbf{D}_i^{-1}\mathcal{O}(h^2\epsilon^{-1/2}))\Big(\sum_{j=1}^{\bar{N}} \mathbf{K}_{ij}\tilde{u}_j + \sum_{j=1}^{\bar{N}} \tilde{u}_j \mathcal{O}(h^2\epsilon^{-1/2})\Big) - \tilde{u}_i \Big) \\
&\quad = \epsilon^{-1}\Big(\mathbf{D}_i^{-1} \sum_{j=1}^{\bar{N}} \mathbf{K}_{ij}\tilde{u}_j - \tilde{u}_i + \mathbf{D}_i^{-1} \sum_{j=1}^{\bar{N}} \tilde{u}_j \mathcal{O}(h^2\epsilon^{-1/2}) \\
&\qquad\qquad - \mathbf{D}_i^{-1} \sum_{j=1}^{\bar{N}} \mathbf{K}_{ij}\tilde{u}_j \mathbf{D}_i^{-1}\mathcal{O}(h^2\epsilon^{-1/2}) \Big) \\
&\quad = \sum_{j=1}^{\bar{N}} \mathbf{L}_{ij}\tilde{u}_j + \mathbf{D}_i^{-1}\Big(\sum_{j=1}^{\bar{N}} (1-\mathbf{D}_i^{-1}\mathbf{K}_{ij})\tilde{u}_j \Big)\mathcal{O}(h^2\epsilon^{-3/2})\big) \\
&\quad = \sum_{j=1}^{\bar{N}} \mathbf{L}_{ij} u_j + \mathcal{O}(h^2\epsilon^{-3/2}, h\epsilon^{-1/2}),
\end{aligned}
$$

where in the last equality, we have used the fact $u_j - \tilde{u}_j = \mathcal{O}(h\epsilon^{1/2})$ on the estimated ghost points due to (A.5), $u \in C^1(B_{\epsilon^r}\partial M)$, and $1 - \mathbf{D}_i^{-1}\mathbf{K}_{ij} \le 1$ and $\mathbf{D}_i \ge 1$ such that $\mathbf{D}_i^{-1}\big(\sum_{j=1}^{\bar{N}} (1-\mathbf{D}_i^{-1}\mathbf{K}_{ij})\tilde{u}_j \big) \le |\tilde{u}|$, where $\tilde{u} \in \mathbb{R}^{\bar{N}}$ is defined below in (A.7). □

The assumption that $|x_i - x_j| = \mathcal{O}(h)$, where $x_j := \gamma_j(h)$ is rather natural in the numerical implementation with the k-nearest neighbor, even if there are many other points x_i that are further away with distance of order-$\sqrt{\epsilon}$. In particular, our construction is such that the perturbed points $\{\tilde{x}_j^{G_0}\}$ are defined to be of order-h away from each boundary point and the corresponding ghost points $\{\tilde{x}_j^{G_k}\}$ as defined in (3.12). Therefore, when the k-nearest neighbor is used in constructing the matrix $\mathbf{L}^h$, then these estimated ghost points either belong to the k-nearest sets of other points whose distance are of order-h or they have a k-nearest neighbor of mostly points of order-h away.

Since $h = \mathcal{O}(\epsilon)$ for the randomly sampled data case, the two error bounds are equivalent. Next, we define the column vector

$$\begin{aligned}\widetilde{u} &:= (\widetilde{u}_1, \dots, \widetilde{u}_{\bar{N}}) \\ &= \big(u(x_1), \dots, u\big(\widetilde{x}_j^{G_0}\big), \dots, u(x_N), u\big(\widetilde{x}_1^{G_1}\big), \dots, u\big(\widetilde{x}_J^{G_K}\big)\big) \in \mathbb{R}^{\bar{N}}.\end{aligned} \tag{A.7}$$

For $\vec{u}$ as defined in (3.17), the error rate in (A.6) can be written in a compact form as

$$\mathbf{L}^h \widetilde{u} = \mathbf{L}\vec{u} + \mathcal{O}(h^2 \epsilon^{-3/2}). \tag{A.8}$$

This error also implies

$$\mathbf{L}^h \vec{u} = \mathbf{L}\vec{u} + \mathcal{O}(h^2 \epsilon^{-3/2}), \tag{A.9}$$

since $\widetilde{u} - \vec{u} = \mathcal{O}(h\epsilon^{1/2})$.

Next, we will deduce the consistency of the extrapolation formula (3.14). For this purpose, we define $\vec{u}_\epsilon = (u(x_1), \dots, u(\widetilde{x}_j^{G_0}), \dots, u(x_N), \widetilde{u}_{\epsilon,1}^{G_1}, \dots, \widetilde{u}_{\epsilon,J}^{G_K})^\top$ as in (3.13). Here, $u(\widetilde{x}_j^{G_0})$ replaces $u(\gamma_j(h))$ in the first N-terms of $\vec{u}$ as in (3.17).

LEMMA A.3 (Consistency of the extrapolation formula). *Under the assumptions of Proposition* 3.6*, for each boundary point $x_j^B \in \partial M$, the truncation error for the first equation in the extrapolation formula* (3.14) *is given by*

$$\begin{aligned}&\Big| \sum_{j',k=1}^{J,K} \widetilde{D}_{B_j,(N+(j'-1)K+k)} \Big(u\big(\widetilde{x}_{j'}^{G_k}\big) - \widetilde{u}_{\epsilon,j'}^{G_k} \Big) \Big| \\ &\quad = \epsilon \big| \big(\mathbf{L}^h \widetilde{u}\big)_{B_j} - \big(\mathbf{L}^h \vec{u}_\epsilon\big)_{B_j} \big| \\ &\quad = \mathcal{O}\big(\epsilon(h^2\epsilon^{-3/2}, \epsilon, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)})\big),\end{aligned} \tag{A.10}$$

in high probability as $\epsilon \to 0$ after $\bar{N} \to \infty$ and $h \to 0$. For the last three equations in (3.14)*, we have*

