2018

Computing Eigenmodes of Elliptic Operators on Manifolds Using Radial Basis Functions

Vladimir Delengov

Computing Eigenmodes of Elliptic Operators on Manifolds Using Radial Basis Functions

Vladimir Delengov

Institute of Mathematical Sciences

Claremont Graduate University

APPROVAL OF THE REVIEW COMMITTEE

This dissertation has been duly read, reviewed, and critiqued by the Committee listed below, which hereby approves the manuscript of Vladimir Delengov as fulfilling the scope and quality requirements for meriting the degree of Doctor of Philosophy.

Chiu-Yen Kao
Department of Mathematical Sciences
Claremont McKenna College
Professor

Marina Chugunova
Institute of Mathematical Sciences
Claremont Graduate University
Associate Professor

Ali Nadim
Institute of Mathematical Sciences
Claremont Graduate University
Professor

Abstract

Computing Eigenmodes of Elliptic Operators on Manifolds

Using Radial Basis Functions

by

Vladimir Delengov

Claremont Graduate University, 2018

In this work, numerical approaches based on meshless methods are proposed to obtain eigenmodes of elliptic operators on manifolds, and their performance is compared against existing alternative methods. Radial Basis Function (RBF)-based methods allow one to obtain interpolation and differentiation matrices easily by using scattered data points. We derive expressions for such matrices for the Laplace-Beltrami operator via so-called Reilly's formulas and use them to solve the respective eigenvalue problem. Numerical studies of proposed methods are performed in order to demonstrate convergence on simple examples of one-dimensional curves and two-dimensional surfaces. Prospective extensions of the methods include application to problems with boundary conditions and incorporating a multi-layer approach in order to improve accuracy. The latter is justified by an asymptotic expansion of eigenvalues of Laplace operator on a thin ring and a thin shell.

Acknowledgements

This work is dedicated to my beloved fiancée Sara, who always supported me unconditionally and believed in me no matter what.

Table of Contents

Chapter 1

Introduction to Numerical Approaches to PDEs on Manifolds

Solving partial differential equations (PDEs) on surfaces or more general manifolds has far-reaching applications. Such problems have recently received a lot of attention due to potential applications in: (i) computer vision on surface reconstruction and geodesic computation, (ii) biology [7, 61, 62, 66] and chemistry [13, 27, 48], (iii) physical modeling of elastic materials, (iv) wave propagation in excitable media [24, 27, 48, 68], and (v) image processing on surfaces (e.g. image segmentation, image restoration, image inpainting, and texture mapping [65]).

One of the most crucial aspects of solving time-dependent partial differential equations is numerical stability. In order to ensure convergence and accuracy of the solution of linear PDEs of the second order involving dissipation on manifolds, one can use information about eigenmodes of Laplace-Beltrami operator.

1.1 Existing Numerical Approaches

Generally all currently available numerical methods for solving PDEs on manifolds can be classified in three families: (i) parametrization, (ii) meshing, and (iii) embedding. Each class of methods has its own strong suits along with corresponding disadvantages.

Parametrization, or *intrinsic* techniques usually rely on using local coordinates in the form of surface-dependent mesh [34, 40, 67]. Such an approach, however, bears risk of complications associated with the local parametrizations (atlas) and corresponding singularities arising in Cartesian coordinates. Indeed, the parameterization introduces coordinate systems which may generate local distortion or require patches of several local coordinate systems [41, 42].

The meshing approach requires the introduction of a (triangular) mesh of the manifold which may be difficult to generate [15, 16]. Besides dealing with singularities, meshing techniques are usually limited to algebraic order of accuracy. Among advantages of such methods are low dimensionality of discretized problems and simplicity of implementation for surfaces with known global parametrization. These methods are generally very efficient once a parametrization or a mesh is provided.

The core idea of an embedding approach is to extend the solution to a neighborhood of the surface and to solve PDE in ambient space which is equal to the original PDE on the surface. This approach avoids the complication of the construction of a parametrization or a mesh. Examples of embedding approaches are level set methods [3, 26] and closest point methods [9, 43, 45, 46, 50].

Recently several new meshless methods were proposed, including grid based particle methods [36–38], point clouds methods [33, 39, 44], and radial basis function (RBF) approaches [11, 18, 20, 54]. The meshless methods based on radial basis functions [6] to solve PDEs on stationary and moving surfaces attract more and more attention due to their simplicity in numerical implementation, flexibility in points cloud datasets, and possibility of spectral accuracy.

In this work, we focus on the eigenmodes of Laplace-Beltrami operator on manifolds due to the fact that corresponding eigenvalues and eigenfunctions are of great importantance for theoretical as well as applied disciplines. Examples of theoretical applications of aforementioned eigenmodes include shape analysis, shape optimization, and differential geometry, as well as the related applied disciplines: spectral analysis, the Hodge decomposition in integral formulation of certain engineering problems [12], computational electromagnetism [28, 51], and some shape analysis applications in computer vision rely on normalized eigenfunctions of Laplace-Beltrami operator to classify and relate geometrical objects [57, 60]. One of the features of eigenmodes of Laplace-Beltrami operator that makes them so useful for shape analysis and other theoretical/abstract disciplines is their invariance under *isometries* (bijective distance-preserving transformations) [56, 57]. This property comes in particularly handy for areas such as computer vision, animation, modeling organs or body parts, as well as any other application dealing with dynamic bendable geometric objects that do not stretch or rapture [58].

Since Laplace-Beltrami operator is self-adjoint, its spectrum lies entirely on the negative half of real axis on the complex plane. Recently there was a spike of interest in numerical techniques for optimization of eigenvalues of Laplace-Beltrami operator via varying underlying domain manifold of fixed genus [32]. Among theoretical applications of Laplace-Beltrami eigenmodes are shape optimization [1], higher-order differential operators on surfaces [30], time-dependent differential equations on manifolds [19, 20, 35], and statistical shape analysis [58].

1.2 Outline

The aim of this thesis is to develop new RBF approaches to find eigenmodes of Laplace-Beltrami operator and to study the numerical convergence of various related methods. We demonstrate the simplicity, efficiency, and robustness of the proposed method based on [55] via many numerical examples. Furthermore, we provide an asymptotic study of Laplace eigenvalues on a thin ring and a thin shell and show that the Laplace eigenvalues with Neumann boundary conditions indeed converge to the eigenvalues of Laplace-Beltrami operator on a circle and a sphere. Several new ideas are proposed to extend the existing RBF methods. The proposed methods are aimed at solving problems with boundary conditions, as well as improving accuracy by using multilayers.

The following chapter 2 poses the problem of interest in an analytical setting and lays down theoretical foundations which will be used throughout the rest of the document. The asymptotic results on the thin ring and thin shell demonstrate that the eigenvalues of the Laplace operator with Neumann boundary conditions indeed converge to the corresponding eigenvalues of Laplace-Beltrami operator on a circle and a sphere. Next, chapter 3 contains an overview and a brief summary of the relevant numerical methods, as well as highlights of their strengths and weaknesses. In chapter 4, new approaches are discussed in detail including a new RBF approach based on Reilly's formulas and general eigenvalue solvers to compute eigenvalues accurately. In chapter 5 we include a summary of conducted numerical studies comparing the performance of presented methods, which are followed by conclusions and future work, including a multilayer approach to improve the accuracy, and solving PDEs on general manifolds with boundary conditions in chapter 6.

Chapter 2

Introduction to Laplace-Beltrami Operator and Its Eigenvalues

2.1 Definitions

In order to avoid confusion with notation, we first provide the formal definitions of a manifold from [29].

Definition 1. A *manifold* $\mathcal{M}$ of dimension m is a connected paracompact Hausdorff space for which every point has a neighborhood U that is homeomorphic to an open subset Ω of $\mathbb{R}^m$. Such a homeomorphism $\boldsymbol{t} : U \to \Omega$ is called a *(coordinate) chart.*

Throughout this section we assume that a smooth compact manifold $\mathcal{M}$ of dimension m is embedded into Euclidean space $\mathbb{E} = \mathbb{R}^n$. The coordinate system in $\mathbb{R}^n$ is the set of Cartesian coordinates $\boldsymbol{x} = [x^1, x^2, \ldots, x^n]^\intercal$, whereas the *local coordinate chart* in $\mathcal{M}$ are given by diffeomorphism $\boldsymbol{t} = [t^1, \ldots, t^m]^\intercal$. More specifically, given a point $p \in U \subset \mathcal{M}$ the components $t^1, \ldots, t^m$ of $\boldsymbol{t}(p)$ are called *local coordinates* of p.

Laplace-Beltrami operator is a generalization of Laplace operator in Euclidean space $\mathbb{E}$ to the case of a Riemannian manifold $(\mathcal{M}, g)$, where g is the *metric tensor*. Laplace-Beltrami operator $\Delta_{\mathcal{M}}$ preserves key properties of Euclidean Laplacian Δ such as linearity, self-adjointness, and uniform ellipticity, and is similarly defined as the divergence of the gradient.

Assume that $\mathcal{M}$ is (Riemannian) differentiable manifold embedded into Euclidean space $\mathbb{E}$ equipped with the metric $g_{ij} = J^{\intercal}J$, where J is the Jacobian matrix mapping from the embedding space coordinates $\boldsymbol{x}$ to the local coordinates $\boldsymbol{t}$ of the manifold $\mathcal{M}$.

Below we provide definitions and examples of differential operators on manifolds and other related concepts. In particular, we present relevant examples of tangential differential operators, present formulas, theorems, and asymptotic analysis concerning differential operators that form theoretical foundation for the newly proposed techniques in chapter 4.

2.1.1 Tangent Spaces in the Context of Differential Geometry

For a manifold $\mathcal{M}$ the *tangent space* $T_p\mathcal{M}$ at a point $p \in \mathcal{M}$ consists of vectors tangent to all possible curves in $\mathcal{M}$ passing through p. The basis vectors of $T_p\mathcal{M}$ are denoted as $\{\partial_i\}$ or $\{{}^{\partial}/_{\partial t^i}\}$.

Note. In this context the notation ${}^{\partial}/_{\partial_i}$ as well as its short form ∂_i are used to denote *vectors* in the tangent space $T_p\mathcal{M}$ because there is one-to-one correspondence between linear functionals acting on functions defined on $\mathcal{M}$ and the basis vectors of the corresponding dual space $T_p\mathcal{M}$ established by the Riesz representation theorem [59]. To avoid confusion in notation, we refer readers to the textbook [29].

Then, any function $f : \mathcal{M} \to \mathbb{R}$ can be differentiated along $\{\partial_i\}$, thus the convention is to use the partial derivative symbol as notation for basis vector.

Following convention of the textbook [29] we use lower indexing for the tangent vectors which are *covariant*, i.e. for those whose coordinates are changed by the same transformation, as the original local basis on $\mathcal{M}$.

Let us denote $T_p^*\mathcal{M}$ the space of all linear functions $w : T_p\mathcal{M} \to \mathbb{R}$ (also known as 1-forms) defined on the linear space of tangent vectors. $T_p^*\mathcal{M}$ is called a *cotangent space*, and its basis is denoted as $\{dt^i\}$ with the upper index, since these linear functions (1-forms) are *contravariant*, i.e. if we transform the local basis on $\mathcal{M}$ by a matrix A, the coordinates of vectors from $T_p\mathcal{M}$ will be changed by the inverse matrix A^{-1}. Note that $T_p\mathcal{M}$ and $T_p^*\mathcal{M}$ are dual spaces, therefore $\{{}^{\partial}/_{\partial t^i}\}$ and $\{dt^i\}$ are the dual bases.

2.1.2 Jacobian Matrix and Metric Tensor

In this work we focus on meshfree numerical methods that utilize Euclidean distance in the embedding space $\mathbb{E}$, thus whenever we talk about metric tensor g of $\mathcal{M}$ we imply the *induced metric* defined below via Jacobian[1].

Definition 2 (*Jacobian*)**.** Assume that an m-dimensional manifold $\mathcal{M}$ is given in terms of Cartesian coordinates $\boldsymbol{x} = [x^1, \dots, x^n]^\intercal$ of a n-dimensional embedding space via parametrization $\boldsymbol{x}(\boldsymbol{t}) : \mathbb{R}^m \to \mathbb{R}^n$ where $\boldsymbol{t} = [t^1, \dots, t^m]^\intercal$. Then *Jacobian Matrix* (or *Jacobian*, for short) is defined as

$$J := \left[\frac{\partial \boldsymbol{x}}{\partial t^1} \dots \frac{\partial \boldsymbol{x}}{\partial t^m}\right] = \begin{bmatrix} \frac{\partial x^1}{\partial t^1} & \dots & \frac{\partial x^1}{\partial t^m} \\ \vdots & \ddots & \vdots \\ \frac{\partial x^n}{\partial t^1} & \dots & \frac{\partial x^n}{\partial t^m} \end{bmatrix}. \tag{2.1}$$

[1] In general, for a given manifold the choice of metric is not unique as metric tensor g is just a smooth positive-definite quadratic form defined on tangent bundle $T_p\mathcal{M}$ in terms of $\partial_i\partial_j$.

The *metric tensor* $g_{ij} = \partial_i \partial_j$ is a twice covariant tensor of order $(0, 2)$, i.e. the bilinear functional defined on tangent vectors from $T_p\mathcal{M}$. It maps two elements from the tangent space into a real number, so $g_{ij} : T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}$. For example, for all $\vec{\boldsymbol{u}}, \vec{\boldsymbol{v}} \in T_p\mathcal{M}$ represented in the basis $\{\partial_i\}$ as $\vec{\boldsymbol{u}} = u^i \partial_i$ and $\vec{\boldsymbol{v}} = v^j \partial_j$, we define inner product as

$$\langle \vec{\boldsymbol{u}}, \vec{\boldsymbol{v}} \rangle := g_{ij} u^i v^j = \left(u^i \partial_i\right)\left(v^j \partial_j\right).$$

More formally, we can define metric tensor in terms of Jacobian in the following manner.

Definition 3 (*Metric Tensor*)**.** Given a parametrization of (local) coordinates (or an embedding of a manifold into Euclidean surface) $\boldsymbol{x} = \vec{\boldsymbol{x}}(\boldsymbol{t})$ we define *metric tensor* for the metric *induced* on $\mathcal{M}$ via the Jacobian:

$$g_{ij} = J^\intercal J. \tag{2.2}$$

This definition allows us to express *inner product* of tangent vectors $\vec{\boldsymbol{u}}, \vec{\boldsymbol{v}} \in T_p\mathcal{M}$ in terms of coordinates of embedding space $\mathbb{E}$.

The inverse of g_{ij} called *inverse* or *dual metric* is denoted as g^{ij}. It is a *contravariant* bilinear form used to perform inner product operation for the fields of linear functional (covector fields) [29].

2.1.3 Tangential Differential Operators

Let us consider first-order differential operators defined on a tangent space of a manifold, in particular gradient and divergence.

Definition 4. The *tangential (or surface) gradient* $\nabla_\mathcal{M} f$ of a function $f : \mathcal{M} \to \mathbb{R}$ at point p is a covariant vector (covector) in the tangent space $T_p\mathcal{M}$, i.e. a tensor of the order $(1, 0)$.

Here once again we rely on duality between the linear operators and vectors in $T_p\mathcal{M}$. Applied to a 1-form from the cotangent space $T_p^*\mathcal{M}$, it produces a real number. Applied to a vector from $T_p\mathcal{M}$ (by the means of inner product) the gradient will produce the amount that function f increases following this vector.

Let us now derive the explicit expression for tangential gradient. Given a (regular) gradient of a function f in the form $\nabla_{\mathcal{M}} f = a^i \partial_i \in T_p\mathcal{M}$ and some vector $\vec{\boldsymbol{v}} = v^j \partial_j \in T_p\mathcal{M}$, we consider their inner product

$$\left\langle \nabla_{\mathcal{M}} f, \vec{\boldsymbol{v}} \right\rangle = g_{ij}\, v^j a^i.$$

We require this inner product to be equal to the directional derivative of f along the vector $\vec{\boldsymbol{v}}$, i.e.

$$\left\langle \nabla_{\mathcal{M}} f, \vec{\boldsymbol{v}} \right\rangle = \sum_{i=1}^{m} v^j \frac{\partial f}{\partial t^j} = v^j \partial_j f$$

Combining the last two expressions we get

$$g_{ij}\, a^i = \partial_j f = \frac{\partial f}{\partial t^j} \iff a^i = \left(g_{ij}\right)^{-1} \frac{\partial f}{\partial t^j} = g^{ij} \frac{\partial f}{\partial t^j} = g^{ij} \partial_j f,$$

where g^{ij} is the *inverse metric* for g_{ij}. In this way, we write the explicit formula for gradient:

$$\nabla_{\mathcal{M}} f = g^{ij} \frac{\partial f}{\partial t^j}\, \partial_i = g^{ij} \frac{\partial f}{\partial t^j} \frac{\partial}{\partial t^i}$$

Note. The differential $\mathrm{d}f = \left\langle \nabla_{\mathcal{M}} f, \vec{\boldsymbol{v}} \right\rangle$ of function f in the direction $\vec{\boldsymbol{v}}$ is actually a 1-form, so it belongs to cotangent space: $\mathrm{d}f \in T_p^*\mathcal{M}$.

Recall that a *vector field* $\vec{\boldsymbol{F}} : \mathcal{M} \to T_p\mathcal{M}$ is a smooth differentiable vector function, which maps a point p to the vector in $T_p\mathcal{M}$. The *divergence* of a vector field $\vec{\boldsymbol{F}}$ defined on a manifold $\mathcal{M}$ maps $\vec{\boldsymbol{F}}$ to the scalar field on $\mathcal{M}$. It is defined using so-called *volume form.*

The divergence operator computes how does a unit volume changes under the action of $\vec{\boldsymbol{F}}$. Let us make use of the following formula:

$$\left\langle \vec{\boldsymbol{F}}, \nabla_{\mathcal{M}} f \right\rangle = \left\langle -\operatorname{div}_{\mathcal{M}} \vec{\boldsymbol{F}}, f \right\rangle$$

which holds for any f with compact support [29].