$$\begin{aligned}\big|\big(u\big(\widetilde{x}_j^{G_2}\big) - 3u\big(\widetilde{x}_j^{G_1}\big)\big) - \big(\widetilde{u}_{\epsilon,j}^{G_2} - 3\widetilde{u}_{\epsilon,j}^{G_1}\big)\big| &= \mathcal{O}(h^3), \\ \big|\big(u\big(\widetilde{x}_j^{G_3}\big) - 3u\big(\widetilde{x}_j^{G_2}\big) + 3u\big(\widetilde{x}_j^{G_1}\big)\big) - \big(\widetilde{u}_{\epsilon,j}^{G_3} - 3\widetilde{u}_{\epsilon,j}^{G_2} + 3\widetilde{u}_{\epsilon,j}^{G_1}\big)\big| &= \mathcal{O}(h^3), \\ \big|\big(u\big(\widetilde{x}_j^{G_k}\big) - 3u\big(\widetilde{x}_j^{G_{k-1}}\big) + 3u\big(\widetilde{x}_j^{G_{k-2}}\big) - u\big(\widetilde{x}_j^{G_{k-3}}\big)\big) & \\ - \big(\widetilde{u}_{\epsilon,j}^{G_k} - 3\widetilde{u}_{\epsilon,j}^{G_{k-1}} + 3\widetilde{u}_{\epsilon,j}^{G_{k-2}} - \widetilde{u}_{\epsilon,j}^{G_{k-3}}\big)\big| &= \mathcal{O}(h^3),\end{aligned} \tag{A.11}$$

for $k = 4, \dots, K$.

PROOF. First, let us proof (A.10). For this case, we only consider the B_j th row corresponding to the boundary point x_j^B. In the case of randomly sampled data, we

have

$$\begin{aligned}
&\left|(\mathbf{L}^h\widetilde{u})_{B_j} - (\mathbf{L}^h\vec{u}_\epsilon)_{B_j}\right| \\
&\quad\le \left|(\mathbf{L}^h\widetilde{u})_{B_j} - (\mathbf{L}\vec{u})_{B_j}\right| + \left|(\mathbf{L}\vec{u})_{B_j} - (\mathbf{L}^h\vec{u}_\epsilon)_{B_j}\right| \\
&\quad= \left|(\mathbf{L}^h\widetilde{u})_{B_j} - (\mathbf{L}\vec{u})_{B_j}\right| + \left|(\mathbf{L}\vec{u})_{B_j} - \mathcal{L}u(x_j^B)\right| \\
&\quad= \mathcal{O}(h^2\epsilon^{-3/2}) + \mathcal{O}\big(\epsilon, \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}\big),
\end{aligned} \tag{A.12}$$

where we have used the fact that $\mathcal{L}u(x_j^B) = f(x_j^B)$ and $(\mathbf{L}^h\vec{u}_\epsilon)_{B_j} = f(x_j^B)$, which is the first equation of the extrapolation formula in (3.14) in deducing the third line above. To obtain the fourth line, we directly used (A.8) for the first bound and Lemma A.1 for the second error bound, 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 (A.1) and (A.2) to simplify the discussion.

For the well-sampled data, based on the discussion in Remark 3.8, the first error term in (A.12) is not applicable and we treat $\mathbf{L}^h$ as $\mathbf{L}$. Since the first N components of $\widetilde{u}_i$ (see (A.7)) are equal to the components of $\vec{u}_\epsilon^M$ defined in (3.16) and the identity $\mathbf{I}$ only contributes to the coefficient of $u(x_j^B) = \widetilde{u}_{\epsilon,j}^{B_j}$ for the boundary point x_j^B, we can simplify the left-hand side of (A.12) as

$$\begin{aligned}
\left|(\mathbf{L}^h\widetilde{u})_{B_j} - (\mathbf{L}^h\vec{u}_\epsilon)_{B_j}\right| &= \frac{1}{\epsilon}\left|(\widetilde{\mathbf{D}}\widetilde{u})_{B_j} - \big(\widetilde{\mathbf{D}}\vec{u}_\epsilon\big)_{B_j}\right| \\
&= \frac{1}{\epsilon}\left|\sum_{j',k=1}^{J,K} \widetilde{D}_{B_j,(N+(j'-1)K+k)}\Big(u\big(\widetilde{x}_{j'}^{G_k}\big) - \widetilde{u}_{\epsilon,j'}^{G_k}\Big)\right|.
\end{aligned} \tag{A.13}$$

Thus, from (A.12) and (A.13), we obtain the result in (A.10).

The proof for (A.11) is straightforward. In particular, for each $j = 1, \ldots, J$, note that $\{\widetilde{x}_j^{G_0}, x^B, \widetilde{x}_j^{G_1}, \ldots, \widetilde{x}_j^{G_K}\}$ are points that lie on a straight line in the direction of $\widetilde{\nu}$ where the distances between the consecutive points are identical, namely, h. For $u \in C^3(M \cup B_{\epsilon^r}(\partial M))$, where $B_{\epsilon^r}(\partial M) \supset \Delta M$ is as in Definition 3.4, then one can deduce (A.11) by employing the standard Taylor's expansion on the interval $[\widetilde{x}_j^{G_0}, \widetilde{x}_j^{G_K}]$. □

Proof of Proposition 3.6. Notice that we can write (A.10) and (A.11) in Proposition A.3 in a matrix form,

$$\mathbf{E}\delta\vec{u}_\epsilon^G = \mathcal{O}\big(h^3, h^2\epsilon^{-1/2}, \epsilon^2, \bar{N}^{-1/2}\epsilon^{-(1+d/4)}, \bar{N}^{-1/2}\epsilon^{(1/2-d/4)}\big), \tag{A.14}$$

where $\delta\vec{u}_\epsilon^G = (|\widetilde{u}_{\epsilon,1}^{G_1} - u(\widetilde{x}_1^{G_1})|, \ldots, |\widetilde{u}_{\epsilon,J}^{G_K} - u(\widetilde{x}_J^{G_K})|)^\top$ and the matrix $\mathbf{E}$ is of size $JK \times JK$. We first show the stability of $\mathbf{E}$, namely $\mathbf{E}$ is invertible with uniformly bounded inverse. To simplify the discussion, we set $J = 1$ to correspond to a boundary point. One can use the same idea for the case of $J > 1$.

In this case, the matrix $\mathbf{E}$ in (A.14) is given by

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

We can obtain the uniform error between $\vec{u}$ and $\vec{u}_\epsilon$ in (A.14) once showing that $\|\mathbf{E}^{-1}\|_\infty < C$. We have the following decomposition for matrix $\mathbf{E}$, $\mathbf{E} = \mathbf{E}_0 + \mathbf{v}_1\mathbf{v}_2^\top$,

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

By induction, one can show that

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

so that $\|\mathbf{E}_0^{-1}\|_\infty = K(K+1)(K+2)/6 < C$ by noticing that K is always fixed to be less than 10 even when $N \to \infty$. One can calculate that $\mathbf{E}_0^{-1}\mathbf{v}_1 = (1, 3, 6, \ldots, K(K+1)/2)^\top$ and $1 + \mathbf{v}_2^\top\mathbf{E}_0^{-1}\mathbf{v}_1 = \sum_{k=1}^K k(k+1)\widetilde{D}_{B_1,(N+k)}/2$, which is nonzero. Thus, according to the Sherman-Morrison formula, we have

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

Inverting $\mathbf{E}$ in (A.14), for each $j = 1, \ldots, J$, $k = 1, \ldots, K$, we have

$$\begin{aligned} &\left| \widetilde{u}_{\epsilon,1}^{G_k} - u\left(\widetilde{x}_j^{G_k}\right) \right| \\ &\quad = \left(\delta\vec{u}_\epsilon^G\right)_{j,k} \\ &\quad = \mathcal{O}\left(h^3, h^2\epsilon^{-1/2}, \epsilon^2, \bar{N}^{-1/2}\epsilon^{-(1+d/4)}, \bar{N}^{-1/2}\epsilon^{(1/2-d/4)}\right), \end{aligned}$$

and the proof is complete by comparing these error rates with $|u(x_j^{G_k}) - u(\tilde{x}_j^{G_k})| = \mathcal{O}(h\sqrt{\epsilon})$.

Appendix B 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 [30]). That is, we will show that $\mathbf{L}^I$ in (4.8) 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 3.9. Before we proceed with these steps, let us first analyze the resulting GPDM estimator, $\mathbf{L}^g(\vec{u}^M) := (\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})\vec{u}^M + \mathbf{L}^{(2)}\vec{b}$, as defined in (3.20).

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 Figure 3.1(a)). The corresponding ghost points are $\tilde{x}_1^{G_0} = x_2$ and $\tilde{x}_2^{G_0} = x_{N-1}$, using the secant line approximation. Otherwise, the same analysis can be carried by relabeling $\tilde{x}_j^{G_0}$ by other arbitrary x_i. The last three equations in (3.14) can be written as

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

Using the same notation as in (A.4), we let $\mathbf{L}^h = (\tilde{\mathbf{D}} - \mathbf{I})/\epsilon := ((\mathbf{D}^h)^{-1}\mathbf{K}^h - \mathbf{I})/\epsilon$ be the $N \times \bar{N}$ matrix, where $\bar{N} = N + 2K$, obtained from the standard diffusion maps as a discrete approximation to one of the diffusion operators in (2.3)–(2.5) with the entries of $\tilde{\mathbf{D}}$ denoted by $\tilde{D}_{i,j}$. To be consistent with the notation in Section 3.4, we emphasize that $\mathbf{L}^h \in \mathbb{R}^{N\times\bar{N}}$ is a nonsquare matrix with $\bar{N} = N + 2K$, where the N-rows correspond to the kernel evaluation at the points $\{x_i \in M\}_{i=1}^N$.

Then, the i^{th} component of $(\tilde{\mathbf{D}} - \mathbf{I})\vec{u}_\epsilon$ is given by

$$\text{(B.2)} \qquad \begin{aligned} &\sum_{j=1}^{N+2K} \tilde{D}_{i,j}u_j - u_i \\ &\quad = \sum_{j=3}^{N-2} \tilde{D}_{i,j}u_j - u_i + \tilde{D}_{i,1}u_1 + \tilde{D}_{i,2}u_2 + \tilde{D}_{i,N-1}u_{N-1} \\ &\qquad + \tilde{D}_{i,N}u_N + \sum_{j,k=1}^{2,K} \tilde{D}_{i,N+(j-1)K+k}\tilde{u}_{\epsilon,j}^{G_k} = \end{aligned}$$

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

where we have defined

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

for convenience. From the first equation in (3.14), we have

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

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

$$
\begin{aligned}
\widetilde{u}_{\epsilon,1}^{G_1} &= \frac{1}{c_{1,0}} \Bigg(\epsilon f(x_1) - \sum_{j=3}^{N-2} \widetilde{D}_{1,j} u_j + (1 - c_{1,1}) u_1 - c_{1,2} u_2 \\
&\qquad - c_{1,N-1} u_{N-1} - c_{1,N} u_N \Bigg), \\
\widetilde{u}_{\epsilon,2}^{G_1} &= \frac{1}{c_{N,N+1}} \Bigg(\epsilon f(x_N) - \sum_{j=3}^{N-2} \widetilde{D}_{N,j} u_j - c_{N,1} u_1 - c_{N,2} u_2 \\
&\qquad - c_{N,N-1} u_{N-1} + (1 - c_{N,N}) u_N \Bigg).
\end{aligned} \tag{B.4}
$$

We should point out that equations (B.4) and (B.1) are components of (3.19). Therefore, the i^{th} row in (B.2) becomes

$$
\begin{aligned}
&\sum_{j=1}^{N+2K} \tilde{D}_{i,j}u_j - u_i \\
&= \sum_{j=3}^{N-2} \tilde{D}_{i,j}u_j - u_i + c_{i,1}u_1 + c_{i,2}u_2 + c_{i,N-1}u_{N-1} + c_{i,N}u_N \\
&\quad + \frac{c_{i,0}}{c_{1,0}}\Big(\epsilon f(x_1) - \sum_{j=3}^{N-2} \tilde{D}_{1,j}u_j + (1-c_{1,1})u_1 - c_{1,2}u_2 - c_{1,N-1}u_{N-1} - c_{1,N}u_N\Big) \\
&\quad + \frac{c_{i,N+1}}{c_{N,N+1}}\Big(\epsilon f(x_N) - \sum_{j=3}^{N-2} \tilde{D}_{N,j}u_j - c_{N,1}u_1 - c_{N,2}u_2 - c_{N,N-1}u_{N-1} + (1-c_{N,N})u_N\Big) \\
&= \sum_{j=3}^{N-2}\Big(\tilde{D}_{i,j} - \frac{c_{i,0}}{c_{1,0}}\tilde{D}_{1,j} - \frac{c_{i,N+1}}{c_{N,N+1}}\tilde{D}_{N,j}\Big)u_j - u_i + \Big(c_{i,1} + \frac{c_{i,0}}{c_{1,0}}(1-c_{1,1}) - \frac{c_{i,N+1}}{c_{N,N+1}}c_{N,1}\Big)u_1 \\
&\quad + \Big(c_{i,2} - \frac{c_{i,0}}{c_{1,0}}c_{1,2} - \frac{c_{i,N+1}}{c_{N,N+1}}c_{N,2}\Big)u_2 + \epsilon\Big(\frac{c_{i,0}}{c_{1,0}}f(x_1) + \frac{c_{i,N+1}}{c_{N,N+1}}f(x_N)\Big) \\
&\quad + \Big(c_{i,N-1} - \frac{c_{i,0}}{c_{1,0}}c_{1,N-1} - \frac{c_{i,N+1}}{c_{N,N+1}}c_{N,N-1}\Big)u_{N-1} + \Big(c_{i,N} - \frac{c_{i,0}}{c_{1,0}}c_{1,N} + \frac{c_{i,N+1}}{c_{N,N+1}}(1-c_{N,N})\Big)u_N.
\end{aligned}
\tag{B.5}
$$