Let us define the divergence operator formally. For a point $p \in \mathcal{M}$ and a vector field $\vec{\boldsymbol{F}} : \mathcal{M} \to T_p\mathcal{M}$,

$$\vec{\boldsymbol{F}} = \left[F_1\ F_2 \cdots F_n\right]^{\intercal},$$

the divergence is defined as

$$\lim_{\Delta\vec{\boldsymbol{F}}\to 0} \frac{1}{\Delta\vec{\boldsymbol{F}}} \oint_U \vec{\boldsymbol{F}} \cdot \vec{\boldsymbol{n}}\, dS,$$

where U is a small neighborhood around $\vec{\boldsymbol{F}}(p)$, and $\vec{\boldsymbol{n}}$ is the unit normal vector pointing away from p. Observe the following property of the divergence:

$$\int_U \operatorname{div}_{\mathcal{M}} \vec{\boldsymbol{F}}\, dV = \int_{\partial U} \vec{\boldsymbol{F}} \cdot \vec{\boldsymbol{n}}\, dS$$

where dV is the volume increment on U. Consider now an open neighborhood $U \in T_p\mathcal{M}$ of p and a smooth function $h : U \to \mathbb{R}$ defined on this neighborhood. Applying divergence theorem to $(h \circ \vec{\boldsymbol{F}})$ in U, we get

$$\int_U \operatorname{div}_{\mathcal{M}}(h \circ \vec{\boldsymbol{F}})\, dV = 0,$$

so that

$$\int_U h \operatorname{div}_{\mathcal{M}} \vec{\boldsymbol{F}}\, dV = -\int_U \vec{\boldsymbol{F}} \cdot \nabla_{\mathcal{M}} h\, dV = -\int_U F^i \frac{\partial h}{\partial t^i} \sqrt{|g|}\, dt^1 \wedge \cdots \wedge dt^m$$

with $|g| = \det(g_{ij})$. If $\phi : U \to \mathbb{R}^n$ is an isomorphism, then we can map everything into Euclidean Space: $\widetilde{\vec{\boldsymbol{F}}} = \vec{\boldsymbol{F}} \circ \phi^{-1}$, $\widetilde{h} = h \circ \phi^{-1}$. Then by Green's Formula

$$\int_U F^i \frac{\partial h}{\partial t^i} \sqrt{|g|}\, dt^1 \wedge \cdots \wedge dt^m = \int_{\phi(U)} \widetilde{\vec{\boldsymbol{F}}}^i \frac{\partial \tilde{h}}{\partial t^i} \sqrt{|g|}\, dt^1 \cdots dt^m =$$

$$= -\int_{\phi(U)} \tilde{h} \frac{\partial\left(\sqrt{|g|}\widetilde{\vec{\boldsymbol{F}}}^i\right)}{\partial t^i}\, dt^1 \cdots dt^m = -\int_U \frac{1}{\sqrt{|g|}} \sum_i \frac{\partial(\sqrt{|g|}F^i)}{\partial t^i} h\, dV.$$

In this way, we get the following definition.

Definition 5 (*Tangential divergence*)**.** The *tangential* (or *surface*) *divergence* is the linear differential operator $\mathrm{div}_{\mathcal{M}}$ acting on a vector field $\vec{\boldsymbol{F}} : \mathcal{M} \to T_p\mathcal{M}$ which is given by the expression

$$\mathrm{div}_{\mathcal{M}} \vec{\boldsymbol{F}} = \frac{1}{\sqrt{|g|}} \sum_i \frac{\partial(\sqrt{|g|}F^i)}{\partial t^i}.$$

For the sake of brevity we denote $\nabla_{\mathcal{M}} \cdot \vec{\boldsymbol{F}} \equiv \mathrm{div}_{\mathcal{M}} \vec{\boldsymbol{F}}$ and rewrite the equation above using Einstein notation

$$\nabla_{\mathcal{M}} \cdot \vec{\boldsymbol{F}} = |g|^{-\frac{1}{2}}\, \partial_i \big(\, |g|^{\frac{1}{2}}\, F^i \big).$$

Let us now combine definitions of divergence and gradient from the previous subsections to define Laplace-Beltrami operator on Euclidean Space.

Definition 6 (*Laplace-Beltrami operator*)**.** The *Laplace-Beltrami* operator on the manifold $\mathcal{M}$ is the linear differential operator of second order $\Delta_{\mathcal{M}} : \mathcal{C}^{\infty}(\mathcal{M}) \to \mathcal{C}^{\infty}(\mathcal{M})$ defined as the divergence of the gradient

$$\Delta_{\mathcal{M}} \boldsymbol{f} := \mathrm{div}_{\mathcal{M}} (\nabla_{\mathcal{M}} \boldsymbol{f}) = \nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} \boldsymbol{f}.$$

The Laplace-Beltrami operator can be written explicitly in terms of metric tensor [29]:

$$\Delta_{\mathcal{M}} \boldsymbol{f} = \frac{1}{\sqrt{|g|}} \frac{\partial}{\partial t^i} \left(\sqrt{|g|} g^{ij} \frac{\partial f}{\partial t^j} \right) = |g|^{-\frac{1}{2}}\, \partial_i \big(\, |g|^{\frac{1}{2}}\, g^{ij} \partial_j \boldsymbol{f} \big), \tag{2.3}$$

where $\nabla_{\mathcal{M}} \boldsymbol{f} = g^{ij}\, \partial_j f$ and $\nabla_{\mathcal{M}} \cdot \vec{\boldsymbol{F}} = |g|^{-\frac{1}{2}}\, \partial_i \big(\, |g|^{\frac{1}{2}}\, F^i \big)$.

2.1.4 Examples

Below we provide several relevant examples intended to demonstrate practical implementation of formulas in subsection 2.1.3 for several test cases considered later in chapter 5. Furthermore, later in section 2.3, these examples will be used to validate alternative ways of computing Laplace-Beltrami operator that involve derivatives and curvatures, namely

Reilly's formulas (2.12) and (2.13).

Example 2.1 (*Parametric planar curve: circle*)**.**

Consider a circle of radius R parametrized in terms of the angle θ

$$\begin{cases} x(\theta) = R\cos\theta, \\ y(\theta) = R\sin\theta. \end{cases}$$

In accordance with formula (2.1), its Jacobian is the 2-by-1 matrix

$$J = \begin{bmatrix} d_\theta x(\theta) \\ d_\theta y(\theta) \end{bmatrix} = \begin{bmatrix} -R\sin\theta \\ R\cos\theta \end{bmatrix},$$

and the metric tensor (2.2) is just a scalar

$$g_{ij} = J^\intercal J = \begin{bmatrix} -R\sin\theta & R\cos\theta \end{bmatrix} \begin{bmatrix} -R\sin\theta \\ R\cos\theta \end{bmatrix} = \begin{bmatrix} R^2 \end{bmatrix}.$$

Thus, Laplace-Beltrami operator (2.3) for the circle is just a second order derivative with respect to the angle scaled by the square of radius

$$\Delta_{\mathcal{M}} = \frac{1}{R}\frac{d}{d\theta}\left(R\frac{1}{R^2}\frac{d}{d\theta}\right) = \frac{1}{R^2}\frac{d^2}{d\theta^2}.$$

Example 2.2 (*Parametric planar curve: ellipse*)**.**

Consider an ellipse with semiaxes a and b parametrized as

$$\begin{cases} x(\theta) = a\cos\theta, \\ y(\theta) = b\sin\theta. \end{cases} \tag{2.4}$$

Its Jacobian (2.1) is the 2-by-1 matrix

$$J = \begin{bmatrix} d_\theta x(\theta) \\ d_\theta y(\theta) \end{bmatrix} = \begin{bmatrix} -a\sin\theta \\ b\cos\theta \end{bmatrix},$$

and the metric tensor (2.2) is just a scalar

$$g_{ij} = J^\intercal J = \begin{bmatrix} -a\sin\theta & b\cos\theta \end{bmatrix} \begin{bmatrix} -a\sin\theta \\ b\cos\theta \end{bmatrix} = a^2\sin^2\theta + b^2\cos^2\theta.$$

Then the Laplace-Beltrami operator (2.3) for an ellipse defined by parametrization (2.4) is

$$\begin{aligned}\Delta_{\mathcal{M}} &= \frac{1}{\sqrt{a^2\sin^2\theta + b^2\cos^2\theta}}\frac{d}{d\theta}\left(\frac{\sqrt{a^2\sin^2\theta+b^2\cos^2\theta}}{a^2\sin^2\theta+b^2\cos^2\theta}\frac{d}{d\theta}\right)\\ &= \frac{1}{a^2\sin^2\theta+b^2\cos^2\theta}\frac{d^2}{d\theta^2} - \frac{(a^2-b^2)\sin\theta\cos\theta}{\left(a^2\sin^2\theta+b^2\cos^2\theta\right)^2}\frac{d}{d\theta}. \qquad (2.5)\end{aligned}$$

Example 2.3 (*Implicit planar curve: ellipse*)**.**

For a m-dimensional manifold given implicitly as zero level-set of a $(m+1)$-dimensional function, the core idea is to extract the underlying parametrization and express Jacobian of that implied parametrization by the means of an implicit equation.

Consider an ellipse given via an implicit equation

$$\mathcal{M} = \left\{ (x,y) \mid \frac{x^2}{a^2} + \frac{y^2}{b^2} - 1 = 0 \right\}.$$

Denote $F(x,y) := x^2/a^2 + y^2/b^2 - 1$, where variables x and y are implicitly assumed to depend on some variable t, i.e.

$$F(x(t), y(t)) = x^2(t)/a^2 + y^2(t)/b^2 - 1 = 0. \qquad (2.6)$$

Taking derivative of the ellipse equation $F(x(t), y(t)) = 0$ with respect to the implied parametrization variable t yields

$$\frac{dF}{dt} = \begin{bmatrix}\partial_x F & \partial_y F\end{bmatrix}\begin{bmatrix}x_t\\ y_t\end{bmatrix} = 0$$

which means that the Jacobian (2.1) takes the form of 2-by-1 matrix

$$J = \begin{bmatrix}x_t\\ y_t\end{bmatrix} = c\begin{bmatrix}\partial_y F(x,y)\\ -\partial_x F(x,y)\end{bmatrix} = c\begin{bmatrix}2y/b^2\\ -2x/a^2\end{bmatrix}, \qquad (2.7)$$

where c is a constant and the metric tensor (2.2) is

$$g_{ij} = J^\intercal J = c^2\begin{bmatrix}2y/b^2 & -2x/a^2\end{bmatrix}\begin{bmatrix}2y/b^2\\ -2x/a^2\end{bmatrix} = c^2\left[\frac{4y^2}{b^4} + \frac{4x^2}{a^4}\right] = \frac{4c^2(b^4x^2 + a^4y^2)}{a^4b^4}$$

which is a scalar. Thus the *inverse metric* g^{ij} takes the form

$$g^{ij} = \left(g_{ij}\right)^{-1} = \frac{a^4 b^4}{4c^2 \left(b^4 x^2 + a^4 y^2\right)}.$$

The Laplace-Beltrami operator (2.3) for the ellipse defined implicitly via $F\left(x,y\right)=0$ is

$$\begin{aligned}\Delta_{\mathcal{M}} &= \frac{a^2 b^2}{2\sqrt{c^2\left(b^4x^2+a^4y^2\right)}}\frac{d}{dt}\left(\frac{2\sqrt{c^2\left(b^4x^2+a^4y^2\right)}}{a^2b^2}\frac{a^4b^4}{4c^2\left(b^4x^2+a^4y^2\right)}\frac{d}{dt}\right)\\ &= \frac{a^2 b^2}{2\sqrt{c^2\left(b^4x^2+a^4y^2\right)}}\frac{d}{dt}\left(\frac{a^2b^2}{2\sqrt{c^2\left(b^4x^2+a^4y^2\right)}}\frac{d}{dt}\right),\end{aligned}$$

where t is the parameter from (2.6) that can be chosen arbitrarily. The constant c can be determined once the parametrization is specified.

We can further simplify the expression for the Laplace-Beltrami operator on an ellipse if we can find the derivatives x_t and y_t. Indeed, by utilizing (2.7) we can rewrite above expression in the closed form as

$$\begin{aligned}\Delta_{\mathcal{M}} &= \frac{a^4 b^4}{4\sqrt{c^2\left(b^4x^2+a^4y^2\right)}}\frac{d}{dt}\left(\frac{1}{\sqrt{c^2\left(b^4x^2+a^4y^2\right)}}\frac{d}{dt}\right)\\ &= \frac{a^4b^4}{4c^2\left(b^4x^2+a^4y^2\right)}\left(\frac{d^2}{dt^2}-\frac{b^4xx_t+a^4yy_t}{b^4x^2+a^4y^2}\frac{d}{dt}\right).\end{aligned}$$

According to (2.7) $x_t = cF_y\left(x,y\right) = {}^{2cy}/_{b^2}$ and $y_t = -cF_x\left(x,y\right) = -{}^{2cx}/_{a^2}$, so we finally get

$$\Delta_{\mathcal{M}} = \frac{a^4b^4}{4c^2\left(b^4x^2+a^4y^2\right)}\left(\frac{d^2}{dt^2}-\frac{2\left(b^2-a^2\right)cxy}{b^4x^2+a^4y^2}\frac{d}{dt}\right). \tag{2.8}$$

Note. Observe that different parametrizations of the same planar curve in Examples 2.2 and 2.3 may result in different metric tensors, the resulting formulas (2.5) and (2.8) for Laplace-Beltrami operator will still yield the same expression if we choose the parametrization $t=\theta$ and set parameter $c=-ab/2$ in the equation (2.8).

After presenting simple examples for concrete curves, we proceed to explicit computations of Laplace-Beltrami operator formulas in slightly more abstract settings.

Example 2.4 (*Graph of a function*)**.**

For a planar curve of the form $y = f(x)$ one can compute Jacobian, metric tensor, and Laplacian in the same way as in Example 2.2 by introducing implicit function $F(x, y) := y - f(x)$, in which case the graph of the curve will be described as

$$F(x, y) = y - f(x) = 0,$$

and so the Jacobian takes the form

$$J = \begin{bmatrix} \partial_y F(x, y) \\ -\partial_x F(x, y) \end{bmatrix} = \begin{bmatrix} 1 \\ f_x \end{bmatrix}.$$

Then we have

$$g_{ij} = \begin{bmatrix} 1 & f_x \end{bmatrix} \begin{bmatrix} 1 \\ f_x \end{bmatrix} = 1 + f_x^2, \qquad |g| = 1 + f_x^2, \qquad g^{ij} = \frac{1}{1 + f_x^2},$$

and

$$\begin{aligned} \Delta_{\mathcal{M}} &= \frac{1}{\sqrt{1 + f_x^2}} \frac{d}{dx} \left(\sqrt{1 + f_x^2} \frac{1}{1 + f_x^2} \frac{d}{dx} \right) \\ &= \frac{1}{\sqrt{1 + f_x^2}} \frac{d}{dx} \left(\frac{1}{\sqrt{1 + f_x^2}} \frac{d}{dx} \right) \\ &= \frac{1}{1 + f_x^2} \frac{d^2}{dx^2} - \frac{f_x f_{xx}}{(1 + f_x^2)^2} \frac{d}{dx}. \end{aligned}$$

Example 2.5 (*Graph of a surface*)**.**

Similar to Example 2.4, we consider a surface given as $z = f(x, y)$. Differentiating the last equation with respect to x and y we get the expression for Jacobian

$$J = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ f_x & f_y \end{bmatrix}.$$

Then we have

$$g_{ij} = \begin{bmatrix} 1 & 0 & f_x \\ 0 & 1 & f_y \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ f_x & f_y \end{bmatrix} = \begin{bmatrix} 1 + f_x^2 & f_x f_y \\ f_x f_y & 1 + f_y^2 \end{bmatrix},$$

so that

$$|g| = \left|\left(1+f_x^2\right)\left(1+f_y^2\right) - f_x^2 f_y^2\right| = 1 + f_x^2 + f_y^2$$

and

$$g^{ij} = \left(g_{ij}\right)^{-1} = \begin{bmatrix} \frac{1+f_y^2}{1+f_x^2+f_y^2} & -\frac{f_x f_y}{1+f_x^2+f_y^2} \\ -\frac{f_x f_y}{1+f_x^2+f_y^2} & \frac{1+f_x^2}{1+f_x^2+f_y^2} \end{bmatrix}.$$

Then using (2.3) we write Laplace-Beltrami operator on the surface $z = f(x,y)$ as

$$\begin{aligned} \Delta_{\mathcal{M}} = & \frac{1}{\sqrt{1+f_x^2+f_y^2}} \frac{\partial}{\partial x}\left[\sqrt{1+f_x^2+f_y^2}\left(\frac{1+f_y^2}{1+f_x^2+f_y^2}\frac{\partial}{\partial x} - \frac{f_x f_y}{1+f_x^2+f_y^2}\frac{\partial}{\partial y}\right)\right] \\ & - \frac{1}{\sqrt{1+f_x^2+f_y^2}} \frac{\partial}{\partial y}\left[\sqrt{1+f_x^2+f_y^2}\left(\frac{f_x f_y}{1+f_x^2+f_y^2}\frac{\partial}{\partial x} - \frac{1+f_x^2}{1+f_x^2+f_y^2}\frac{\partial}{\partial y}\right)\right]. \end{aligned}$$

Simplifying expression above we get

$$\Delta_{\mathcal{M}} = \frac{1}{\sqrt{1+f_x^2+f_y^2}}\left(\frac{\partial}{\partial x}\left[\frac{\left(1+f_y^2\right)\partial_x - f_x f_y \partial_y}{\sqrt{1+f_x^2+f_y^2}}\right] - \frac{\partial}{\partial y}\left[\frac{f_x f_y \partial_x - \left(1+f_x^2\right)\partial_y}{\sqrt{1+f_x^2+f_y^2}}\right]\right).$$

Example 2.6 (*Implicit Surface*)**.**

For a two-dimensional surface given implicitly as

$$\mathcal{M} = \left\{(x,y,z) \in \mathbb{R}^3 \;\middle|\; F(x,y,z) = 0\right\}$$

the core idea is the same as in the case of implicit curve (see Example 2.3): to extract underlying parametrization of the surface $\mathcal{M}$ and express Jacobian in terms of the implicit equation. In order to implement this idea we assume underlying parametrization of Euclidean coordinates of manifold in terms of some variables s, t:

$$\begin{cases} x = x(s,t), \\ y = y(s,t), \\ z = z(s,t). \end{cases}$$

Differentiating equation $F(x,y,z) = 0$ with respect to s and t yields

$$\frac{\partial F}{\partial(s,t)} = \begin{bmatrix} \frac{\partial F}{\partial s} & \frac{\partial F}{\partial t} \end{bmatrix} = \begin{bmatrix} \frac{\partial F}{\partial x} & \frac{\partial F}{\partial y} & \frac{\partial F}{\partial z} \end{bmatrix} \begin{bmatrix} x_s & x_t \\ y_s & y_t \\ z_s & z_t \end{bmatrix} = 0.$$

Thus the Jacobian (2.1) is 3-by-2 matrix

$$J = \begin{bmatrix} x_s & x_t \\ y_s & y_t \\ z_s & z_t \end{bmatrix},$$

and the metric tensor (2.2) is

$$g_{ij} = J^\intercal J = \begin{bmatrix} x_s & y_s & z_s \\ x_t & y_t & z_t \end{bmatrix} \begin{bmatrix} x_s & x_t \\ y_s & y_t \\ z_s & z_t \end{bmatrix} = \begin{bmatrix} x_s^2 + y_s^2 + z_s^2 & x_s x_t + y_s y_t + z_s z_t \\ x_s x_t + y_s y_t + z_s z_t & x_t^2 + y_t^2 + z_t^2 \end{bmatrix}$$

with the inverse metric g^{ij}

$$g^{ij} = \left(g_{ij}\right)^{-1}.$$

2.2 Eigenvalue Problem for Laplace-Beltrami Operator

After reviewing the basic machinery necessary for understanding Laplace-Beltrami operator in the previous section, we can formally state the eigenvalue problem that is of the core interest of this work. Given linear differential operators $\mathcal{B}$ representing Dirichlet, Neumann, or Robin boundary conditions on manifold $\mathcal{M}$ the eigenvalue problem

$$\begin{cases} -\Delta_{\mathcal{M}} u = \lambda u, & x \in \mathcal{M}, \\ \mathcal{B}u = 0, & x \in \partial\mathcal{M}. \end{cases} \tag{2.9}$$

Our goal is to develop robust approach to solve above problem numerically using Radial Basis Functions (RBFs). Additionally, we aim to derive RBF-based numerical method that would inherit some of the analytical properties of Laplace-Beltrami operator and its eigenvalues. Some of these relevant properties are reviewed in the following theorem.