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

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

In fact, since $c_{i,2} > c_{1,2}$ and $c_{i,2} > c_{N,2}$ for $i = 2, \ldots, N-1$, 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,

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

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

are also strictly positive. Thus, all of the nondiagonal coefficients of (B.5) are strictly positive.

We should point out that the expression on the right-hand-side of (B.5) is nothing but the i^{th} row of the affine operator in (3.20), that is,

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

Let us denote $\mathbf{M} = \epsilon(\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})$. Notice that if $u_i = 1$ for all $i = 1, \ldots, N + 2K$, then from (B.3) and the fact that $\sum_{j=1}^{N+2K} \widetilde{D}_{i,j} = 1$, one can verify that $f(x_1) = f(x_N) = 0$, which means $(\mathbf{L}^{(2)}\vec{b})_i = 0$. Evaluating (B.5) at $u_i = 1$, one can see that

$$
\begin{aligned}
0 &= \sum_{j=1}^{N+2K} \widetilde{D}_{i,j} - 1 \\
&= \sum_{\substack{j=3 \\ j\neq i}}^{N-2} \left(\widetilde{D}_{i,j} - \frac{c_{i,0}}{c_{1,0}} \widetilde{D}_{1,j} - \frac{c_{i,N+1}}{c_{N,N+1}} \widetilde{D}_{N,j} \right) \\
&\quad + \left(\left(\widetilde{D}_{i,i} - \frac{c_{i,0}}{c_{1,0}} \widetilde{D}_{1,i} - \frac{c_{i,N+1}}{c_{N,N+1}} \widetilde{D}_{N,i} \right) - 1 \right) \\
&\quad + \left(c_{i,1} + \frac{c_{i,0}}{c_{1,0}}(1 - c_{1,1}) - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,1} \right) + \left(c_{i,2} - \frac{c_{i,0}}{c_{1,0}} c_{1,2} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,2} \right) \\
&\quad + \left(c_{i,N-1} - \frac{c_{i,0}}{c_{1,0}} c_{1,N-1} - \frac{c_{i,N+1}}{c_{N,N+1}} c_{N,N-1} \right) + \left(c_{i,N} - \frac{c_{i,0}}{c_{1,0}} c_{1,N} + \frac{c_{i,N+1}}{c_{N,N+1}}(1 - c_{N,N}) \right) \\
&= \sum_{\substack{j=3 \\ j\neq i}}^{N-2} \mathbf{M}_{i,j} + \mathbf{M}_{i,i} + \mathbf{M}_{i,1} + \mathbf{M}_{i,2} + \mathbf{M}_{i,N-1} + \mathbf{M}_{i,N},
\end{aligned}
\tag{B.6}
$$

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 $\vec{v} = (v(x_2), \ldots, v(x_{N-1})$ 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,

$$
\begin{aligned}
-\mathbf{M}_{i,i}\, v(x_i) = \sum_{\substack{j=2 \\ j\neq i}}^{N-1} \mathbf{M}_{i,j}\, v(x_j) - \epsilon(\mathbf{L}^I \vec{v})_i &\leq \sum_{\substack{j=2 \\ j\neq i}}^{N-1} \mathbf{M}_{i,j}\, v(x_j) \\
&\leq \left(\sum_{\substack{j=2 \\ j\neq i}}^{N-1} \mathbf{M}_{i,j} \right) v(x_i).
\end{aligned}
\tag{B.7}
$$

Here, we use the fact that the matrix $\epsilon\mathbf{L}^I$ (as defined in (4.7)) is nothing but the submatrix of $\mathbf{M}$, ignoring the first and N^{th} columns. From (B.6), $-\mathbf{M}_{i,i} = \sum_{j=1, j\neq i}^{N} \mathbf{M}_{i,j} > \sum_{j=2, j\neq i}^{N-1} \mathbf{M}_{i,j}$, which contradicts (B.7), 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,

$$
\max_{1 \leq j \leq N} v(x_j) = \{v(x_1), v(x_N)\}. \tag{B.8}
$$

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 $v(x) \le 0$. Also, since M is compact, it attains the global minimum on M. Define $v_s(x) := v(x) - v_{\min}$, where $v_{\min} = \min_{x\in M} v(x) \le 0$. Thus it is clear that $0 \le v_s(x) \le C_2 = |v_{\min}|$ 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 3.9), it is clear that for the column vector, $\vec{v}_s^M := (v_s(x_1), \vec{v}_s^I, v_s(x_N)) \in \mathbb{R}^N$, where $\vec{v}_s^I := (v_s(x_2), \ldots, v_s(x_{N-1}))$, we have $\left|\left(\mathbf{L}^g(\vec{v}_s^M)\right)_i - \mathcal{L}v_s(x_i)\right| \le c_1\delta$, where $\delta := \max\{h^3\epsilon^{-1}, h^2\epsilon^{-3/2}\}$. Notice that

$$\begin{aligned}\left|\mathcal{L}v_s(x_i) - \left(\mathbf{L}^g\left(\vec{v}_s^M\right)\right)_i\right| &= \left|\mathcal{L}v_s(x_i) - \left((\mathbf{L}^{(1)} + \mathbf{L}^{(2)}\mathbf{A})\vec{v}_s^M + \mathbf{L}^{(2)}\vec{b}\right)_i\right| \\ &= \left|\mathcal{L}v_s(x_i) - (\mathbf{L}^{(2)}\vec{b})_i - \left(\mathbf{L}^B\vec{g} + \mathbf{L}^I\vec{v}_s^I\right)_i\right|,\end{aligned} \tag{B.9}$$

where we have used the decomposition in (4.7) and the affine estimator (3.20). This means

$$\left(\mathbf{L}^I\vec{v}_s^I\right)_i \ge C - c_1\delta - (\mathbf{L}^{(2)}\vec{b})_i - (\mathbf{L}^B\vec{g})_i.$$

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

$$\begin{aligned}\left(\mathbf{L}^I\vec{v}_s^I\right)_i &\ge 2 - c_1\delta + \left(\|\mathbf{L}^{(2)}\vec{b}\|_\infty - (\mathbf{L}^{(2)}\vec{b})_i\right) + \left(\|\mathbf{L}^B\vec{g}\|_\infty - (\mathbf{L}^B\vec{g})_i\right) \\ &\ge 2 - c_1\delta \ge 0.\end{aligned}$$

Basically $0 \le v_s(x_i) \le C_2$ 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 (4.8), then for $\hat{u}^I$ that solves (4.8), we have

$$\mathbf{L}^I\left(\hat{u}^I + M\vec{v}_s^I\right) \ge \vec{f}^I - \mathbf{L}^B\vec{g} + (2 - c_1\delta)M \ge 0$$

for small enough δ, which depends on h and fixed $0 < \epsilon \ll 1$. By the discrete maximum principle in (B.8), it is clear that,

$$\begin{aligned}\max_{x_i\in M}\hat{u}^I \le \max_{x_i\in M}\left(\hat{u}^I + M\vec{v}_s^I\right) &\le \max_{x_i\in\partial M}\hat{u}^B + \max_{x_i\in\partial M} M\vec{v}_s^I \\ &\le \|\hat{u}^B\|_\infty + C_2\|\vec{f}^I - \mathbf{L}^B\vec{g}\|_\infty.\end{aligned}$$

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

$$\|\hat{u}^I\|_\infty \le \|\hat{u}^B\|_\infty + C_2\|\vec{f}^I - \mathbf{L}^B\vec{g}\|_\infty. \tag{B.10}$$

Convergence: Applying (B.10) on $\hat{u}^I - \vec{u}^I$, where components of $\vec{u}^I$ are the true solution of the PDE in (4.1) with Dirichlet boundary condition, we obtain

$$\|\hat{u}^I - \vec{u}^I\|_\infty \le \|\hat{u}^B - \vec{u}^B\|_\infty + C_2\|\vec{f}^I - \mathbf{L}^B\vec{g} - \mathbf{L}^I\vec{u}^B\|_\infty. \tag{B.11}$$

Using the same argument as in (B.9) and the error bound in Theorem 3.9, we immediately see the consistency of the estimator, that is,

$$\begin{aligned}
&\left| f(x_i) - (\mathbf{L}^{(2)}\vec{b})_i - (\mathbf{L}^B \vec{g} + \mathbf{L}^I \vec{u}^I)_i \right| \\
&\quad = \left| \mathcal{L}u(x_i) - \left(\mathbf{L}^g(\vec{u}^M)\right)_i \right| \\
&\quad = \mathcal{O}\big(h^3\epsilon^{-1}, h^2\epsilon^{-3/2}, h\epsilon^{-1/2}, \epsilon, \\
&\qquad\qquad \bar{N}^{-1/2}\epsilon^{-(2+d/4)}, \bar{N}^{-1/2}\epsilon^{-(1/2+d/4)}\big),
\end{aligned} \tag{B.12}$$

in high probability, where as $\epsilon \to 0$ after $\bar{N} \to \infty$ and $h \to 0$. Since $\vec{u}^B = \hat{u}^B = \vec{g}$, combining (B.11) and (B.12), the proof is completed.

Appendix C An Alternative Method for Estimating the Normal Derivatives

In this appendix, we discuss a method for estimating normal derivatives at the boundary of a 2D manifold that requires no specification of ghost points. This scheme is used for estimating the directional derivatives of Neumann or Robin boundary conditions used in the classical diffusion maps algorithm. Specifically, the normal derivatives are estimated as follows.

ALGORITHM C.1. *Assume that $\boldsymbol{\nu}$ is the exterior normal direction to the boundary ∂M and $\widetilde{\boldsymbol{\nu}}$ is its numerical estimate as defined in Section* 3.1 *at a boundary point $x^B \in \partial M$. Then, the normal derivative $\partial_{\boldsymbol{\nu}} u$ at x^B is estimated as follows*:

(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,*

$$\widetilde{\boldsymbol{\nu}}^L := \frac{x^L - x^B}{|x^L - x^B|} \quad \text{and} \quad \widetilde{\boldsymbol{\nu}}^R := \frac{x^R - x^B}{|x^R - x^B|}.$$

Here, x^L is the nearest point to x^B in the region such that the angle between $\widetilde{\boldsymbol{\nu}}^L$ and $-\widetilde{\boldsymbol{\nu}}$ satisfies $\Theta(\widetilde{\boldsymbol{\nu}}^L, -\widetilde{\boldsymbol{\nu}}) < \Theta_0$ (in our implementation, $\Theta_0 = \pi/4$). This basic argument also applies to x^R. Moreover, the "left" and "right" can be numerically distinguished by the negative inner product $\langle \widetilde{\boldsymbol{w}}^L, \widetilde{\boldsymbol{w}}^R \rangle < 0$ where $\widetilde{\boldsymbol{w}}^L$ and $\widetilde{\boldsymbol{w}}^R$ are components orthogonal to $-\widetilde{\boldsymbol{\nu}}$, that is, $\widetilde{\boldsymbol{w}}^L = \widetilde{\boldsymbol{\nu}}^L - (\widetilde{\boldsymbol{\nu}}^L \cdot \widetilde{\boldsymbol{\nu}})(\widetilde{\boldsymbol{\nu}})$ and $\widetilde{\boldsymbol{w}}^R = \widetilde{\boldsymbol{\nu}}^R - (\widetilde{\boldsymbol{\nu}}^R \cdot \widetilde{\boldsymbol{\nu}}))(\widetilde{\boldsymbol{\nu}})$.