Theorem 1 (Properties of Laplace-Beltrami eigenmodes)**.** *The solution of the eigenvalue problem* (2.9) *with the appropriate boundary condition operator $\mathcal{B}$ satisfies the following properties [8]:*

(i) all eigenvalues are real,

(ii) all eigenfunctions can be chosen to be real-valued,

(iii) eigenfunctions corresponding to the distinct eigenvalues are also distinct and orthogonal.

2.3 Mean Curvature and Laplace-Beltrami Operator

In this section we explore the relation between Cartesian Laplacian of a function F in the embedding space $\mathbb{E}$ and Laplace-Beltrami operator of the restriction of F onto the manifold $\mathcal{M}$, which we denote f. Specifically, we quote two results which were proved by Reilly in 1982 for curve in $\mathbb{R}^2$ and surface in $\mathbb{R}^3$ [55].

Denote partial derivatives by subscripts such as

$$\frac{\partial F}{\partial x_j} = F_j, \quad \frac{\partial^2 F}{\partial x_j \partial x_k} = F_{kj}.$$

Denote the first and second directional directives along a vector $\vec{\boldsymbol{v}} = [v_1, v_2, \dots, v_n]^\intercal$ in $\mathbb{R}^n$ by $D_{\vec{\boldsymbol{v}}}$ and $D^2_{\vec{\boldsymbol{v}}}$ respectively:

$$D_{\vec{\boldsymbol{v}}} F(p) = \frac{d}{dt} F(p + t\vec{\boldsymbol{v}})\Big|^{t=0} = \sum_{j=1}^{n} F_j(p) v_j,$$

and

$$D^2_{\vec{\boldsymbol{v}}} F(p) = \frac{d^2}{dt^2} F\left(p + t\vec{\boldsymbol{v}}\right)\Big|^{t=0} = \sum_{j,k=1}^{n} F_{jk}(p) v_j v_k.$$

The following two theorems are true.

Theorem 2. *Suppose that F is a $\mathcal{C}^2$ function on an open set U in $\mathbb{R}^2$, Let p be a point of U and let F be a curve in U passing through p. Denote the restriction of F to Γ by f. Let $\vec{\boldsymbol{n}}_p$ be a unit normal to Γ at p and let $\kappa(p)$ be the corresponding curvature. Denote the arc* *length along* Γ *by* s*. Then*

$$(\Delta F)(p) = \frac{d^2 f}{ds^2}(p) - \kappa(p) D_{\vec{\boldsymbol{n}}_p} F(p) + D^2_{\vec{\boldsymbol{n}}_p} F(p). \tag{2.10}$$

Theorem 3. *Suppose that* G *is a* $\mathcal{C}^2$ *function on an open set* W *in* $\mathbb{R}^3$*, Let* p *be a point of* W *and let* S *be an oriented* $\mathcal{C}^2$ *surface in* W *passing through* p*. Denote the restriction of* G *to* S *by* g*. Let* $\vec{\boldsymbol{n}}_p$ *be a unit normal to* S *at* p *and let* $H(p)$ *be the mean curvature. Denote the surface Laplacian by* $\Delta_{\mathcal{M}}$*. Then*

$$(\Delta G)(p) = \Delta_{\mathcal{M}} g(p) - 2H(p) D_{\vec{\boldsymbol{n}}_p} G(p) + D^2_{\vec{\boldsymbol{n}}_p} G(p). \tag{2.11}$$

Example 2.7. Taking Γ as a circle of radius r with center $(0,0)$, the formula (2.10) is just the classical formula for the Laplacian in polar coordinates

$$\Delta F(p) = \frac{1}{r}\frac{\partial^2 F}{\partial \theta^2}(p) + \frac{1}{r}\frac{\partial F}{\partial r}(p) + \frac{\partial^2 F}{\partial r^2}(p).$$

Example 2.8. Taking $\mathcal{M}$ as a sphere of radius r with center $(0,0,0)$, the formula (2.11) is the classical formula for the Laplacian in spherical coordinates

$$\Delta F(p) = \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\big(\sin\theta \, G_\theta\big) + \frac{1}{r^2 \sin^2\theta} G_{\phi\phi} + \frac{2}{r} G_r + G_{rr}.$$

The formulas (2.10) and (2.11) provide an alternative way to compute the Laplace-Beltrami operator on a curve Γ

$$\Delta_\Gamma f(p) = (\Delta F)(p) + \kappa(p) D_{\vec{\boldsymbol{n}}_p} F(p) - D^2_{\vec{\boldsymbol{n}}_p} F(p), \tag{2.12}$$

and on a two-dimensional surface $\mathcal{M}$

$$\Delta_{\mathcal{M}} g(p) = (\Delta G)(p) + 2H(p) D_{\vec{\boldsymbol{n}}_p} G(p) - D^2_{\vec{\boldsymbol{n}}_p} G(p). \tag{2.13}$$

2.3.1 Curvature Expressions for Planar Curves and Surfaces

In order to utilize formulas (2.12) and (2.13) for more general curves and surfaces respectively, the *closed-form expressions* for computing different types curvatures are required. Below we provide several relevant formulas which are presented and derived in [23] by Goldman.

Some definitions and geometrical insights into the concept of curvature applied to planar curves and surfaces can be found in Appendix A. Below we will provide several curvature formulas [23] which are relevant to our examples.

Implicit Planar Curves

For an implicit planar curve defined as $F\left(x,y\right)=0$ we adopt the following notation:

- unit normal vector $\vec{\boldsymbol{n}} = \left[F_x,\ F_y\right]^\intercal / \sqrt{F_x^2+F_y^2}$,
- unit tangent vector $\vec{\boldsymbol{t}} = \left[-F_y,\ F_x\right]^\intercal / \sqrt{F_x^2+F_y^2}$,
- hessian $H = \begin{bmatrix} F_{xx} & F_{xy} \\ F_{xy} & F_{yy} \end{bmatrix}$.

Then the following statement holds [23]:

Statement 1. The *curvature* of an implicit planar curve $F\left(x,y\right)=0$ can be computed as

$$\kappa = -\frac{\vec{\boldsymbol{t}}^\intercal \cdot H \cdot \vec{\boldsymbol{t}}}{|\nabla F|} = -\left(F_x^2+F_y^2\right)^{-\frac{3}{2}} \begin{bmatrix} -F_y & F_x \end{bmatrix} \begin{bmatrix} F_{xx} & F_{xy} \\ F_{xy} & F_{yy} \end{bmatrix} \begin{bmatrix} -F_y \\ F_x \end{bmatrix}. \tag{2.14}$$

Example 2.9 (*Curvature of an ellipse*)**.**

For an ellipse

$$\mathcal{M} = \left\{ \left(x,y\right) \middle| \, 1 - x^2/a^2 - y^2/b^2 = 0 \right\}$$

$F_x = -2x/a^2,\ F_y = -2y/b^2,\ F_{xx} = -2/a^2,\ F_{yy} = -2/b^2$, and $F_{xy} = 0$. Then the curvature can be computed via formula (2.14) as

$$
\begin{aligned}
\kappa &= -\left(\frac{4x^2}{a^4}+\frac{4y^2}{b^4}\right)^{-\frac{3}{2}} \begin{bmatrix} \frac{2y}{b^2} & -\frac{2x}{a^2} \end{bmatrix} \begin{bmatrix} -2/a^2 & 0 \\ 0 & -2/b^2 \end{bmatrix} \begin{bmatrix} 2y/b^2 \\ -2x/a^2 \end{bmatrix} \\
&= \quad \left(\frac{x^2}{a^4}+\frac{y^2}{b^4}\right)^{-\frac{3}{2}} \frac{b^2x^2+a^2y^2}{2a^4b^4}.
\end{aligned}
$$

Example 2.10 (*Laplace-Beltrami operator on an ellipse*)**.**

Similar to Example 2.3, consider

$$
\mathcal{M} = \Big\{(x,y)\in\mathbb{R}^2 \;\Big|\; 1 - x^2/a^2 - y^2/b^2 = 0\Big\}.
$$

The unit normal vector can be computed as

$$
\vec{\boldsymbol{n}} = \begin{bmatrix} n_1 \\ n_2 \end{bmatrix} = \frac{-1}{\sqrt{(x/a^2)^2+(y/b^2)^2}} \begin{bmatrix} x/a^2 \\ y/b^2 \end{bmatrix},
$$

and the curvature is

$$
\kappa = \nabla\cdot\vec{\boldsymbol{n}} = \left(\frac{x^2}{a^4}+\frac{y^2}{b^4}\right)^{-\frac{3}{2}} \frac{b^2x^2+a^2y^2}{2a^4b^4}.
$$

Furthermore,

$$
\begin{aligned}
D_{\boldsymbol{n}} &= \vec{\boldsymbol{n}}\cdot\nabla = n_1\partial_x + n_2\partial_y = -\frac{x/a^2}{\sqrt{x^2/a^4+y^2/b^4}}\frac{\partial}{\partial x} - \frac{y/b^2}{\sqrt{x^2/a^4+y^2/b^4}}\frac{\partial}{\partial y}, \\
D_{\boldsymbol{nn}}F &= n_1^2\partial_{xx} + 2n_1n_2\partial_{xy} + n_2^2\partial_{yy} \\
&= \frac{x^2/a^4}{x^2/a^4+y^2/b^4}\frac{\partial^2}{\partial x^2} + 2\frac{xy/a^2b^2}{x^2/a^4+y^2/b^4}\frac{\partial^2}{\partial x\partial y} + \frac{y^2/b^4}{x^2/a^4+y^2/b^4}\frac{\partial^2}{\partial y^2}.
\end{aligned}
$$

Thus according to Reilly's formula (2.12) the Laplace-Beltrami operator on an ellipse takes the form

$$
\begin{aligned}
\Delta_\Gamma =& \Delta + \kappa D_{\boldsymbol{n}} - D_{\boldsymbol{nn}} \\
\Delta_\Gamma =& \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \left(\frac{x^2}{a^4}+\frac{y^2}{b^4}\right)^{-2} \frac{b^2x^2+a^2y^2}{2a^4b^4}\left(\frac{x}{a^2}\frac{\partial}{\partial x}+\frac{y}{b^2}\frac{\partial}{\partial y}\right) \\
&\qquad - \left(\frac{x^2}{a^4}+\frac{y^2}{b^4}\right)^{-1}\left(\frac{x^2}{a^4}\frac{\partial^2}{\partial x^2} + \frac{2xy}{a^2b^2}\frac{\partial^2}{\partial x\partial y} + \frac{y^2}{b^4}\frac{\partial^2}{\partial y^2}\right).
\end{aligned}
$$

Implicit Surfaces

Similar to the previous case, given an implicit surface $F(x, y, z) = 0$ and denoting unit normal vector $\vec{\boldsymbol{n}} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^{\intercal} / \sqrt{F_x^2 + F_y^2 + F_z^2}$, the following statement holds [23]:

Statement 2 (Divergence formula for the mean curvature)**.**

The *mean curvature* of a surface given by $F(x, y, z) = 0$ can be computed as *divergence of a unit normal vector*:

$$\kappa_m = -\frac{1}{2}\nabla \cdot \vec{\boldsymbol{n}} = -\frac{1}{2}\nabla \cdot \big(\nabla F / |\nabla F|\big). \tag{2.15}$$

Example 2.11 (Surface)**.**

Given an implicit surface

$$\mathcal{M} = \Big\{ (x, y, z) \in \mathbb{R}^3 \Big|\, f(x, y) - z = 0 \Big\}$$

for some function $f : \mathbb{R}^2 \to \mathbb{R}$ with unit normal vector $\vec{\boldsymbol{n}} = [\, f_x,\ f_y,\ -1\,]^{\intercal} / \sqrt{f_x^2 + f_y^2 + 1}$. Its curvature can be computed via formula (2.15) as

$$\begin{aligned} \kappa &= -\nabla \cdot \big(\nabla F / |\nabla F|\big)/2 \\ &= -\frac{\partial}{\partial x}\left(\frac{f_x}{2\sqrt{f_x^2 + f_y^2 + 1}}\right) - \frac{\partial}{\partial y}\left(\frac{f_y}{2\sqrt{f_x^2 + f_y^2 + 1}}\right) + \frac{\partial}{\partial z}\left(\frac{1}{2\sqrt{f_x^2 + f_y^2 + 1}}\right) \\ &= -\frac{\left(1 + f_y^2\right) f_{xx} - 2 f_x f_y f_{xy} + \left(1 + f_x^2\right) f_{yy}}{2\left(f_x^2 + f_y^2 + 1\right)^{\frac{3}{2}}}. \end{aligned}$$

2.4 Asymptotic Analysis of Eigenvalues of Laplace-Beltrami Operator

The asymptotic results show that the eigenvalues of Laplace operator with Neumann boundary conditions on a thin ring/shell converge to the eigenvalues of Laplace-Beltrami operator on a circle/sphere.

The asymptotic expansions for eigenvalues obtained in this section motivate the numerical study via embedded methods and radial basis methods discussed in chapter 4.

2.4.1 One-Dimensional Laplace-Beltrami Eigenvalue Problem

Consider the eigenvalues of Laplace-Beltrami operator

$$-\Delta_{\mathcal{M}} u = \lambda u$$

on a circle with radius $r = R$ in two dimensions. The eigenvalue problem can be written as

$$-y_{\theta\theta} = \lambda R^2 y, \quad y(0) = y(2\pi), \quad y'(0) = y'(2\pi)$$

which has eigenvalues

$$\lambda R^2 = n^2, \quad n \in \mathbb{N}.$$

When $n = 0$, the eigenfunction is a constant. When n is a positive integer, there are two corresponding eigenfunctions $\sin(n\theta)$ and $\cos(n\theta)$.

2.4.2 Two-Dimensional Laplace-Beltrami Eigenvalue Problem

Denote B_R as a circle with radius R in two dimensions and a sphere with radius R in three dimensions. Consider eigenvalues of Laplace-Beltrami operator

$$-\Delta_{\mathcal{M}} u = \lambda u$$

on a sphere $\mathcal{M} = B_R$ in three dimensions. In the spherical coordinates, the eigenvalue problem becomes

$$-\left(\frac{1}{R^2 \sin\theta} \frac{\partial}{\partial\theta} \left(\sin\theta u_\theta\right) + \frac{1}{R^2 \sin^2\theta} u_{\phi\phi} \right) = \lambda u.$$

The eigenvalues are

$$\lambda R^2 = l(l+1), \quad l \in \mathbb{N},$$

and eigenfunctions are spherical harmonic functions.

2.4.3 Two-Dimensional Eigenvalue Problem on a Thin ε Ring.

Consider the eigenvalue of Laplace operator

$$-\Delta u = \lambda u$$

on $\Omega = B_{R+\varepsilon} \setminus B_{R-\varepsilon}$ with Neumann boundary conditions

$$\left.\frac{\partial u}{\partial r}\right|_{r=R\pm\varepsilon} = 0.$$

In the polar coordinates, the equation becomes

$$-\left(u_{rr} + \frac{1}{r}u_r + \frac{1}{r^2}u_{\theta\theta}\right) = \lambda u.$$

Looking for separable solutions $u\left(r,\theta\right) = R\left(r\right)\Theta\left(\theta\right)$ gives

$$-\left(R^{''}\Theta + \frac{1}{r}R'\Theta + \frac{1}{r^2}R\Theta^{''}\right) = \lambda R\Theta,$$
$$-\left(\frac{R^{''}}{R} + \frac{1}{r}\frac{R^{'}}{R} + \frac{1}{r^2}\frac{\Theta^{''}}{\Theta}\right) = \lambda.$$

We must have

$$\frac{\Theta^{''}}{\Theta} = -n^2,$$

where n is the integer and

$$r^2R^{''} + R^{'} + (\lambda r^2 - n^2)R = 0.$$

The equations has the general solution in the following form

$$R = c_1J_n(\sqrt{\lambda}r) + c_2Y_n(\sqrt{\lambda}r).$$

The boundary conditions require

$$R'|_{r=R+\varepsilon} = 0 = c_1J_n^{'}(\sqrt{\lambda}(R+\varepsilon)) + c_2Y_n'(\sqrt{\lambda}(R+\varepsilon)),$$
$$R'|_{r=R-\varepsilon} = 0 = c_1J_n^{'}(\sqrt{\lambda}(R-\varepsilon)) + c_2Y_n'(\sqrt{\lambda}(R-\varepsilon)).$$

In order to have nontrivial solution, we must have

$$\det \begin{pmatrix} J_n'(\sqrt{\lambda}(R-\varepsilon)) & Y_n'(\sqrt{\lambda}(R-\varepsilon)) \\ J_n'(\sqrt{\lambda}(R+\varepsilon)) & Y_n'(\sqrt{\lambda}(R+\varepsilon)) \end{pmatrix} = 0. \tag{2.16}$$

A similar problem has been studied by Gottlieb [25], McMahon [49], and Buchholz [4] on $\Omega = B_b \setminus B_a$ where $r = {}^{b}/_{a} \to 1$.

Denote $x = \sqrt{\lambda}(R-\varepsilon)$, $\rho x = \sqrt{\lambda}(R+\varepsilon)$. We have

$$\gamma = \frac{R+\varepsilon}{R-\varepsilon} = 1 + 2\frac{\varepsilon}{R} + \frac{2\varepsilon^2}{R^2} + O(\varepsilon)^3,$$

which leads to

$$\gamma - 1 = 2\frac{\varepsilon}{R} + \frac{2\varepsilon^2}{R^2} + O(\varepsilon)^3.$$

Here γ is the ratio of outer radius of the ring to the inner radius. The asymptotic expansions of zero roots in terms of γ have the following forms. When $n \neq 0$, the first zero is given as

$$\sqrt{\lambda_n^1}(R-\varepsilon) = n\left[1 - \frac{1}{2}(\gamma-1) + \frac{7}{24}(\gamma-1)^2 + O(\gamma-1)^3\right],$$

where the asymptotic expansion is obtained from [25, equation (A.5)].

Rearranging terms, we get expression for the eigenvalues λ_n^1:

$$\begin{aligned} \lambda_n^1 &= \frac{n^2}{R^2(1-{}^{\varepsilon}/_{R})^2}\left[1 - \frac{1}{2}(\gamma-1) + \frac{7}{24}(\gamma-1)^2 + O(\gamma-1)^3\right]^2 \\ &\approx \frac{n^2}{R^2}\left(1 + \frac{\varepsilon^2}{3R^2} + O(\varepsilon^3)\right). \end{aligned} \tag{2.17}$$

On the other hand, for $n \geq 2$ we have

$$\begin{aligned} \sqrt{\lambda_0^{(s)}}(R-\varepsilon) &= \frac{\pi s}{\gamma-1} + \frac{3}{8\pi\gamma}\left(\frac{\gamma-1}{s}\right) + O\left(\frac{\gamma-1}{s}\right)^3, \\ \sqrt{\lambda_n^{(s+1)}}(R-\varepsilon) &= \frac{\pi s}{\gamma-1} + \frac{4\nu^2+3}{8\pi\gamma}\left(\frac{\gamma-1}{s}\right) + O\left(\frac{\gamma-1}{s}\right)^3, \end{aligned} \tag{2.18}$$

where $n \neq 0$, $s \in \mathbb{N}^+$. These formulas come from [25, equation (A.4)].

Since $\lambda_n^{(s)} \to \infty$ as $n \to \infty$, the assymptotic expansion (2.18) for eigenvalues go to infinity.

Thus we conclude that λ_n^1 in (2.17) go to n^2/R^2 which are eigenvalues of Laplace-Beltrami operator on a circle with radius R.