(2) *Write $-\widetilde{\boldsymbol{\nu}}$ as a linear combination of $\widetilde{\boldsymbol{\nu}}^L$ and $\widetilde{\boldsymbol{\nu}}^R$ using the linear regression*

$$-\widetilde{\boldsymbol{\nu}} = \widetilde{a}^L \widetilde{\boldsymbol{\nu}}^L + \widetilde{a}^R \widetilde{\boldsymbol{\nu}}^R, \tag{C.1}$$

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

(3) *Estimate the normal derivative* $-\partial_{\boldsymbol{\nu}} u$ *numerically using the difference method,*

$$\begin{aligned}\frac{\partial u}{\partial(-\boldsymbol{\nu})}(x^B) &\approx \frac{\Delta u}{\Delta(-\widetilde{\boldsymbol{\nu}})}(x^B)\\ &:= \left[\widetilde{a}^L \frac{\Delta u}{\Delta \widetilde{\boldsymbol{\nu}}^L} + \widetilde{a}^R \frac{\Delta u}{\Delta \widetilde{\boldsymbol{\nu}}^R}\right](x^B)\\ &:= \widetilde{a}^L \frac{u(x^L) - u(x^B)}{|x^L - x^B|} + \widetilde{a}^R \frac{u(x^R) - u(x^B)}{|x^R - x^B|},\end{aligned} \tag{C.2}$$

where we have used equation (C.1) *and the fact that* $\widetilde{\boldsymbol{\nu}}$, $\widetilde{\boldsymbol{\nu}}^L$, *and* $\widetilde{\boldsymbol{\nu}}^R$ *are all unit vectors. Then, the normal derivative* $\partial_{\boldsymbol{\nu}} u$ *term in the boundary condition* (4.1) *in the following section can be numerically estimated using equation* (C.2) *for all points on the boundary.*

Next, we provide the error rate for estimating the directional derivative $\partial_{\boldsymbol{\nu}} u$ with equation (C.2) at the boundary points.

PROPOSITION C.2. *Let* $u \in C^3(M)$ *be a smooth function on a 2D manifold* M *with 1D boundary* ∂M. *Let* $\{x_1, \ldots, x_N\} \subset M$ *be a set of data points, among which some labeled points lie on the boundary* ∂M. *Let* x^B *be a boundary point on the 1D smooth* ∂M *and* $\boldsymbol{\nu}$ *be the unit exterior normal direction to the boundary* ∂M *at* x^B. *Let* x^L *and* $x^R \in \{x_1, \ldots, x_N\}$ *be the "left" and "right" nearest neighbors, respectively, for the boundary point* x^B. *Then, the normal derivative* $\partial_{\boldsymbol{\nu}} u$ *at* x_B *estimated by equation* (C.2) *in Algorithm* C.1 *has an error rate of*

$$\left|\frac{\partial u}{\partial \boldsymbol{\nu}}(x^B) - \frac{\Delta u}{\Delta \widetilde{\boldsymbol{\nu}}}(x^B)\right| = \mathcal{O}(h),$$

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,* $\left|\nabla_i \nabla_j u\left(x^B\right)\right|$ *with* $\nabla_i \nabla_j$ *being the Hessian operator*).

PROOF. The error has two parts, one from the regression coefficients $\widetilde{a}^L$ and $\widetilde{a}^R$, and the other from estimation of the directional derivatives $\partial_{\boldsymbol{\nu}^L} u$ and $\partial_{\boldsymbol{\nu}^R} u$. First, we estimate the error from the regression coefficients $\widetilde{a}^L$ and $\widetilde{a}^R$. Let $\gamma_L(\ell)$ be a geodesic parametrized with the arclength 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 $\boldsymbol{\nu}^L := \gamma_L'(0) \in T_{x^B} M$ as a unit tangent vector by noticing that $|\gamma_L'(t)| \equiv 1$ for $0 \le t \le \ell$ due to the arclength parametrization. Following the proof in Proposition 3.1, we have the error estimate

$$|\boldsymbol{\nu}^L - \widetilde{\boldsymbol{\nu}}^L| = \mathcal{O}(h).$$

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

$$|\boldsymbol{\nu}^R - \widetilde{\boldsymbol{\nu}}^R| = \mathcal{O}(h).$$

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

$$-\boldsymbol{\nu} = a^L \boldsymbol{\nu}^L + a^R \boldsymbol{\nu}^R.$$

By comparing equation (C.1) and noticing that $|\boldsymbol{\nu} - \widetilde{\boldsymbol{\nu}}| = \mathcal{O}(h)$, we have the estimation for coefficients,

$$|a^L - \widetilde{a}^L| = \mathcal{O}(h) \quad \text{and} \quad |a^R - \widetilde{a}^R| = \mathcal{O}(h).$$

Next, we estimate the error between the analytic directional derivative $\frac{\partial u}{\partial \boldsymbol{\nu}^L}$ and the numerical estimation

$$\frac{\Delta u}{\Delta \widetilde{\boldsymbol{\nu}}^L} := \frac{u(x^L) - u(x^B)}{|x^L - x^B|}.$$

Let $\vec{z} = (z_1, \ldots, z_d)$ denote the d-dimensional ($d = 2$) geodesic normal coordinate of x_L defined by an exponential map $\exp_{x^B} : T_{x^B}M \to M$; then $\vec{z}$ satisfies

$$\vec{z} = \ell \boldsymbol{\nu}^L = \ell \gamma_L'(0) \quad \text{and} \quad \exp_{x^B} \vec{0} = x^B,\ \exp_{x^B} \vec{z} = x^L,$$

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

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

which is equivalent to

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

by noticing that $\vec{z} = \ell \boldsymbol{\nu}^L$ 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 denotes 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 $\frac{\partial u}{\partial \boldsymbol{\nu}^L}$ and the numerical $\frac{\Delta u}{\Delta \widetilde{\boldsymbol{\nu}}^L}$:

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

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

$$\frac{u(x^R) - u(x^B)}{|x^R - x^B|} = \frac{\partial u}{\partial \boldsymbol{\nu}^R}(x^B) + \mathcal{O}(\ell),$$

where we have introduced an arclength 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:

$$\begin{aligned}
&\left| \frac{\partial u}{\partial \boldsymbol{\nu}}(x^B) - \frac{\Delta u}{\Delta \widetilde{\boldsymbol{\nu}}}(x^B) \right| (x^B) + \widetilde{a}^L \frac{u(x^L) - u(x^B)}{|x^L - x^B|} + \widetilde{a}^R \frac{u(x^R) - u(x^B)}{|x^R - x^B|} \Bigg| \\
&\quad \le \left| a^L \frac{\partial u}{\partial \boldsymbol{\nu}^L}(x^B) - \widetilde{a}^L \frac{u(x^L) - u(x^B)}{|x^L - x^B|} \right| + \left| a^R \frac{\partial u}{\partial \boldsymbol{\nu}^R}(x^B) - \widetilde{a}^R \frac{u(x^R) - u(x^B)}{|x^R - x^B|} \right| \\
&\quad \le |a^L - \widetilde{a}^L| \left| \frac{\partial u}{\partial \boldsymbol{\nu}^L}(x^B) \right| + |\widetilde{a}^L| \left| \frac{\partial u}{\partial \boldsymbol{\nu}^L}(x^B) - \frac{u(x^L) - u(x^B)}{|x^L - x^B|} \right| \\
&\qquad + |a^R - \widetilde{a}^R| \left| \frac{\partial u}{\partial \boldsymbol{\nu}^R}(x^B) \right| + |\widetilde{a}^R| \left| \frac{\partial u}{\partial \boldsymbol{\nu}^R}(x^B) - \frac{u(x^R) - u(x^B)}{|x^R - x^B|} \right| = \mathcal{O}(h). \quad \square
\end{aligned}$$