2.4.4 Three-Dimensional Eigenvalue Problem on a Thin ε Shell.

Consider following eigenvalue problem

$$-\Delta u = \lambda u$$

for the Laplace operator $\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}$ defined on $\Omega = B_{R+\varepsilon} \backslash B_{R-\varepsilon}$. In the spherical coordinates, the equation becomes

$$-\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2 u_r\right) + \frac{1}{r^2 \sin\theta}\frac{\partial}{\partial \theta}\left(\sin\theta u_\theta\right) + \frac{1}{r^2\sin^2\theta}u_{\phi\phi}\right) = \lambda u.$$

We assume a variables separable solution of the form

$$u(r,\theta,\phi) = R(r)Y(\theta,\phi)$$

where $Y(\theta,\phi)$ is a spherical harmonic function which satisfies

$$\frac{1}{Y}\frac{1}{\sin\theta}\frac{\partial}{\partial\theta}\left(\sin\theta Y_\theta\right) + \frac{1}{Y}\frac{1}{\sin^2\theta}Y_{\phi\phi} = -l(l+1)$$

where l is a constant. Thus we have the equation in the radial direction satisfies

$$r^2 R^{''}(r) + 2rR^{'}(r) + \left(\lambda r^2 - l(l+1)\right)R = 0.$$

Denote $\sqrt{\lambda} r = x$. This leads to

$$x^2 R^{''}(x) + 2xR^{'}(x) + \left(r^2 - l(l+1)\right)R(x) = 0.$$

Thus the solution are given as spherical Bessel function

$$R(x) = c_1 j_n(\sqrt{\lambda} r) + c_2 y_n(\sqrt{\lambda} r).$$

The boundary conditions require

$$R'|_{r=R+\varepsilon} = 0 = c_1 j_n^{'}(\sqrt{\lambda}(R+\varepsilon)) + c_2 y_n^{'}(\sqrt{\lambda}(R+\varepsilon)),$$

$$R'|_{r=R-\varepsilon} = 0 = c_1 j_n^{'}(\sqrt{\lambda}(R-\varepsilon)) + c_2 y_n'(\sqrt{\lambda}(R-\varepsilon)).$$

In order to have nontrivial solutions, we must have

$$\det \begin{pmatrix} j_n^{'}(\sqrt{\lambda}(R-\varepsilon)) \; y_n'(\sqrt{\lambda}(R-\varepsilon)) \\ j_n^{'}(\sqrt{\lambda}(R+\varepsilon)) \; y_n'(\sqrt{\lambda}(R+\varepsilon)) \end{pmatrix} = 0. \tag{2.19}$$

Recall that $x = \sqrt{\lambda}(R-\varepsilon)$, $\rho x = \sqrt{\lambda}(R+\varepsilon)$, which leads to

$$\gamma = \frac{R+\varepsilon}{R-\varepsilon} = 1 + 2\frac{\varepsilon}{R} + \frac{2\varepsilon^2}{R^2} + O(\varepsilon)^3,$$
$$\gamma - 1 = 2\frac{\varepsilon}{R} + \frac{2\varepsilon^2}{R^2} + O(\varepsilon)^3.$$

Here γ is the ratio of outer radius of the ring to the inner radius. The solutions have the following asymptotic forms [25, equation (5.4)]

$$\sqrt{\lambda_n^1}(R-\varepsilon) = \sqrt{n(n+1)}\left\{1 - \frac{1}{2}(\gamma-1) + \frac{5}{24}(\gamma-1)^2 + O(\gamma-1)^3\right\},$$

Rearranging terms, we obtain the following expression

$$\lambda_n^1 = \frac{n(n+1)}{R^2(1-\frac{\varepsilon}{R})^2}\left\{1 - \frac{1}{2}(\gamma-1) + \frac{5}{24}(\gamma-1)^2 + O(\gamma-1)^3\right\}^2$$

if $n \neq 0\,(-1)$, and

$$\sqrt{\lambda_n^{(s')}}(R-\varepsilon) = \frac{\pi s}{\gamma-1} + \frac{4\left(n+\frac{1}{2}\right)^2 + 7}{8\pi\gamma}\left(\frac{\gamma-1}{s}\right) + O\left(\frac{\gamma-1}{s}\right)^3,$$

where $s' = s$, $\nu = 0\,(-1)$, and $s' = s+1$, $\nu \neq 0\,(-1)$, which comes from [25, equation (A.9)]. Thus we have

$$\lambda_n^1 \approx \frac{n(n+1)}{R^2}\left(1 - \frac{\varepsilon^2}{3R^2} + O(\varepsilon^3)\right) \tag{2.20}$$

which converge to eigenvalues of Laplace-Beltrami operator on sphere.

In Figures 2.1–2.2, we show the solutions of (2.16) and (2.19) obtained by finding roots numerically for 40 different choices of ε which are 50 logarithmically equally spaced points between decades 10^{-4} and 10^{-2} in red dots and the asymptotic expansion solutions in formulas (2.17) and (2.20) up to second order terms. It is clear that the asymptotic results agree well with the numerical results.

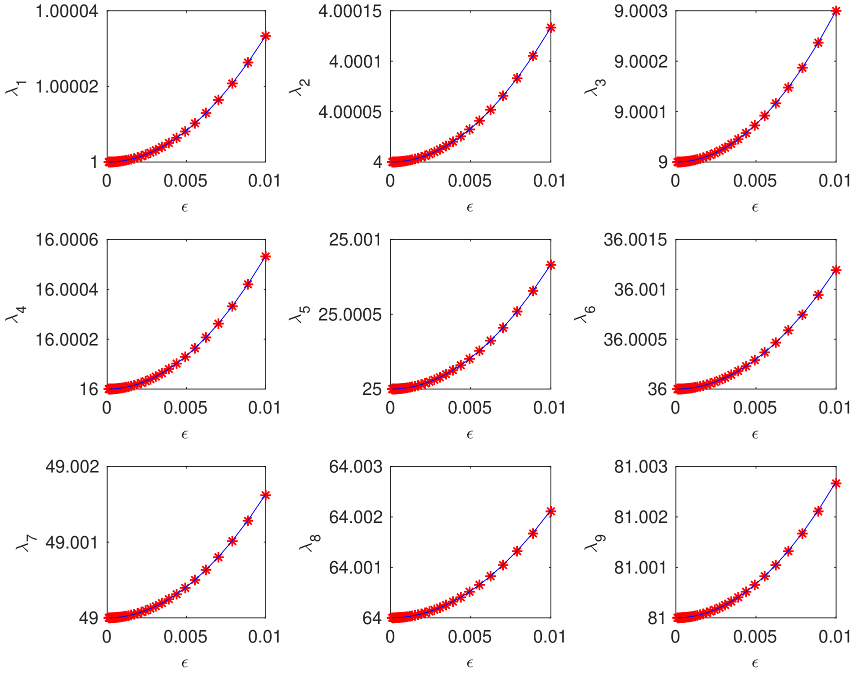


Figure 2.1: The solutions of (2.16) obtained by finding roots numerically for 40 different choices of ε which are 50 logarithmically equally spaced points between decades 10^{-4} and 10^{-2} in red dots and the asymptotic expansion solutions in formula (2.17) up to second order terms.

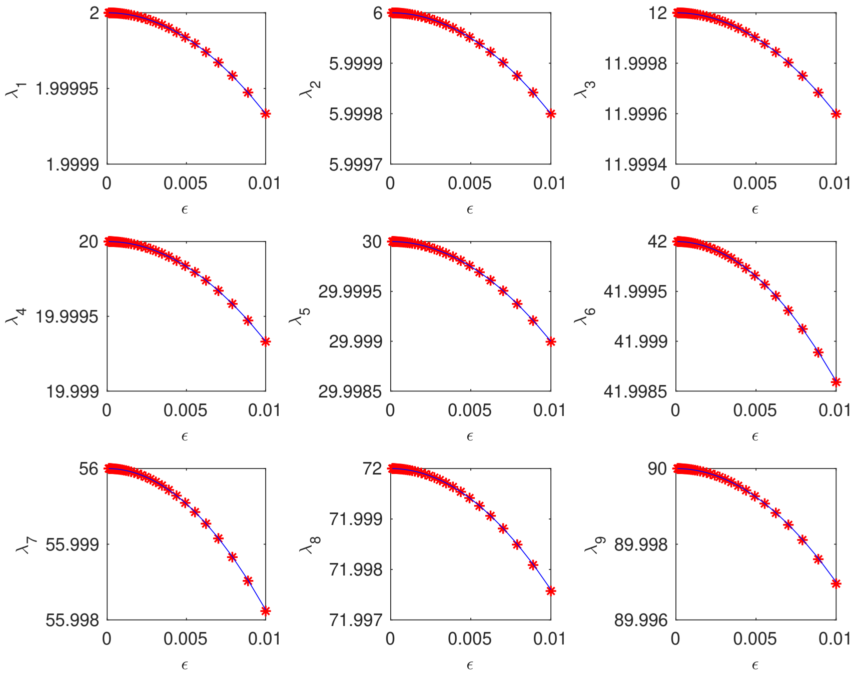


Figure 2.2: The solutions of (2.19) obtained by finding roots numerically for 40 different choices of ε which are 50 logarithmically equally spaced points between decades 10^{-4} and 10^{-2} in red dots, and the asymptotic expansion solutions in formula (2.20) up to second order terms.

Chapter 3

Existing RBF Methods for Solving Differential Equations on Manifolds

The scope of this work is build upon the concept of Partial Differential Equations defined on manifolds. Similarly to the case of flat domain, PDEs on manifolds can be time-dependent or stationary. Our intention is to focus on stationary linear differential operators defined on manifolds, as we are particularly interested in solving eigenproblems involving such operators. One example of such problems would be finding eigenmodes of Laplace-Beltrami operator given underlying domain (manifold) and possibly boundary conditions.

3.1 History of RBF Approaches to Solving PDEs

The first numerical algorithm for solving Partial Differential Equations on surfaces using Radial Basis Functions was published by Kansa [31] in 1990. That approach to solving PDE problems on manifolds formulated in the strong form by collocation quickly gained popularity and became known as *Kansa's method* in academic community. The biggest advantages of the proposed approach include intuitive underlying mathematical idea as well as simplicity of both formulation and implementation of algorithm. Inherently meshless, Kansa's method works in any number of dimensions and does not have restrictions on geometrical complexity of underlying manifold.

Although originally formulated only for the multiquadric radial functions, the RBF-based meshfree methods have been successfully used with other radial functions [17]. Popular choices of (global) continuously differentiable basis functions include

- Gaussian $e^{-(\varepsilon r)^2}$
- Multiquadric $\sqrt{1+(\varepsilon r)^2}$
- Inverse multiquadric $1/\sqrt{1+(\varepsilon r)^2}$
- Inverse quadratic $1/\left(1+(\varepsilon r)^2\right)$
- Polyharmonic spline $\phi\left(r\right)=r^k\ln\left(r\right), \quad k=2,4,6,\ldots$

More recent developments of RBF- and kernel-based methods of function approximation and solving PDEs on manifolds have been focused on use of implicit compact RBF-FD schemes to improve stability, accuracy [35], and computational cost [63]. Additionally, recently introduced eigen-rational kernel-based scheme [5] takes advantage of rescaling the classical RBF expansion to provide more robust approximations.

Another prominent trend in RBF community is the search for the formulation of the method that only requires scattered points on manifold without explicit analytical representation. For example, see [2] for the closed-form formulations of orthogonal projection method. In addition, several recent articles presented hybrid methods that incorporate features of RBF approach alongside embedding methods [53].

3.1.1 Overview of Related Numerical Methods

In general, modern RBF-based meshfree techniques for solving eigenvalue problems rely in their core either on *orthogonal projection* or on *orthogonal gradient* for relating surface differential operator and regular differentiation in the embedding Euclidean space $\mathbb{R}^n$. We explore these methods in the rest of the chapter.

3.2 Mathematical Formulation of Eigenvalue Problem

Consider linear differential operators $\mathcal{L} : \mathcal{C}^\infty(\mathcal{M}) \to \mathcal{C}^\infty(\mathcal{M})$ of order $p \in \mathbb{N}^+$, where $\mathcal{M}$ is $(n-1)$ dimensional compact Riemannian manifold with smooth boundary $\partial\mathcal{M}$ embedded into Euclidean Space $\mathbf{E} = \mathbb{R}^n$. Denote $\mathcal{M}_I$ interior of $\mathcal{M}$, so $\mathcal{M} = \mathcal{M}_I \cup \partial\mathcal{M}$.

Assuming $u \in \mathcal{C}^\infty(\mathcal{M})$, we want to solve the following eigenvalue problem:

$$\begin{cases} \mathcal{L}(u)\,(x) = \lambda u\,(x)\,, & x \in \mathcal{M}_I, \\ \mathcal{B}(u)(x) = 0, & x \in \partial\mathcal{M}, \end{cases} \tag{3.1}$$

where $\mathcal{B} : \mathcal{C}^\infty(\partial\mathcal{M}) \to \mathcal{C}^\infty(\partial\mathcal{M})$ is boundary condition operator.

For examples, Laplacian and bilaplacian eigenmode problems take forms

$$\begin{cases} -\Delta u = \lambda u, & x \in \mathcal{M}_I \\ u = 0, & x \in \partial\mathcal{M} \end{cases} \quad \text{or} \quad \begin{cases} \Delta^2 u = \lambda u, & x \in \mathcal{M}_I \\ u = 0, & x \in \partial\mathcal{M} \\ \partial u/\partial\vec{\boldsymbol{b}} = 0, & x \in \partial\mathcal{M} \end{cases}, \tag{3.2}$$

where $\vec{\boldsymbol{b}}$ is the binormal unit vector representing flux through the boundary, which is defined as $\vec{\boldsymbol{b}} = \vec{\boldsymbol{t}}_{\partial\mathcal{M}} \times \vec{\boldsymbol{n}}$. Here $\vec{\boldsymbol{t}}_{\partial\mathcal{M}}$ is the tangent vector of the boundary $\partial\mathcal{M}$ and $\vec{\boldsymbol{n}}$ is the unit normal vector of the manifold $\mathcal{M}$.

Remark. If $\mathcal{M}$ is a plane, our problem degenerates to a classic flat domain one, and if $\mathcal{M}$ is closed, we get an eigenvalue problem without boundary.

3.3 RBF Orthogonal Gradient Method

Given an arbitrary manifold we can approximate it by the linear combination of a family of Radial Basis Functions.

3.3.1 Surface Reconstruction

Consider a two-dimensional orientable manifold embedded into three-dimensional Euclidean space which is given in the implicit form $s(x) = 0$. We intend to interpolate it using a radial basis function ϕ:

$$s(x) = \sum_{i=1}^{N} a_i \phi\left(\|x - \xi_i\|\right),$$

where ξ_i are fixed points on the manifold, a_i are the interpolation coefficients, and $\|\cdot\|$ is a Euclidian norm.

In order to do that, we represent our two-dimensional manifold as an isosurface of the three-dimensional one. We can do so by appending $2N$ nodes to the N ones which we already have. These additional $2N$ nodes will represent two extra isosurfaces of the master three-dimensional manifold. For convenience we can call them *plus*-layer and *minus*-layer implying that one of them is located “inside” of the original manifold and another one is “outside”. In fact, the key point is that these layers are on the different sides of the manifold.

When new points are introduced on the extra layers we make sure that each of them lies on the normal vector of the corresponding point on the manifold. The distance between them is denoted as δ.

We do not reconstruct extra layers, we only use points on them in order to provide extra nodes for interpolation of the manifold by RBFs. In this way, our reconstruction will look like

$$d(x) = \sum_{i=1}^{N} a_i \phi\left(\|x - \xi_i\|\right) + b_i \phi\left(\|x - \xi_i + \delta \vec{\boldsymbol{n}}_{\xi_i}\|\right) + c_i \phi\left(\|x - \xi_i - \delta \vec{\boldsymbol{n}}_{\xi_i}\|\right),$$

so that $d\left(\xi_i\right) = 0$, $d\left(\xi_i^+\right) = 1$ and $d\left(\xi_i^-\right) = -1$, where $\xi_i^+ = \xi_i + \delta \vec{\boldsymbol{n}}_{\xi_i}$ and $\xi_i^- = \xi_i - \delta \vec{\boldsymbol{n}}_{\xi_i}$.

3.3.2 Differentiation

If we have a function f defined on the manifold we can also reconstruct it by using RBFs:

$$f(x) = \sum_{i=1}^{N} \alpha_j \phi\left(\|x - \xi_i\|\right).$$

Now, we can apply the differentiation operator:

$$g(x) = \mathcal{L} f(x) = \mathcal{L} \sum_{i=1}^{N} \alpha_j \phi(||x - \xi_i||) = \sum_{i=1}^{N} \alpha_j \mathcal{L} \phi\left(||x - \xi_i||\right).$$

In the matrix form, if $\{A_{i,j}\} = \phi\left(\|x_i - \xi_j\|\right)$ and $\{B_{i,j}\} = \mathcal{L}\phi\left(\|x_i - \xi_j\|\right)$, we will have

$$f = A\vec{\boldsymbol{\alpha}}, \quad g = B\vec{\boldsymbol{\alpha}},$$

We introduce the differentiation matrix D which corresponds to the differentiation operator $\mathcal{L}$:

$$g = B\vec{\boldsymbol{\alpha}}, \quad \vec{\boldsymbol{\alpha}} = A^{-1} f \implies g = BA^{-1} f.$$

This leads to

$$\mathcal{L} \approx D = BA^{-1}. \tag{3.3}$$

Note. The matrix A is non-singular, so the differentiation matrix D is well-defined.

3.3.3 Surface Operators

Let us consider a gradient ∇f and represent it in the orthogonal systems of coordinates $\{\vec{\boldsymbol{n}}, \vec{\boldsymbol{t}}_1, \vec{\boldsymbol{t}}_2\}$ of the normal, first tangent and second tangent directions. We will get

$$\nabla f = \partial_n f \vec{\boldsymbol{n}} + \partial_{t_1} f \vec{\boldsymbol{t}}_1 + \partial_{t_2} f \vec{\boldsymbol{t}}_2.$$

We can convert it into the surface gradient $\nabla_{\mathcal{M}}$ if we eliminate the normal component, so

$$\nabla_{\mathcal{M}} f = \partial_{t_1} f \vec{\boldsymbol{t}}_1 + \partial_{t_2} f \vec{\boldsymbol{t}}_2.$$

Two main techniques for constructing surface differential operators are the Projection and the Orthogonal Gradient methods. Here we focus on the Orthogonal Gradient method.

Now, let us consider the Laplacian operator

$$\Delta f = \nabla^2 f = \left(\vec{\boldsymbol{n}}\partial_n + \vec{\boldsymbol{t}}_1\partial_{t_1} + \vec{\boldsymbol{t}}_2\partial_{t_2}\right)\left(\vec{\boldsymbol{n}}\partial_n + \vec{\boldsymbol{t}}_1\partial_{t_1} + \vec{\boldsymbol{t}}_2\partial_{t_2}\right) f$$

which can be transformed into the Laplace-Beltrami operator by eliminating both $\partial_n f$ and $\partial_n^2 f$ components. There are two ways to do that.

3.3.4 Low Order

In the section 3.3.1 we established that we have N points representing our surface and $2N$ additional nodes. For every node on the main surface we have two additional nodes along its positive and negative estimated normal direction. Then, by calculating three-dimensional Laplacian of a function f satisfying $f(\xi_i^{\pm}) = f(\xi_i)$ we obtain a rough estimation of the Laplace-Beltrami operator of f on $\mathcal{M}$.