Acknowledgment. The research of JH was partially supported under National Science Foundation 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 a sample code of FELICITY FEM, and Ryan Vaughn and Tyrus Berry for the helpful discussion on various aspects of differential geometry.

Bibliography

[1] Ahlberg, J. H.; Nilson, E. N. Convergence properties of the spline fit. *J. Soc. Indust. Appl. Math.* **11** (1963), no. 1, 95–104.

[2] Aslam, T.; Luo, S.; Zhao, H. A static PDE approach for multidimensional extrapolation using fast sweeping methods. *SIAM J. Sci. Comput.* **36** (2014), no. 6, A2907–A2928.

[3] Bayona, V.; Flyer, N.; Fornberg, B.; Barnett, G. A. On the role of polynomials in RBF-FD approximations: II. Numerical solution of elliptic PDEs. *J. Comput. Phys.* **332** (2017), 257–273. doi:10.1016/j.jcp.2016.12.008

[4] Berry, T.; Harlim, J. Variable bandwidth diffusion kernels. *Appl. Comput. Harmon. Anal.* **40** (2016), no. 1, 68–96. doi:10.1016/j.acha.2015.01.001

[5] Berry, T.; Harlim, J. Iterated diffusion maps for feature identification. *Appl. Comput. Harmon. Anal.* **45** (2018), no. 1, 84–119. doi:10.1016/j.acha.2016.08.005

[6] Berry, T.; Sauer, T. Local kernels and the geometric structure of data. *Appl. Comput. Harmon. Anal.* **40** (2016), no. 3, 439–469. doi:10.1016/j.acha.2015.03.002

[7] Berry, T.; Sauer, T. Density estimation on manifolds with boundary. *Comput. Statist. Data Anal.* **107** (2017), 1–17.

[8] Berry, T.; Sauer, T. Consistent manifold representation for topological data analysis. *Foundations of Data Science* **1** (2019), 1.

[9] Bertalmıo, M.; Cheng, L.-T.; Osher, S.; Sapiro, G. Variational problems and partial differential equations on implicit surfaces. *J. Comput. Phys.* **174** (2001), no. 2, 759–780. doi:10.1006/jcph.2001.6937

[10] Boissonnat, J.-D.; Ghosh, A. Triangulating smooth submanifolds with light scaffolding. *Math. Comput. Sci.* **4** (2010), no. 4, 431–461. doi:10.1007/s11786-011-0066-5

[11] Bonito, A.; Cascón, J. M.; Mekchay, K.; Morin, P.; Nochetto, R. H. High-order AFEM for the Laplace-Beltrami operator: convergence rates. *Found. Comput. Math.* **16** (2016), no. 6, 1473–1539. doi:10.1007/s10208-016-9335-7

[12] Calder, J.; Trillos, N. G. Improved spectral convergence rates for graph Laplacians on epsilon-graphs and k-NN graphs. Preprint, 2019. arXiv:1910.13476 [math.PR]

[13] Calder, J.; Trillos, N. G.; Lewicka, M. Lipschitz regularity of graph Laplacians on random data clouds. Preprint, 2020. arXiv:2007.06679 [math.AP]

[14] Camacho, F.; Demlow, A. L_2 and pointwise a posteriori error estimates for FEM for elliptic PDEs on surfaces. *IMA J. Numer. Anal.* **35** (2015), no. 3, 1199–1227. doi:10.1093/imanum/dru036

[15] Coifman, R. R.; Lafon, S. Diffusion maps. *Appl. Comput. Harmon. Anal.* **21** (2006), no. 1, 5–30. doi:10.1016/j.acha.2006.04.006

[16] Coifman, R. R.; Shkolnisky, Y.; Sigworth, F. J.; Singer, A. Graph Laplacian tomography from unknown random projections. *IEEE Trans. Image Process.* **17** (2008), no. 10, 1891–1899. doi:10.1109/TIP.2008.2002305

[17] Crane, K. Keenan's 3d model repository. Available at: `http://www.cs.cmu.edu/~kmcrane/Projects/ModelRepository`.

[18] Dunson, D.; Wu, H.-T.; Wu, N. Spectral convergence of graph Laplacian and Heat kernel reconstruction in L^∞ from random samples. Preprint, 2019. arXiv:1912.05680 [math.ST]

[19] Dziuk, G.; Elliott, C. M. Finite element methods for surface PDEs. *Acta Numer.* **22** (2013), 289–396. doi:10.1017/S0962492913000056

[20] Elliott, C. M.; Stinner, B. Modeling and computation of two phase geometric biomembranes using surface finite elements. *J. Comput. Phys.* **229** (2010), no. 18, 6585–6612. doi:10.1016/j.jcp.2010.05.014

[21] Feynman, R. P.; Leighton, R. B.; Sands, M. *The Feynman Lectures on Physics, Vol. I: The New Millennium Edition: Mainly Mechanics, Radiation, and Heat*. Basic Books, 2011.

[22] Fornberg, B.; Driscoll, T. A.; Wright, G.; Charles, R. Observations on the behavior of radial basis function approximations near boundaries. *Comput. Math. Appl.* **43** (2002), no. 3-5, 473–490. doi:10.1016/S0898-1221(01)00299-1

[23] Fuselier, E. J.; Wright, G. B. A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. *J. Sci. Comput.* **56** (2013), no. 3, 535–565. doi:10.1007/s10915-013-9688-x

[24] García Trillos, N.; Gerlach, M.; Hein, M.; Slepčev, D. Error estimates for spectral convergence of the graph Laplacian on random geometric graphs toward the Laplace-Beltrami operator. *Found. Comput. Math.* **20** (2020), no. 4, 827–887. doi:10.1007/s10208-019-09436-w

[25] Gilani, F.; Harlim, J. Approximating solutions of linear elliptic PDE's on a smooth manifold using local kernel. *J. Comput. Phys.* **395** (2019), 563–582. doi:10.1016/j.jcp.2019.06.034

[26] Gilbarg, D.; Trudinger, N. S. *Elliptic partial differential equations of second order*. Springer, Berlin–New York, 2015.