However, it is not hard to improve the precision of this calculation by getting a true normal direction, which is the exact normal to RBF approximation of the surface. The normal can be calculated by

$$\vec{\boldsymbol{n}} = \frac{1}{\sqrt{s_{x_1}^2 + s_{x_2}^2 + s_{x_3}^2}} \left[s_{x_1} \ s_{x_2} \ s_{x_3} \right].$$

After that, we can introduce points $y^{\pm}$ along the true normal on the small distance ε from $\vec{\boldsymbol{\xi_i}}$. Again, we calculate three-dimensional Laplacian of f imposing that $f(y_i^{\pm}) = f(\xi_i)$ we will obtain more precise estimation for the Laplace-Beltrami operator.

The function f and the level-set function $s(x)$ used to represent manifold $\mathcal{M}$ can be approximated by RBF centered at the same points on three layers. As in (3.3), the differentiation matrix takes the form

$$D = \begin{pmatrix} \Delta\Phi_x(x) & \Delta\Phi_{x^+}(x) & \Delta\Phi_{x^-}(x) \\ \Delta\Phi_x(x^+) & \Delta\Phi_{x^+}(x^+) & \Delta\Phi_{x^-}(x^+) \\ \Delta\Phi_x(x^-) & \Delta\Phi_{x^+}(x^-) & \Delta\Phi_{x^-}(x^-) \end{pmatrix} \begin{pmatrix} \Phi_x(x) & \Phi_{x^+}(x) & \Phi_{x^-}(x) \\ \Phi_x(y^+) & \Phi_{x^+}(y^+) & \Phi_{x^-}(y^+) \\ \Phi_x(y^-) & \Phi_{x^+}(y^-) & \Phi_{x^-}(y^-) \end{pmatrix}^{-1}$$

where $\Phi_u(v) = \left\{ \phi\left(\|u_i - v_j\| \right) \right\}_{i,j}$ is $N \times N$ matrix. Denote

$$D = \begin{pmatrix} D_{1,1} & D_{1,2} & D_{1,3} \\ D_{2,1} & D_{2,2} & D_{2,3} \\ D_{3,1} & D_{3,2} & D_{3,3} \end{pmatrix}$$

where $D_{i,j}$ is the $N \times N$ submatrix of the differentiation matrix D for $i, j = 1, 2, 3$.

Denote also $\vec{\boldsymbol{f}} = f(\vec{\boldsymbol{\xi}})$ the vector of values of f estimated on points $\vec{\boldsymbol{\xi}} = \{\xi_i\}_{i=1}^N$ on manifold $\mathcal{M}$ and $\vec{\boldsymbol{g}} = g(\vec{\boldsymbol{\xi}})$ where $g = \mathcal{L}f$. Similarly, $\vec{\boldsymbol{g}}^{\pm}$ stands for the vectors of values of g on points $\vec{\boldsymbol{\xi}}^{\pm}$ on plus and minus layers of $\mathcal{M}$. Then applying D to the vector of values of $\vec{\boldsymbol{f}}$ we will get a vector which contains approximation of Laplace-Beltrami operator of that function on given manifold. These values are defined on all three layers:

$$\begin{pmatrix} D_{1,1} & D_{1,2} & D_{1,3} \\ D_{2,1} & D_{2,2} & D_{2,3} \\ D_{3,1} & D_{3,2} & D_{3,3} \end{pmatrix} \begin{pmatrix} \vec{\boldsymbol{f}} \\ \vec{\boldsymbol{f}} \\ \vec{\boldsymbol{f}} \end{pmatrix} = \begin{pmatrix} \vec{\boldsymbol{g}} \\ \vec{\boldsymbol{g}}^{+} \\ \vec{\boldsymbol{g}}^{-} \end{pmatrix}.$$

Since we are only looking for the values of $\vec{\boldsymbol{g}}$ we can define $D_{\mathcal{M}} = D_{1,1} + D_{1,2} + D_{1,3}$ and get $D_{\mathcal{M}}\vec{\boldsymbol{f}} = \vec{\boldsymbol{g}}$. The eigenvalues of the problem (2.9) without boundary can then be approximated by eigenvalues of the matrix $-D_{\mathcal{M}}$.

3.3.5 High Order

The further improvement of this algorithm can be gained by the direct imposing $\partial_n f = 0$ and $\partial_n^2 f = 0$. Since $\vec{\boldsymbol{n}} \cdot \nabla f = \partial_n f$ and $\partial_n^2 f = (\vec{\boldsymbol{n}} \cdot \nabla)(\vec{\boldsymbol{n}} \cdot \nabla) f$ we require both expressions to be equal zero. That can be achieved by setting $\varepsilon \to 0$ in the algorithm described above.

Denoting $\Phi(x)$ the matrix of RBF functions centered at $3N$ points on the manifold, and/or on its plus and minus layers estimated at point x, we can write:

$$\begin{pmatrix} \Phi(\vec{\boldsymbol{\xi}}) \\ (\vec{\boldsymbol{n}} \cdot \nabla)\Phi(\vec{\boldsymbol{\xi}}) \\ (\vec{\boldsymbol{n}} \cdot \nabla)^2\Phi(\vec{\boldsymbol{\xi}}) \end{pmatrix} \begin{pmatrix} \vec{\boldsymbol{\alpha}} \\ \vec{\boldsymbol{\beta}} \\ \vec{\boldsymbol{\gamma}} \end{pmatrix} = \begin{pmatrix} \vec{\boldsymbol{f}} \\ \vec{\boldsymbol{0}} \\ \vec{\boldsymbol{0}} \end{pmatrix}, \tag{3.4}$$

where $\vec{\boldsymbol{\alpha}}$, $\vec{\boldsymbol{\beta}}$, and $\vec{\boldsymbol{\gamma}}$ are the vectors of coefficients. Thus, the differentiation matrix takes the form

$$D = \begin{pmatrix} \Delta\Phi(\vec{\boldsymbol{\xi}}) \\ \Delta\Phi(\vec{\boldsymbol{\xi}}^+) \\ \Delta\Phi(\vec{\boldsymbol{\xi}}^-) \end{pmatrix} \begin{pmatrix} \Phi(\vec{\boldsymbol{\xi}}) \\ (\vec{\boldsymbol{n}} \cdot \nabla)\Phi(\vec{\boldsymbol{\xi}}) \\ (\vec{\boldsymbol{n}} \cdot \nabla)^2\Phi(\vec{\boldsymbol{\xi}}) \end{pmatrix}^{-1} \tag{3.5}$$

and

$$\begin{pmatrix} D_{1,1} & D_{1,2} & D_{1,3} \\ D_{2,1} & D_{2,2} & D_{2,3} \\ D_{3,1} & D_{3,2} & D_{3,3} \end{pmatrix} \begin{pmatrix} \vec{\boldsymbol{f}} \\ \vec{\boldsymbol{0}} \\ \vec{\boldsymbol{0}} \end{pmatrix} = \begin{pmatrix} \vec{\boldsymbol{g}} \\ \vec{\boldsymbol{g}}^+ \\ \vec{\boldsymbol{g}}^- \end{pmatrix},$$

which reduces to $D_{1,1}\vec{\boldsymbol{f}} = \vec{\boldsymbol{g}}$. Then the eigenvalues of the problem (2.9) without boundary can then be approximated by eigenvalues of the matrix $-D_{1,1}$.

3.4 Projection Methods

Historically preceding Orthogonal Gradient (OGr) methods, the family of Projection methods utilize the idea of applying projection operator $\mathcal{P} = \mathcal{I} - \vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{n}}^{\intercal}$, where $\mathcal{I}$ is the *identity*, and $\vec{\boldsymbol{n}} = \vec{\boldsymbol{n}}\,(\boldsymbol{x})$ is the unit normal vector at a point $\boldsymbol{x} \in \mathcal{M}$. Projection operator allows to express tangential gradient in terms of Cartesian coordinates, assuming knowledge of the unit normal vectors $\vec{\boldsymbol{n}}$ at each point of interest.

This approach was first introduced in [19] and justified analytically by Fuselier and Wright in [21], where authors provide proof of convergence and Sobolev error estimates for interpolation of scattered data on manifold, followed by the error estimates for tangential derivatives in [20]. The short outline of the Projection Operator method follows below. For the purpose of demonstration we consider a case of smooth surface in $\mathbb{R}^3$ only, as generalization for higher-dimensional cases is straightforward.

Denote components of the unit normal vector $\vec{\boldsymbol{n}} = [\, n_x \; n_y \; n_z \,]^{\intercal}$, then

$$\mathcal{P} = \mathcal{I} - \vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{n}}^{\intercal} = \begin{bmatrix} (1 - n_x n_x) & -n_x n_y & -n_x n_z \\ -n_x n_y & (1 - n_y n_y) & -n_y n_z \\ -n_x n_z & -n_y n_z & (1 - n_z n_z) \end{bmatrix}. \tag{3.6}$$

Each column of the matrix (3.6) corresponds to the projections onto x, y, and z unit vectors respectively. Thus, given regular gradient operator $\nabla = [\, \partial_x \; \partial_y \; \partial_z \,]^{\intercal}$ in Cartesian coordinates we construct *tangential gradient* by combining it with Projection operator $\mathcal{P}$ columnwise.

$$\nabla_{\mathcal{M}} = \mathcal{P}\nabla = \begin{bmatrix} (1 - n_x n_x) & -n_x n_y & -n_x n_z \\ -n_x n_y & (1 - n_y n_y) & -n_y n_z \\ -n_x n_z & -n_y n_z & (1 - n_z n_z) \end{bmatrix} \begin{bmatrix} \partial_x \; \partial_x \; \partial_x \\ \partial_y \; \partial_y \; \partial_y \\ \partial_z \; \partial_z \; \partial_z \end{bmatrix},$$

$$\nabla_{\mathcal{M}} = \mathcal{P}\nabla = \begin{bmatrix} (1-n_x n_x)\partial_x - n_x n_y \partial_y - n_x n_z \partial_z \\ -n_x n_y \partial_x + (1-n_y n_y)\partial_y - n_y n_z \partial_z \\ -n_x n_z \partial_x - n_y n_z \partial_y + (1-n_z n_z)\partial_z \end{bmatrix} = \begin{bmatrix} \mathcal{G}^x \\ \mathcal{G}^y \\ \mathcal{G}^z \end{bmatrix} \tag{3.7}$$

where $\mathcal{G}^x$, $\mathcal{G}^y$, and $\mathcal{G}^z$ represent x-component, y-component, and z-component of surface gradient operator $\nabla_{\mathcal{M}}$. Similarly, the *surface divergence* operator acting on a vector field $\vec{\boldsymbol{F}} = [F_1, F_2, F_3]^\intercal$ can be expressed as

$$\nabla_{\mathcal{M}} \cdot \vec{\boldsymbol{F}} = \mathcal{P}\,\mathrm{div}\,\vec{\boldsymbol{F}} = \mathcal{G}^x F_1 + \mathcal{G}^y F_2 + \mathcal{G}^z F_3. \tag{3.8}$$

In order to justify application of (3.7) and (3.8) to solving Laplace-Beltrami eigenvalue problem on manifolds we need to invoke the following *equivalence principles* [10]:

1. Given a function $f : \mathbb{R}^3 \to \mathbb{R}$ that is *constant along the normal* $\vec{\boldsymbol{n}}$ of the manifold $\mathcal{M}$, its tangential gradient is equivalent to the regular gradient restricted to $\mathcal{M}$, i.e. $\nabla_{\mathcal{M}} f = \nabla f$.

2. Given a vector field $\vec{\boldsymbol{F}} : \mathbb{R}^3 \to \mathbb{R}^3$ with each component $[\,F_x\ F_y\ F_z\,]^\intercal$ *constant along the normal* $\vec{\boldsymbol{n}}$, the tangential divergence is equivalent to the regular divergence in embedding Euclidean space restricted to $\mathcal{M}$, i.e. $\nabla_{\mathcal{M}} \cdot \vec{\boldsymbol{F}} = \nabla \cdot \vec{\boldsymbol{F}}$.

The logic behind this construction is straightforward. Given a function $\widetilde{f}$ defined in $\mathbb{R}^3$ and a two-dimensional manifold $\mathcal{M} \in \mathbb{R}^3$, denote f the restriction of $\widetilde{f}$ to $\mathcal{M}$. Recall that one can always expand gradient $\nabla \widetilde{f}$ of a function defined within Euclidean space in any orthonormal system. Denote $\{\vec{\boldsymbol{n}} = \vec{\boldsymbol{n}}(x), \vec{\boldsymbol{t}}_1 = \vec{\boldsymbol{t}}_1(x), \vec{\boldsymbol{t}}_2 = \vec{\boldsymbol{t}}_2(x)\}$ the normal, first tangent, and second tangent unit vectors of $\mathcal{M}$ at point x. The regular gradient at point x can be written as

$$\nabla \widetilde{f}(x) = \widetilde{f}_n \vec{\boldsymbol{n}} + \widetilde{f}_{t_1} \vec{\boldsymbol{t}}_1 + \widetilde{f}_{t_2} \vec{\boldsymbol{t}}_2,$$

and the surface gradient $\nabla_{\mathcal{M}} f(x)$ at a point $x \in \mathcal{M}$ can be represented as the projection of the regular gradient on the plane tangent to $\mathcal{M}$ at point x:

$$\nabla_{\mathcal{M}} f = \mathcal{P}(x) \nabla \widetilde{f} = \widetilde{f}_{t_1} \vec{\boldsymbol{t}}_1 + \widetilde{f}_{t_2} \vec{\boldsymbol{t}}_2,$$

where $\mathcal{P}(x) = (\mathcal{I} - \vec{\boldsymbol{n}} \cdot \vec{\boldsymbol{n}}^{\intercal})$ is projection operator defined as above (3.6). In case $\mathcal{M}$ can be represented as the zero level set of a smooth function $\mathcal{S} : \mathbb{R}^3 \to \mathbb{R}$, the normal vector can be easily computed as $\vec{\boldsymbol{n}} = {}^{\nabla \mathcal{S}}/_{|\nabla \mathcal{S}|}$.

Once the surface gradient is defined, higher order differential operators can be derived in the similar way. For example, the Laplace-Beltrami operator (also known as surface Laplacian) is defined as

$$\Delta_{\mathcal{M}} f = \nabla_{\mathcal{M}} \cdot \nabla_{\mathcal{M}} f = \big(\mathcal{P}\, \nabla\big) \cdot \big(\mathcal{P}\, \nabla\, \widetilde{f}\,\big).$$

Using explicit notation for tangential gradient (3.7) and divergence (3.8) operators, we get the following expression for surface Laplacian in terms of Cartesian coordinates of embedding space.

$$\Delta_{\mathcal{M}} = \mathcal{G}^x \mathcal{G}^x + \mathcal{G}^y \mathcal{G}^y + \mathcal{G}^z \mathcal{G}^z.$$

In order to solve PDEs on a given surface $\mathcal{M}$ via the embedding idea which solves PDEs in a neighborhood of $\mathcal{M}$ in ambient space $\mathbb{R}^3$, one needs to assume that the differentiated functions are defined not only on the surface but in the ambient space as well. Whenever only surface data f is given, the key idea is to extend f off the surface $\mathcal{M}$ such that the gradient of the normal component of its extension $\widetilde{f}$ is zero, i.e.

$$\frac{\partial \widetilde{f}}{\partial n} = \nabla \widetilde{f} \cdot \vec{\boldsymbol{n}} = \nabla \widetilde{f} \cdot \nabla \phi = 0.$$

Sometimes vanishing of higher order derivatives in the normal directions is required too. Take the Laplace-Beltrami operator as an example, $\Delta \widetilde{f}$ equals to $\Delta_{\mathcal{M}} \widetilde{f}$ if $\widetilde{f}_{\boldsymbol{n}}$ and $\widetilde{f}_{\boldsymbol{nn}}$ both vanish everywhere on the surface. As discussed in [3, 26], a wide variety of PDEs on surfaces may be represented as PDEs that depend on an appropriate projection of derivatives in the ambient space.

3.5 Comparison of Presented RBF Methods

In this section we discuss relative strengths and weaknesses of reviewed RBF-based methods and present suggestions for improving some of these methods quantitatively and qualitatively specifically for the framework of Laplace-Beltrami eigenvalue problem.

Note. Even though in introduction we touch on some meshfree methods that are not based on RBF (e.g. level set or closest point) comparing these, as well as some of the more recent finite difference approaches such as [64] with RBF-based techniques lies out of the scope of presented work. In other words, we are going to focus on purely kernel-based techniques, even though in some cases methods like [53] incorporate ideas from multiple approaches mentioned before, making classification somewhat arbitrary.

While RBF or kernel-based methods share many common advantages, among which most frequently mentioned are

- simplicity of implementation,
- geometrical flexibility,
- admission of unstructured set of points,
- possibility of using the same interpolant for reconstructing underlying surface, computing normal components, and interpolating solutions,
- theoretical foundation for unique solvability of interpolation problem (positive-definite kernels only),
- and many others,

they also tend to suffer from common drawbacks:

- Despite of not needing structured mesh, nearly all RBF techniques impose strict requirements for uniformity of underlying mesh.
- Interpolation matrix involved in most of *global* RBF techniques tends to suffer from ill-conditioning, thus imposing strict limitations on accuracy of the solution, admissible by standard precision floating point arithmetic.
- Lack of consistency of certain analytical properties of Laplace-Beltrami operator and its discretized RBF version. Most prominently, very often discretization matrix for Laplace-Beltrami operator constructed via RBF techniques fails to inherit self-adjointness of analytical version of operator perhaps due to round-off errors. This flaw is particularly harmful when computing eigenvalues as the matrix fails to satisfy eigenvalues properties listed in Theorem 1.
- Perhaps the most common and severe limiting factor for using RBF techniques in applications is the lack of concrete guidelines for choosing optimal shape parameter for general domain.

Let us now compare performance and applicability of aforementioned techniques in detail.

3.5.1 Projection Operator: Strengths and Weaknesses

The RBF-based techniques for solving differential equations on manifolds that rely on projection operator are, perhaps, the most commonly used and well-studied family of meshless numerical methods considered in this work. Their main strengths, among others, include:

- Solid formal theoretical justification and error estimates [21].
- No extra RBF centers needed.

On the other hand, the orthogonal projection methods have significant drawbacks:

- Non-symmetric differentiation matrix.

- Common issues with ill-conditioning of distance matrix, especially for schemes using global RBFs as opposed to local RBF-FD ones.

3.5.2 Orthogonal Gradient: Strengths and Weaknesses

In comparison with projection operator based RBF techniques, the RBF-OGr method is more recent and is still undergoing the process of refinement. Despite of having extra computational complexity and somewhat complicated technical preprocessing steps, it seem to possess potential for more flexibility and higher convergence rate than projection-based methods, assuming its main drawbacks can be addressed and mitigated.

The most prominent advantages of RBF-OGr method appear to be the following:

- Intuitive visualizations and simplicity of implementation.
- Spectral accuracy in most published tests.
- Stability: the method is easy to adopt for solving PDEs on manifolds, e.g. for heat equation or nonlinear reaction-diffusion equations such as Brusselator [54].
- Flexibility: a lot of analytical computations can be performed on pre-processing stage, significantly reducing computational complexity of the method and making verification of intermediate steps much easier.

The disadvantages that received the most attention of the authors of the current work include

- Lack of theoretical justification for proposed formulas.
- Cumbersome analytical computations.
- Additional interpolation steps required for the cases when the analytical expression for underlying manifold is not known.

In chapter 4 we will address some of the aforementioned disadvantages and propose techniques aimed to eliminate or alleviate problems associated with the OGr method.

Note. In both cases it might be possible to ensure symmetry of discretized Laplace-Beltrami matrix by utilizing Hermite formulation of eigenvalue problem. However, the tradeoff is that one would have to use more RBF centers, thus increasing computational complexity.

Chapter 4

New RBF Methods for Solving Differential Equation on Manifolds

In this chapter, we propose a new RBF method based on Reilly's formulas (2.12) – (2.13) to reduce computational complexity and enable reformulation of discretized equation for eigenvalues of Laplace-Beltrami operator as *generalized eigenvalue problem*. In this way we try to make the numerical solution capture the analytical properties of Laplace-Beltrami operator, namely the fact that its eigenvalues are supposed to be real.

4.1 Reilly's Formulas

In previous chapters we reviewed two families of RBF-based methods applicable for reconstructing Laplace-Beltrami operator, namely orthogonal gradient (OGr) and orthogonal projection. The core idea of both approaches is to utilize Kansa's original method of reconstructing Laplace operator in Euclidean space and apply it for the case of manifolds by restricting a function to the manifold and forcing it to be constant along normal direction one way or another. From Reilly's formulas (2.12) – (2.13), it is clear that forcing zeros of the first and the second normal derivatives of a function guarantees that its Laplace and Laplace-Beltrami operators coincide on manifold. However, the downsides of this approach is the potential source of numerical error from additional constrains on the system of linear equations, as well as increased computational complexity. Thus, it seems natural to seek analytical expression that would allows to compute Laplace-Beltrami operator directly, without explicitly imposing extra conditions on its argument.

We propose to exploit deep relationship between Laplace-Beltrami operator on the one hand, and manifold information provided by curvature, as well as first and second normal derivative operators combined with Euclidean Laplacian on the other hand. This relationship is captured in the section 2.3 and for cases of planar curves and three-dimensional surfaces essentially boils down to *Reilly's formulas* (2.12) and (2.13) respectively. This way we avoid extra constraints on the interpolation matrix and thus reduce the total number of required RBF nodes to N without sacrificing accuracy.

We start from RBF approximation of a function u defined on a manifold $\mathcal{M}$, i.e. given $u\left(x\right),\ x \in \mathcal{M}$ we write

$$u\left(x\right) \approx \sum_{i=1}^{N} \alpha_i \varphi\left(\|x - \xi_i\|\right) \tag{4.1}$$

where $\varphi : \mathbb{R} \to \mathbb{R}$ is the positive-definite radial function, and ξ_i are the RBF centers lying on or in the vicinity of manifold $\mathcal{M}$, $i = 1, \ldots, N$.

4.1.1 Reconstructing Laplace-Beltrami Operator Using Reilly's Formulas

Given a set of N RBF centers $\vec{\boldsymbol{\xi}} = \{\xi_j\}_{j=1}^{N}$ and the set of N nodes $\vec{\boldsymbol{x}} = \{x_i\}_{i=1}^{N}$ with known values $\vec{\boldsymbol{u}} = \left[u\left(x_1\right), u\left(x_2\right), \ldots, u\left(x_N\right)\right]^{\intercal}$ one can obtain coefficients $\vec{\boldsymbol{\alpha}} = \{\alpha_k\}_{k=1}^{N}$ of function approximation (4.1) by computing the inverse of the distance matrix.

Denote M the distance metric of RBF functions centered at $\vec{\boldsymbol{\xi}}$ and estimated at $\vec{\boldsymbol{x}}$:

$$M := \varphi(\vec{\boldsymbol{x}}, \vec{\boldsymbol{\xi}}) = \begin{bmatrix} \varphi\left(x_1, \xi_1\right), & \dots & \varphi\left(x_1, \xi_N\right) \\ \vdots & \ddots & \vdots \\ \varphi\left(x_N, \xi_1\right), & \dots & \varphi\left(x_N, \xi_N\right) \end{bmatrix},$$

then

$$\vec{\boldsymbol{u}} \approx M \vec{\boldsymbol{\alpha}}$$

and

$$\vec{\boldsymbol{\alpha}} = M^{-1} \vec{\boldsymbol{u}}.$$

Given a linear differential operator $\mathcal{L}$ and denoting $v(x) = \mathcal{L}u(x)$, we get

$$\vec{\boldsymbol{u}} = M\vec{\boldsymbol{\alpha}}, \qquad \vec{\boldsymbol{v}} = M_{\mathcal{L}}\vec{\boldsymbol{\alpha}},$$

where $\{M\}_{i,j} = \varphi\left(\|x_i - \xi_j\|\right)$ and $\{M_{\mathcal{L}}\}_{i,j} = \mathcal{L}\varphi\left(\|x_i - \xi_j\|\right)$. Therefore the differentiation matrix D corresponding to the operator $\mathcal{L}$ can be expressed as:

$$\vec{\boldsymbol{v}} = D\vec{\boldsymbol{u}} \implies D = M_{\mathcal{L}} M^{-1}.$$

For Laplace-Beltrami operator $\mathcal{L} = \Delta_{\mathcal{M}}$ we can use Reilly's formulas (2.12) and (2.13). For the sake of brevity, we rewrite these formulas as a single more general expression

$$\Delta_{\mathcal{M}} = \Delta + K D_{\boldsymbol{n}} - D_{\boldsymbol{nn}} \tag{4.2}$$

where K stands for regular curvature κ in case of planar curve, and $K = 2H$ is the mean curvature for the case of surface embedded in $\mathbb{R}^3$. As before, Δ is the Laplace operator in embedding Euclidean space, and $D_{\boldsymbol{n}} = \vec{\boldsymbol{n}} \cdot \nabla$, $D_{\boldsymbol{nn}} = \vec{\boldsymbol{n}} \cdot H \cdot \vec{\boldsymbol{n}}^{\intercal}$ are the first and second normal derivatives operators respectively. Here ∇ stands for regular gradient, $\vec{\boldsymbol{n}}$ is the unit normal vector, and H is the Hessian matrix. Curvature K for both planar curves and surfaces can be computed using formulas from subsection 2.3.1.

In this way, given radial basis function φ we can use equation (4.2) to write out explicitly the Laplace-Beltrami operator metric matrix entries:

$$\{M_{\mathcal{L}}\}_{i,j} = \mathcal{L}\varphi\left(\|x_i - \xi_j\|\right) = \Delta\varphi_{\xi_j}(x_i) - \left(\nabla \cdot \vec{\boldsymbol{n}}\right) \vec{\boldsymbol{n}} \cdot \nabla\varphi_{\xi_j}(x_i) + \vec{\boldsymbol{n}} \cdot \left(H\varphi_{\xi_j}(x_i)\right) \cdot \vec{\boldsymbol{n}}^{\intercal}$$

In the last expression $\varphi_{\xi_j}(x) := \varphi\left(\|x - \xi_j\|\right)$ is the radial basis function centered at ξ_j, and all derivatives are taken with respect of x variable only.

The eigenvalue problem for Laplace-Beltrami operator can be formulated in the discrete form as follows

$$D\vec{\boldsymbol{u}} = \lambda\vec{\boldsymbol{u}} \quad \Longleftrightarrow \quad M_{\mathcal{L}}M^{-1}\vec{\boldsymbol{u}} = \lambda\vec{\boldsymbol{u}}. \tag{4.3}$$

4.1.2 Generalized Eigenvalue Problem Formulation

Since we are interested in the eigenvalue problem in particular, it would be beneficial to reformulate discretized version of eigenvalue problem (4.3) into *generalized eigenvalue problem*

$$M_{\mathcal{L}}\vec{\boldsymbol{u}} = \lambda M\vec{\boldsymbol{u}}, \tag{4.4}$$

which is particularly easy to implement if the differentiation matrix is computed using Reilly's approach (4.2).

The key advantage of generalized eigenvalue formulation is that it preserves such an important property of Laplace-Beltrami operator as *self-adjointness*. Since discretization of both M and $M_{\mathcal{L}}$ are expected to be real and symmetric, one can use Cholesky factorization or other techniques to reduce or or eliminate the spurious non-zero imaginary parts of eigenvalues of (4.3), like one would expect in analytical setting (2.9) according to the Sturm-Liouville theory.

There is an alternative way to minimize imaginary part of the numerical approximations of

Laplace-Beltrami eigenvalues, namely to use *Hermite formulation* in order to make differentiation matrix D “more symmetric”. It boils down to using both radial basis functions and their Euclidean Laplacian to interpolate the original function u. However, this approach requires to compute the inverse of the metric matrix whose size is twice as large compared to generalized eigenvalue formulation (4.4). Additional drawback of Hermite formulation for discretizing Laplace-Beltrami operator arises in applications. For example, one would have to use implicit numerical schemes if trying to solve time-dependent problem using Hermite interpolation technique.

Chapter 5

Numerical Results

This part of the document is devoted to presenting and comparing numerical results for meshfree methods described in previous sections. First, we investigate convergence for simple planar curves in two-dimensional Euclidean space. Next, we proceed to the case of surfaces in three-dimensional Euclidean space.

Analytical solutions for the Laplace-Beltrami eigenvalue problem

$$-\Delta_{\mathcal{M}} u = \lambda u, \qquad x \in \mathcal{M} \tag{5.1}$$

used to compute errors are provided for all the planar curve and some of the surface test cases. For the surfaces for which the analytical solutions are unknown the convergence rates are estimated by comparing eigenvalues computed on meshes of various sizes.

All computations in the examples below are performed using Gaussian kernel RBF $\varphi_{\boldsymbol{\xi}}(\boldsymbol{x}) = \exp\left(-\varepsilon^2 \left\|\boldsymbol{x} - \boldsymbol{\xi}\right\|^2\right)$ with the shape parameter $\varepsilon = 4$ and the distance between layers $\delta = 0.0005$.

5.1 Planar Curves

In this section we investigate and compare convergence rates of both Low Order (LO) and High Order (HO) Orthogonal Gradient (OGr) [54], Orthogonal Projection (OP) [19–21], and Reilly's formula [55] (4.2) methods by estimating eigenvalues of Laplace-Beltrami operator on several closed planar curves. The name of the method, values of parameters, numbers of scattered points, and brief comments can be found in the description of each figure.

The eigenvalues of Laplace-Beltrami operator defined on any closed smooth planar curve without intersections depend only on the arclength of the curve since such a curve is isometric to a circle [56, 57]. Thus the analytical solutions for each example in this section are

$$\lambda_n = \begin{cases} n^2\pi^2/L^2, & n = 0, 2, 4, \dots \\ (n+1)^2\pi^2/L^2, & n = 1, 3, 5 \dots \end{cases} \tag{5.2}$$

where L is the arclength of the curve. The corresponding eigenfunctions are

$$u_n(l) = \begin{cases} 1/\sqrt{L}, & n = 0, \\ \sqrt{\frac{2}{L}}\cos\left(\frac{\pi n l}{L}\right), & n = 2, 4, \dots \\ \sqrt{\frac{2}{L}}\sin\left(\frac{\pi(n+1)l}{L}\right), & n = 1, 3, 5, \dots \end{cases} \tag{5.3}$$

with the arclength parametrization variable l.

Each presented test case contains error plots for eigenvalues of Laplace-Beltrami operator (5.1) on the given domain, as well as figures with corresponding eigenfunctions computed via Reilly's formulas from section 2.3 on the finest set of scattered points. Additionally, we present plots for condition numbers corresponding to each of the aforementioned computational techniques. In each case the condition numbers of metric matrices based on Reilly's formula are significantly smaller than those of OGr or OP methods using the same number of points distributed on a given manifold.

Tables 5.1 through 5.6 contain information about the first eight analytically computed eigenvalues as well as numerical approximations of the first non-zero eigenvalue computed via various RBF-based methods of interest presented so that readers could observe and compare convergence rates.

Note. In order to be able to use Reilly's formula (4.2) to compute eigenvalues of Laplace-Beltrami operator on orientable manifolds in general and on closed planar curves in particular, it is crucial to keep track of the direction of unit vector and to match the sign of curvature correspondingly. In this thesis, we choose that normal vector at any given node on the curve points towards the region enclosed by the curve.

5.1.1 Unit Circle

The standard and the simplest test case for numerical methods dealing with manifolds is the unit circle given by the equation $x^2 + y^2 = 1$. In order to make the unit normal vector to point towards the center of the circle, we rewrite the equation in the following form

$$\mathcal{M} = \big\{(x, y) \mid \ 1 - x^2 - y^2 = 0\big\}.$$

The exact analytical solutions to the eigenvalue problem (5.1) is given by formulas (5.2) and (5.3) with unit circle arclength $L = 2\pi$.

In order to make use of Reilly's formula (4.2) one needs to know curvature κ and unit normal vector $\vec{\boldsymbol{n}}$ at each point in the set $\boldsymbol{x}$ of nodes on manifold $\mathcal{M}$ with high precision. The formulas for curvature (2.14) and for the unit normal vector of a planar curve are presented in subsection 2.3.1. For the case of circle the curvature is the inverse of the radius, and thus is equal to 1 for the unit circle. The first eight eigenvalues of Laplace-Beltrami operator on unit circle are presented in Table 5.1.

Table 5.1: The first eight "exact" eigenvalues of Laplace-Beltrami operator on the unit circle obtained from formula (5.2).

n	λ_n
0	0
1	1
2	1
3	2
4	2
5	3
6	3
7	4

Table 5.2 contains values of the first non-zero eigenvalue $\lambda_1 = 1$ of the Laplace-Beltrami operator computed numerically via Reilly's, OP, and OGr HO and OGr LO methods using various numbers of points uniformly distributed on the manifold $\mathcal{M}$. One can observe the higher order of precision of Reilly's formula over OGr methods. While comparable in terms of precision, OP method results in spurious eigenmodes with non-zero imaginary part. The numerical results for other eigenvalues demonstrate similar behaviors.

Table 5.2: First non-zero eigenvalue $\lambda_1 = 1$ of Laplace-Beltrami operator on the unit circle.

N	Reilly	Orthogonal Projection	OGr High Order	OGr Low Order
40	1.000000554896507	0.999999970125916	1.00000353525886	0.999873276468444
46	1.000000001021029	0.99999999994697 - 5.86e-15i	1.00000000984682	0.999914700230672
52	1.000000000000952	0.999999999999953	0.999999999609015	0.999892926336349
58	0.999999999999999	0.999999999999337	0.999999999940617	0.999879928635164
64	0.999999999999999	0.99999999999295 - 1.64e-12i	0.999999999213433	0.999837227376132
70	0.999999999999998	1.00000000085605	1.00000000023322	0.999921032223512
76	0.999999999999998	1.00000000080555	1.00000000104087	0.999916751781468

Computations are performed with $N = 40, 46, 52, 58, 64, 70,$ and 76 points uniformly distributed over the unit circle. The absolute errors of the first eight eigenvalues are presented in Figure 5.1, and the corresponding eigenfunctions computed with the finest mesh can be found in Figure 5.2. The condition numbers of corresponding metric matrices are presented in Figure 5.3.

Note. In Figure 5.1 the error plots of Laplace-Beltrami eigenvalues λ_5 and λ_6 computed via Reilly's formula have points missing due to the fact that corresponding errors computed with standard precision are zero.

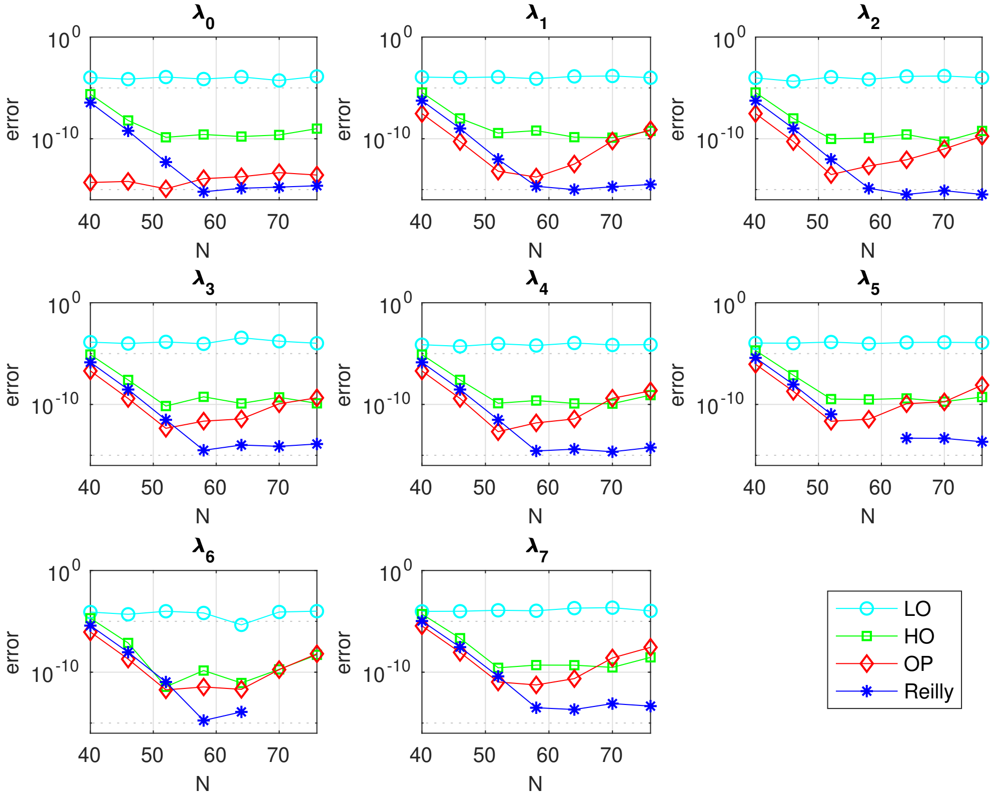


Figure 5.1: Absolute errors for the first 8 eigenvalues of Laplace-Beltrami operator on a unit circle computed via LO, HO, and OP methods with $N = 40, 46, 52, 58, 64, 70,$ and 76 uniformly distributed points compared to computations via Reilly's formula (2.10).

5.1.2 Ellipse

Slightly more complicated than the unit circle, the ellipse test case provides a better insight into comparative performance of presented RBF techniques, due to the less uniform distribution of points, as well as higher variation in distances between them. The former is because

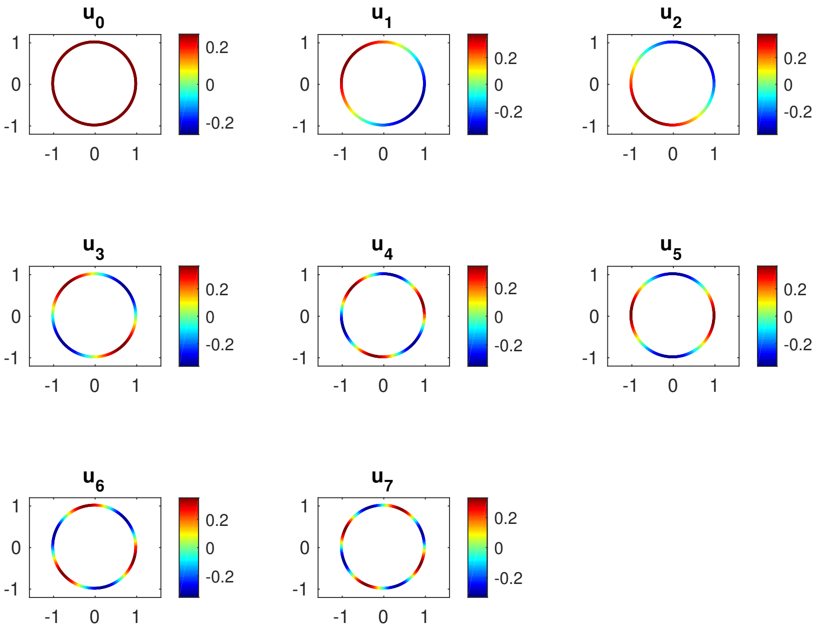


Figure 5.2: First 8 eigenfunctions of Laplace-Beltrami operator computed via Reilly's formula (2.10) with 76 points uniformly distributed over a unit circle.