[27] Han, Q.; Lin, F. *Elliptic partial differential equations.* Second edition. Courant Lecture Notes in Mathematics, 1. Courant Institute of Mathematical Sciences, New York; American Mathematical Society, Providence, R.I., 2011.

[28] Harlim, J. *Data-driven computational methods: Parameter and operator estimations*. Cambridge University Press, Cambridge, 2018. doi:10.1017/9781108562461

[29] Harlim, J.; Sanz-Alonso, D.; Yang, R. Kernel methods for Bayesian elliptic inverse problems on manifolds. *SIAM/ASA J. Uncertain. Quantif.* **8** (2020), no. 4, 1414–1445. doi:10.1137/19M1295222

[30] Larsson, S.; Thomée, V. *Partial differential equations with numerical methods*. Texts in Applied Mathematics, 45. Springer, Berlin, 2009.

[31] Lee, J. M. *Introduction to Smooth Manifolds*. Second edition. Graduate Texts in Mathematics, 218. Springer, New York, 2013.
[32] Lee, J. M. *Introduction to Riemannian manifolds*. Graduate Texts in Mathematics, 176. Springer, Cham, 2018.
[33] LeVeque, R. J. *Finite difference methods for ordinary and partial differential equations. Steady-state and time-dependent problems*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, 2007.
[34] Li, Z.; Shi, Z. A convergent point integral method for isotropic elliptic equations on a point cloud. *Multiscale Model. Simul.* **14** (2016), no. 2, 874–905. doi:10.1137/15M102592X
[35] Li, Z.; Shi, Z.; Sun, J. Point integral method for solving Poisson-type equations on manifolds from point clouds with convergence guarantees. *Commun. Comput. Phys.* **22** (2017), no. 1, 228–258. doi:10.4208/cicp.111015.250716a
[36] Loftsgaarden, D. O.; Quesenberry, C. P. A nonparametric estimate of a multivariate density function. *Ann. Math. Statist.* **36** (1965), no. 3, 1049–1051. doi:10.1214/aoms/1177700079
[37] Lu, J. Graph approximations to the Laplacian spectra. *Journal of Topology and Analysis* (2020), 1–35. doi:10.1142/S1793525320500442
[38] Macdonald, C. B.; Ruuth, S. J. The implicit closest point method for the numerical solution of partial differential equations on surfaces. *SIAM J. Sci. Comput.* **31** (2010), no. 6, 4330–4350. doi:10.1137/080740003
[39] McLaughlin, D.; Townley, L. R. A reassessment of the groundwater inverse problem. *Water Resources Research* **32** (1996), no. 5, 1131–1161. doi:10.1029/96WR00160
[40] Mémoli, F.; Sapiro, G.; Thompson, P. Implicit brain imaging. *NeuroImage* **23** (2004), S179–S188. doi:10.1016/j.neuroimage.2004.07.072
[41] Mörters, P.; Peres, Y. *Brownian motion*. Cambridge Series in Statistical and Probabilistic Mathematics, 30. Cambridge University Press, Cambridge, 2010. doi:10.1017/CBO9780511750489
[42] Nardi, G. Schauder estimate for solutions of Poisson's equation with Neumann boundary condition. *Enseign. Math.* **60** (2015), no. 3-4, 421–435. doi:10.4171/LEM/60-3/4-9
[43] Petras, A.; Ling, L.; Ruuth, S. J. An RBF-FD closest point method for solving PDEs on surfaces. *J. Comput. Phys.* **370** (2018), 43–57. doi:10.1016/j.jcp.2018.05.022
[44] Piret, C. The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces. *J. Comput. Phys.* **231** (2012), no. 14, 4662–4675. doi:10.1016/j.jcp.2012.03.007
[45] Rauter, M.; Tuković, Ž. A finite area scheme for shallow granular flows on three-dimensional surfaces. *Comput. & Fluids* **166** (2018), 184–199. doi:10.1016/j.compfluid.2018.02.017
[46] Ruuth, S. J.; Merriman, B. A simple embedding method for solving partial differential equations on surfaces. *J. Comput. Phys.* **227** (2008), no. 3, 1943–1961. doi:10.1016/j.jcp.2007.10.009
[47] Shi, Z. Enforce the Dirichlet boundary condition by volume constraint in point integral method. *Commun. Math. Sci.* **15** (2017), no. 6, 1743–1769. doi:10.4310/CMS.2017.v15.n6.a12
[48] Singer, A. From graph to manifold Laplacian: The convergence rate. *Appl. Comp. Harmonic Anal.* **21** (2006), no. 1, 128–134. doi:10.1016/j.acha.2006.03.004
[49] Thiede, E. H.; Giannakis, D.; Dinner, A. R.; Weare, J. Galerkin approximation of dynamical quantities using trajectory data. *The Journal of Chemical Physics* **150** (2019), no. 24, 244111. doi:10.1063/1.5063730
[50] Varah, J. M. A lower bound for the smallest singular value of a matrix. *Linear Algebra Appl.* **11** (1975), no. 1, 3–5. doi:10.1016/0024-3795(75)90112-3
[51] Vaughn, R.; Berry, T.; Antil, H. Diffusion maps for embedded manifolds with boundary with applications to PDEs. Preprint, 2019. arXiv:1912.01391 [math.NA]
[52] Virga, E. G. *Variational theories for liquid crystals*. CRC Press, Boca Raton, Fla., 2018. doi:10.1201/9780203734421
[53] Wachspress, E. L. *Iterative solution of elliptic systems, and applications to the neutron diffusion equations of reactor physics*. Prentice-Hall, Englewood Cliffs, N.J., 1966.

[54] Walker, S. W. FELICITY: A MATLAB/C++ toolbox for developing finite element methods and simulation modeling. *SIAM J. Sci. Comput.* **40** (2018), no. 2, C234–C257. doi:10.1137/17M1128745

SHIXIAO JIANG
Institute of Mathematical Sciences
ShanghaiTech University
Shanghai 201210
CHINA
E-mail: jiangshx@shanghaitech.edu.cn

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
E-mail: jharlim@psu.edu

Received June 2020.
Revised March 2021.