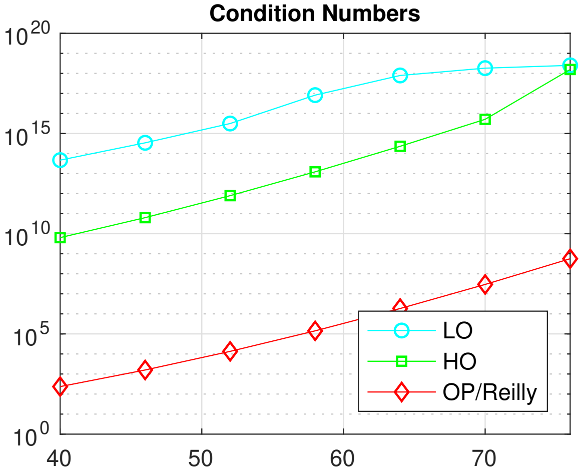


Figure 5.3: Condition numbers of the metric matrices used to compute eigenvalues of Laplace-Beltrami operator on a unit circle computed via LO, HO, and OP/Reilly's methods with $N = 40, 46, 52, 58, 64, 70$, and 76 uniformly distributed points.

of the fact that we use uniform angle for generating RBF nodes via polar parametrization of the equation of an ellipse, and the latter is due to the fact that an ellipse has fewer axises of symmetry than a circle.

The implicit equation of the ellipse is

$$\mathcal{M} = \left\{(x, y) \mid 1 - \frac{x^2}{a^2} - \frac{y^2}{b^2} = 0\right\}, \tag{5.4}$$

where the sign of the left-hand side of the equation is chosen in a way so that the unit normal vector would point towards the area enclosed by the curve. For our numerical experiment we choose $a = 2$ and $b = 1$, and the arclength

$$L = \int_0^{2\pi} \sqrt{1 - \left(1 - b^2/a^2\right) \sin^2\theta}\, d\theta \approx 9.688448220547675$$

was computed numerically using an adaptive quadrature rule with the standard double-precision arithmetic in MATLAB by using the built-in function “integral”. The exact analytical eigenvalues used to compute absolute error for each method are given by formula (5.2) and are listed in Table 5.3.

Table 5.3: The first eight “exact” eigenvalues of Laplace-Beltrami operator on the ellipse (5.4) computed via (5.2) with semiaxes $a = 2$ and $b = 1$ and the arclength $L \approx 9.688448220547675$.

n	λ_n
0	0
1	0.42058259050316
2	0.42058259050316
3	1.68233036201264
4	1.68233036201264
5	3.78524331452844
6	3.78524331452844
7	6.72932144805055

The curvature of the ellipse defined implicitly by (5.4) is computed in Example 2.9, and analytical expressions for the unit normal vector and for Reilly’s formula for the Laplace-Beltrami operator can be found in Example 2.10.

Table 5.4 contains values of the first non-zero eigenvalue $\lambda_1 = 0.42058259050316$ of the Laplace-Beltrami operator computed numerically via Reilly’s, OP, and OGr LO and OGr

HO methods with $N = 60, 68, 76, 84, 92, 100, 108$, and 116 points on the ellipse. All presented methods except for the one based on the Reilly's formula have spurious eigenmodes with non-zero imaginary part. The numerical results for the other eigenvalues demonstrate similar behavior.

Table 5.4: First non-zero eigenvalue $\lambda_1 = 0.42058259050316$ of Laplace-Beltrami operator on the ellipse with semiaxis $a = 2$ and $b = 1$ computed via different RBF-based methods.

N	Reilly	Orthogonal Projection
60	0.420699812837476	0.419634761517436 - 0.0017i
68	0.420602197826696	0.426695127324411
76	0.420582899854937	0.425001625017688 - 0.0006i
84	0.420582591070875	0.420279501051672 - 0.0016i
92	0.420582590507273	0.42067846333157
100	0.420582590503181	0.405359318499848
108	0.42058259050316	0.420964174317887
116	0.420582590503161	0.420323380534003
	OGr Low Order	OGr High Order
60	0.421379654189749	0.420723329297999
68	0.420518882910492	0.420607864505688
76	0.420490249675412	0.420583032097795
84	0.420523164310360	0.42058259170771
92	0.420211828249658	0.420583056827321
100	0.420485529569251 - 5.5128e-06i	0.420580322454658
108	0.420560876032518	0.420582588584226 - 5.1347e-08i
116	0.420041242307339	0.420582746651726

Computations are performed on using $N = 60, 68, 76, 84, 92, 100, 108$, and 116 points distributed uniformly in angle over the ellipse. The absolute errors of the first eight eigenvalues are presented in Figure 5.4, and the corresponding eigenfunction computed with the finest mesh can be found in Figure 5.5. The condition numbers of corresponding metric matrices are presented in Figure 5.6.

Looking at Figure 5.4 one can easily see the superior convergence of Reilly's formula method over the rest when non-uniform scattered points are used.

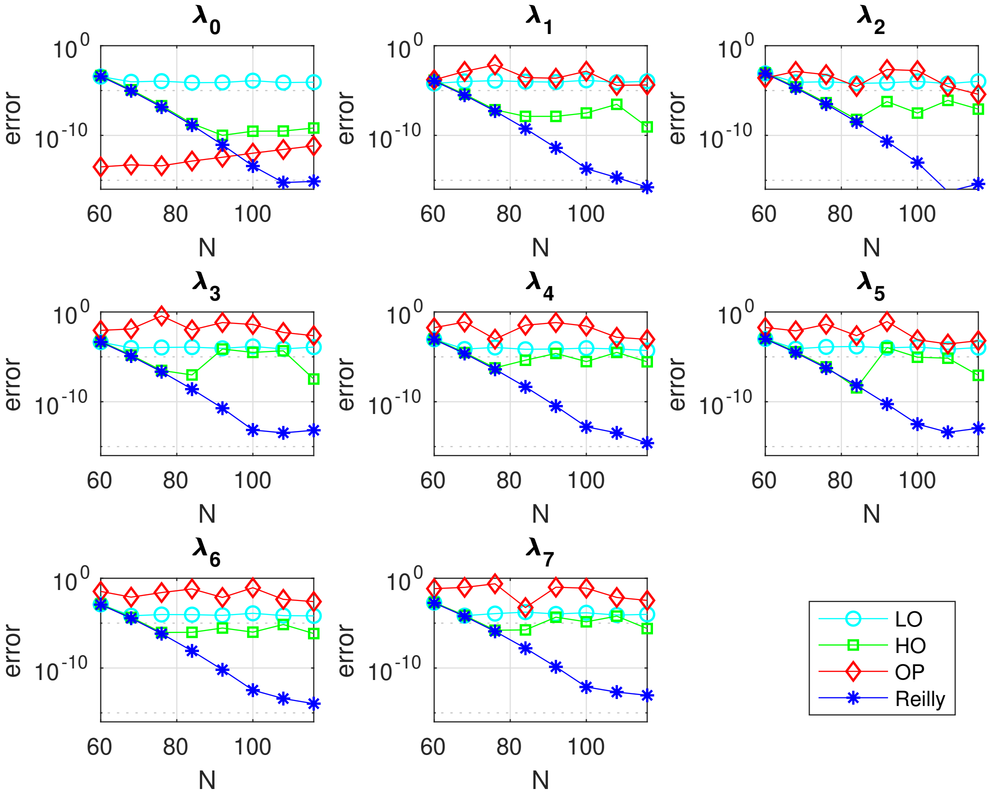


Figure 5.4: Absolute errors for the first 8 eigenvalues of Laplace-Beltrami operator on ellipse with semiaxes $a = 2,\ b = 1$ computed via LO, HO, and OP compared to computations via Reilly's formula (2.10). Computations are performed for $N = 60, 68, 76, 84, 92, 100, 108$, and 116 points.

5.1.3 Hippopede

Next we present the hippopede shape. In polar coordinates the equation of the hippopede is

$$r^2 = 4b\left(a - b\sin^2\theta\right), \quad a > b. \tag{5.5}$$

In Cartesian coordinates the level-set representation of this curve is

$$\mathcal{M} = \left\{(x, y) \mid \ 4b^2x^2 - \left(x^2 + y^2\right)^2 - 4b\,(b - a)\left(x^2 + y^2\right) = 0\right\}, \tag{5.6}$$

which is a curve of degree 4.

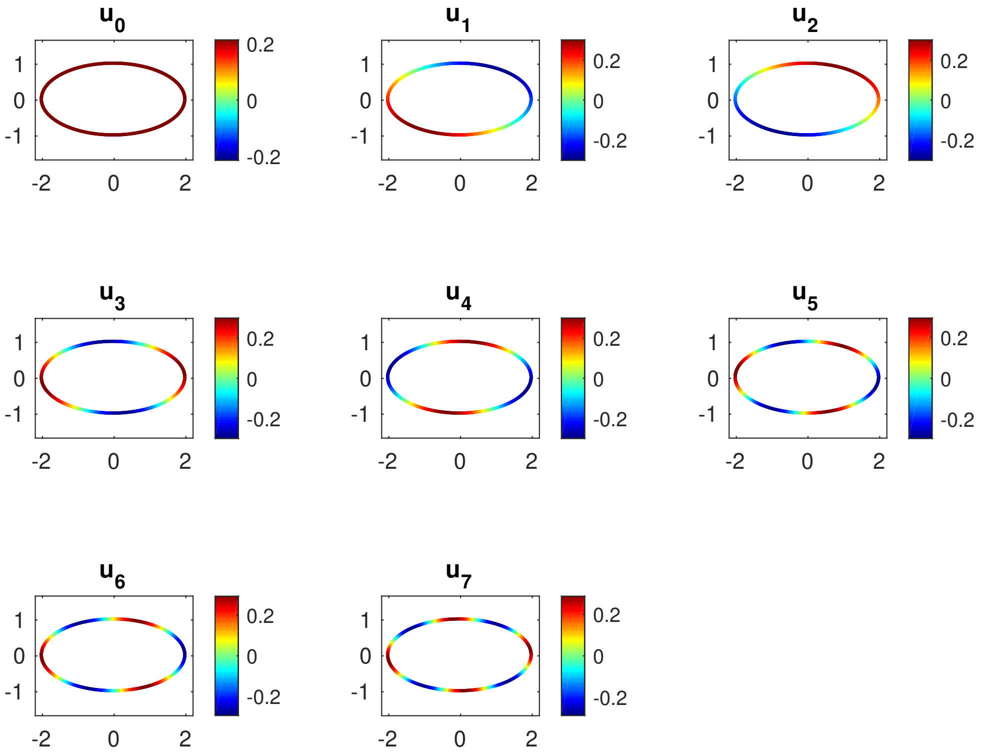


Figure 5.5: Eigenfunctions corresponding to the first 8 eigenvalues of Laplace-Beltrami operator on ellipse with semiaxes $a = 2,\ b = 1$ computed via Reilly's formula (2.10) with $N = 116$ points.

Note. In Equation (5.6) the point $(0, 0)$ should be excluded from the set of solutions because according to (5.5) the radius r cannot equal 0 because $a > b$.

For the present numerical example we used parameters $a = 1.33$ and $b = 1$. The arclength of corresponding shape is $L \approx 12.331833535762643$ was computed similarly to the case of the ellipse. The exact analytical eigenvalues used to compute absolute error for each method are given by (5.2) and are listed in Table 5.5. The curvature of the hippopede defined implicitly by (5.6) is computed using formula (2.14).

Table 5.6 contains values of the first non-zero eigenvalue $\lambda_1 = 0.259599845336741$ of the

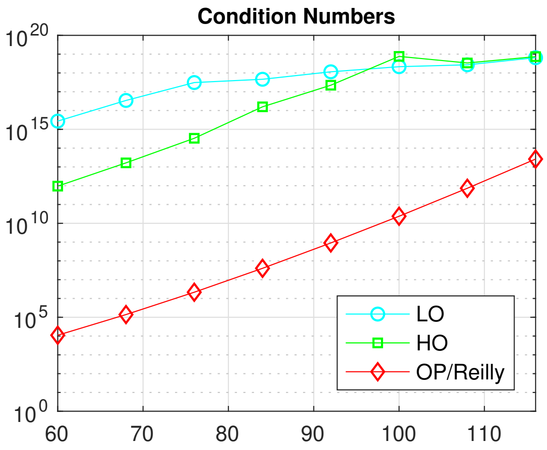


Figure 5.6: Condition numbers of metric matrix used to compute eigenvalues of Laplace-Beltrami operator on the ellipse with semiaxes $a = 2,\ b = 1$ computed via LO, HO, and OP/Reilly's methods with $N = 116$ points.

Table 5.5: The first eight "exact" eigenvalues of Laplace-Beltrami operator on hippopede (5.6) computed via (5.2) with parameters $a = 1.33$ and $b = 1$ and the arclength $L \approx 12.331833535762643$.

n	λ_n
0	0
1	0.259599845336741
2	0.259599845336741
3	1.038399381346965
4	1.038399381346965
5	2.336398608030672
6	2.336398608030672
7	4.153597525387862

Laplace-Beltrami operator computed numerically via Reilly's, OP, and OGr LO and OGr HO methods with N = 48, 66, 86, 98, 110, 136, 164 and 191 points on hippopede. The numerical results for the other eigenvalues demonstrate similar behavior.

Computations are performed on using N = 48, 66, 86, 98, 110, 136, 164 and 191 points distributed over the hippopede. The mesh was generated using the DistMesh package for MATLAB [52].

The absolute errors of the first eight eigenvalues are presented in Figure 5.7, and the corresponding eigenfunction computed with the finest mesh can be found in Figure 5.8. The

Table 5.6: First non-zero eigenvalue $\lambda_1 = 0.259599845336741$ of Laplace-Beltrami operator on hippopede with parameters $a = 1.33$ and $b = 1$ computed via different RBF methods.

N	Reilly	Orthogonal Projection
48	0.409040548464180	0.232360278194370
66	0.269774625610826	0.262954163465159 - 0.00156i
86	0.260260685545799	0.259914760907661 - 2.8429e-05i
98	0.259686046282354	0.259739532030377
110	0.259623517564784 - 6.7536e-06i	0.259621608010336
136	0.259599921207862	0.259597560812404 - 1.5345e-06i
164	0.259599841580032	0.259594460252486
191	0.259599847530652	0.259530317456710
	OGr Low Order	OGr High Order
48	0.475645553448326	0.475720990663987
66	0.270777294203673	0.270873965176631
86	0.262308684075893	0.260337120879924
98	0.259732405961322 - 3.4343e-05i	0.259820153323240
110	0.259539012756428 - 4.5661e-05i	0.259632006301648 - 5.2014e-05i
136	0.258798300281343	0.259590964071525
164	0.258615684392707	0.259599967993140 - 1.9084e-06i
191	0.260392939596244 - 0.0034025i	0.259031158460990

condition numbers of corresponding metric matrices are presented in Figure 5.9. Looking at Figure 5.7 one can observe better convergence of Reilly's formula method compared to the rest when relatively uniform points are used on the non-trivial curve.

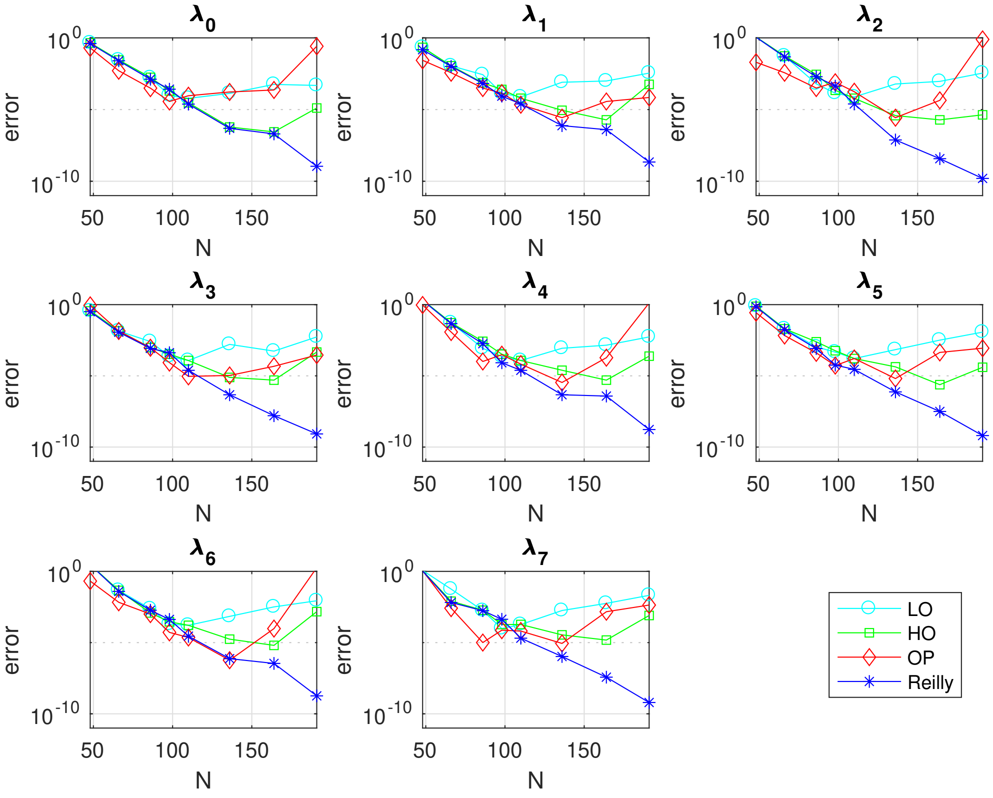


Figure 5.7: Absolute errors for the first 8 eigenvalues of Laplace-Beltrami operator on hippopede with parameters $a = 1.33$, $b = 1$ computed with $N = 48$, 66, 86, 98, 110, 136, 164 and 191 points with LO, HO, and OP methods compared to computations via Reilly's formula (2.10).

5.2 Surfaces

After describing numerical experiments on closed planar curves, we proceed to the closed two-dimensional surfaces embedded in three-dimensional Euclidean space. As before, the description of each figure below includes the name of the method, values of parameters, numbers of scattered points, and brief comments.

Just like in the case of planar curves, we present numerical study for eigenvalues of Laplace-Beltrami operator on multiple surfaces, involving OGr LO and OGr HO [54], OP [19–21], and Reilly's formula [55] (4.2) methods. The error plots for computed eigenmodes of Laplace-

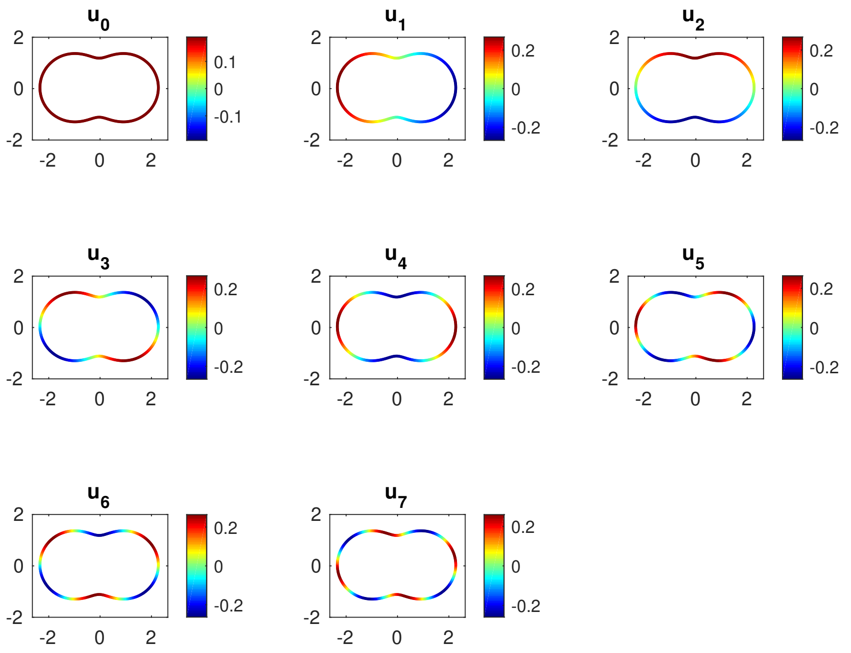


Figure 5.8: Eigenfunctions corresponding to the first 8 eigenvalues of Laplace-Beltrami operator on hippopede with parameters $a = 1.33,\ b = 1$ computed with $N = 191$ nodes via Reilly's formula (2.10).

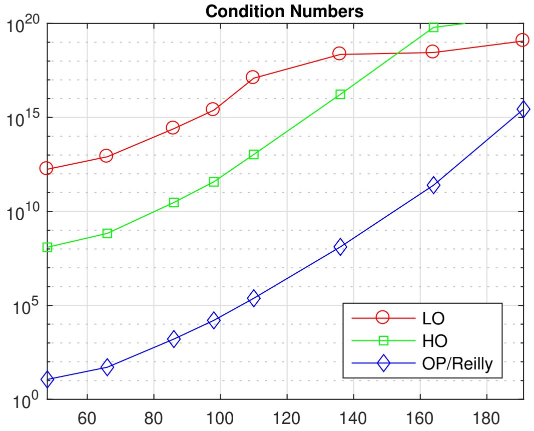


Figure 5.9: Condition numbers of metric matrices used to compute eigenvalues of Laplace-Beltrami operator on hippopede with parameters $a = 1.33,\ b = 1$ computed via LO, HO, and OP/Reilly's methods on $N = 48,\ 66,\ 86,\ 98,\ 110,\ 136,\ 164$ and 191 points.

Beltrami operator (5.1) on a given domain via several RBF-based methods, including the one based on the Reilly's formulas from section 2.3, are presented in the figures below.

Since for the most of the surfaces in this section the exact analytical solution to Laplace-Beltrami eigenvalue problem (5.1) is unavailable, we estimate the convergence of each method by observing the saturation of computed numerical values as we increase the number of RBF nodes lying on the manifold. The corresponding numerical estimates of the first non-zero eigenvalue computed via different RBF methods are presented on the tables 5.8. The exception is the case of unit sphere for which the analytical solution is readily available.

In order to obtain points distributed nearly uniformly on a given manifold we used MATLAB package DistMesh [52] developed by Persson and Strang which is available online for free. The package uses implicit level-set representation of a surface and base resolution parameter to generate triangulation of the surface. For nearly every case tested the accuracy of generated points on manifold is within 10^{-5} to 10^{-6}, so we used "projection" procedure to increase the accuracy whenever the parametric representation of the surface is available alongside the implicit equation. The accuracy was computed by plugging the surface triangulation coordinates into the implicit equation representing zero level-set of the surface, and after "parametric adjustment" procedure was improved up to 10^{-14} to 10^{-15} absolute error.

For every test case below we provide figures with eigenfunctions of Laplace-Beltrami operator (5.1) computed using the finest set of scattered points, as well as figures with the condition numbers of metric matrices plotted against the number of points on manifold. We also provide tables with computed values in order to demonstrate saturation of eigenvalues. The error plot for eigenvalues is only present for the case of unit sphere, as the rest of examples do not have explicit analytical formula for eigenvalues that we could use to compute the absolute error.

The analytical expressions for the curvature and for the unit normal vector of each surface were obtained according to formulas from subsection 2.3.1. In particular, the curvature of a surface given by an equation $z = f(x, y)$ is computed explicitly in Example 2.11. Analytical expressions for the Laplace-Beltrami operator are derived in terms of metric tensor in Examples 2.5 and 2.6.

5.2.1 Unit Sphere

Similar to the numerical study of planar curves, we start overview of performance of RBF methods on the unit sphere given by the equation $x^2 + y^2 + z^2 = 1$, which is the simplest closed surface in three dimensions. In order to make the unit normal vector point towards the enclosed area of Euclidean space $\mathbb{R}^3$, we rewrite the equation in the following form

$$\mathcal{M} = \left\{ (x, y, z) \mid 1 - x^2 - y^2 - z^2 = 0 \right\}.$$

The analytical solution of Laplace-Beltrami eigenproblem on the unit sphere consists of the set of Laplace's spherical harmonics $Y_l^m(\theta, \phi)$ of degree $l \in \mathbb{N}$ and of the order $m \in \mathbb{N}$ such that $-l \leq m \leq l$ with corresponding eigenvalue $\lambda = l(l + 1)$, the first fifteen of which are listed in the Table 5.7 below.

Table 5.7: The first fifteen exact eigenvalues of Laplace-Beltrami operator on the unit sphere.

n	λ_n	n	λ_n
0	0	8	6
1	2	9	12
2	2	10	12
3	2	11	12
4	6	12	12
5	6	13	12
6	6	14	12
7	6		

Just like in case of planar curves, using Reilly's formula (4.2) for Laplace-Beltrami operator on surfaces requires mean curvature κ_m and the unit normal vector $\vec{\boldsymbol{n}}$ at each point in the set $\boldsymbol{x}$ of nodes on manifold $\mathcal{M}$ with high precision. Corresponding formulas for κ_m (2.15) and for the unit normal vector are presented in subsection 2.3.1. In case of the unit sphere, the mean curvature is 1.

Table 5.8 contains values of the first non-zero eigenvalue $\lambda = 2$ of the Laplace-Beltrami operator on the unit sphere computed numerically via Reilly's, OP, OGr LO, and OGr HO methods using various numbers of points. One can observe the higher order of precision of Reilly's formula over Orthogonal Gradient methods. The numerical results for other eigenvalues demonstrate similar behavior.

Table 5.8: First non-zero eigenvalue $\lambda = 2$ of Laplace-Beltrami operator on the unit sphere.

N	Reilly	Orthogonal Projection
472	2.000044079955452	1.999998805582061
608	2.000004032705993	1.999999979459302
832	2.000000165378249	2.000000011128450
1206	2.000000001268887	1.999999999909044 - 1.81e-09i
1828	2.000000000000464	1.999995630624536
	OGr Low Order	OGr High Order
472	2.000013409523629	2.000113213025058
608	1.999933834391497	2.000011231385069
832	1.999919373838853	2.000000570771269
1206	1.999669862492228	2.000000006651276
1828	1.997411185018087	1.999998352503409

The graphs in Figure 5.10 depict absolute errors of eigenvalues of Laplace-Beltrami operator computed with OGr LO, OGr HO, and OP methods compared to those computed using Reilly's formula (2.13). The superior convergence of the last method can be partially attributed to the lower condition numbers of corresponding metric matrices, which are presented in Figure 5.11.

Figure 5.12 contains images of eigenfunctions Y_l^m corresponding to the first fifteen eigenvalues of Laplace-Beltrami operator on the unit sphere computed using Reilly's formula (4.2) on the finest set of nodes that includes $N = 1828$ points.

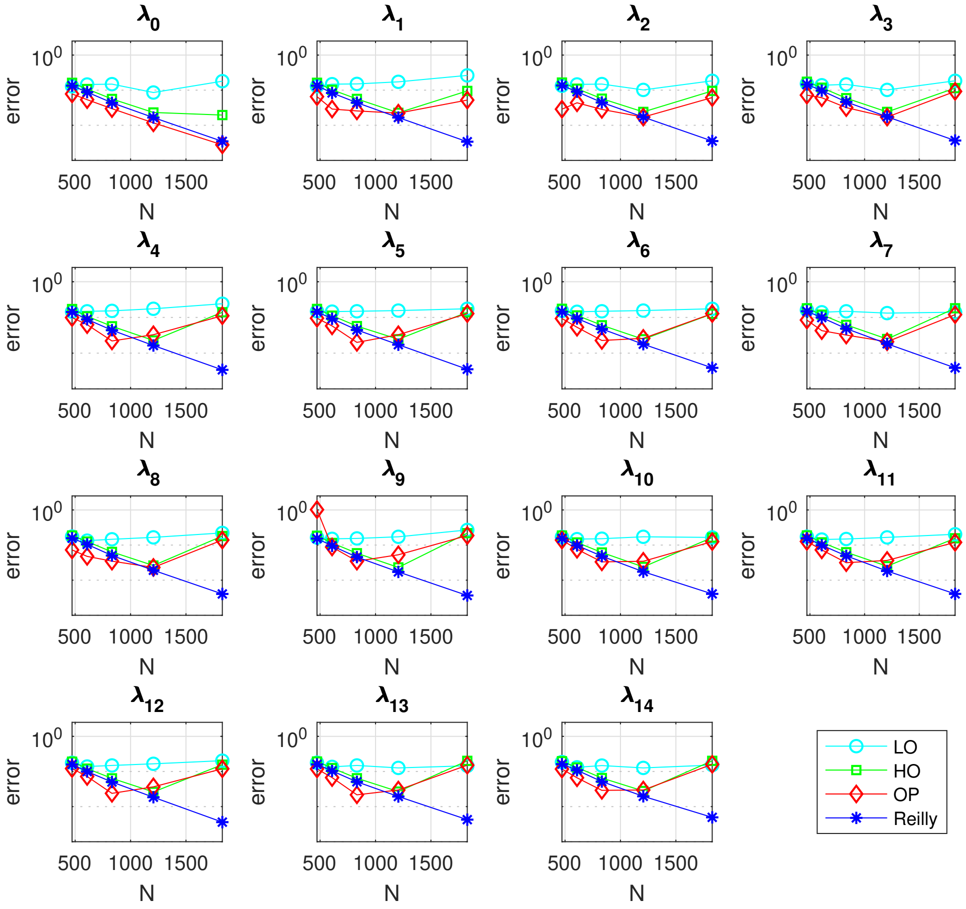


Figure 5.10: Absolute errors for the first 15 eigenvalues of Laplace-Beltrami operator on the unit sphere computed via LO, HO, and OP with N = 472, 608, 832, 1206, and 1828 scattered points compared to computations via Reilly's formula (2.11).

5.2.2 Torus

In order to compare performance of the method of computing eigenvalues of Laplace-Beltrami operator on two-dimensional surfaces proposed in section 4.1, we consider torus defined by

$$\left(x^2 + y^2 + z^2 + R^2 - r^2\right)^2 - 4R^2\left(x^2 + y^2\right) = 0, \tag{5.7}$$

where R is the radius from the center of the torus to the center of the tube, and r is the

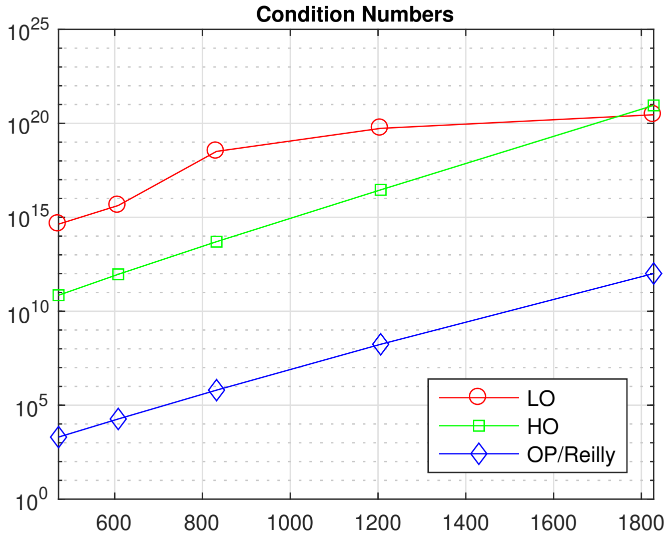


Figure 5.11: Condition numbers of metric matrix used to compute eigenvalues of Laplace-Beltrami operator on the unit sphere computed via LO, HO, and OP/Reilly's methods with $N = 472$, 608, 832, 1206, and 1828 scattered points.

radius of the tube. In parametric form, the torus is given by the following system

$$\begin{cases} x(\phi,\theta) = (R + r\cos\theta)\cos\phi, \\ y(\phi,\theta) = (R + r\cos\theta)\sin\phi, \\ z(\phi,\theta) = r\cos\theta. \end{cases}$$

In this example we choose parameters $R = {}^{4}\!/_{3}$ and $r = 1$ following the numerical experiment presented by Glowinski and Sorensen in [22]. Since there are no closed-form analytical formulas for eigenvalues of Laplace-Beltrami operator on torus known to authors, we are unable to compute the exact errors. However, we still can get some sense of the accuracy of each method by comparing obtained eigenvalues to the ones in [22] and matching multiplicity of the eigenvalues of different order.

In order to be able to use Reilly's formula (4.2) we compute mean curvature κ_m (2.15) and unit normal vector $\vec{\boldsymbol{n}}$ at each point in the set of nodes $\vec{\boldsymbol{x}}$ lying on the torus using formulas from the section 2.3.

Table 5.9 contains eigenvalues of the Laplace-Beltrami operator on torus (5.7) with the major

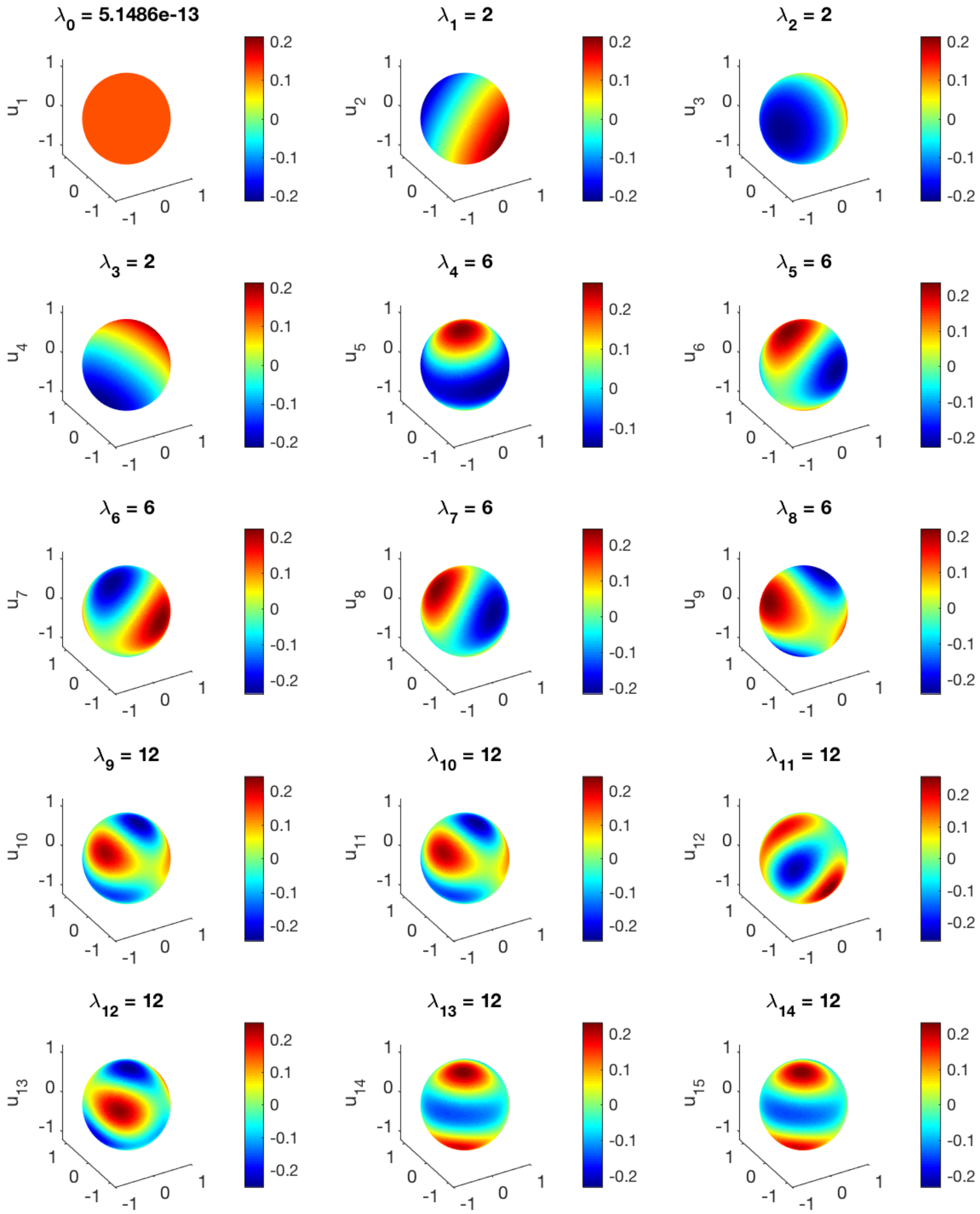


Figure 5.12: Laplace-Beltrami eigenfunctions Y_l^m corresponding to the first 15 eigenvalues on the unit sphere computed via Reilly's formula (2.10) on 1828 nodes.

radius $R = {}^4\!/_3$ and minor radius $r = 1$ computed using Reilly's formula (4.2) for different numbers of points scattered on manifold. Corresponding eigenfunctions can be found in

Figure 5.13.

Table 5.9: The first fifteen eigenvalues of Laplace-Beltrami operator on torus (5.7) computed via (5.2) with major and minor radii $R = {}^4/3$ and $r = 1$ respectively.

N	956	1888	2668	3472
λ_0	0.036751534131179	0.000075342536481	0.000002441476837	0.000000121455757
λ_1	0.480547661027475	0.444102386920198	0.444035418941456	0.444033269948001
λ_2	0.481543311105374	0.444108272635499	0.444035489723862	0.444033285546754
λ_3	0.977234801118360	0.937121978746625	0.937048566169670	0.937045907568466
λ_4	1.310776528163255	1.271961627244771	1.271889226885949	1.271887227538113
λ_5	1.313582129728811	1.271964340228779	1.271889744932636	1.271887256562822
λ_6	1.407158921802641	1.353701091989552	1.353584124322381	1.353580728982300
λ_7	1.702066304427230	1.659455000657586	1.659386876671300	1.659384164508465
λ_8	1.703252012672184	1.659474768093446	1.659386965942933	1.659384193013252
λ_9	2.512593134113784	2.468044929802211	2.467971731856655	2.467969422892098
λ_{10}	2.513963329440845	2.468056785052581	2.467971947556423	2.467969462816265
λ_{11}	3.111015997044699	3.063692803786345	3.063607599840778	3.063604933137443
λ_{12}	3.115971935320059	3.063697451790484	3.063608231159288	3.063604955497150
λ_{13}	3.303336016413147	3.251615879516153	3.251530205647079	3.251527165201466
λ_{14}	3.306543847241047	3.251633500722744	3.251530273501585	3.251527175880005

Table 5.10 contains values of the first non-zero eigenvalue computed numerically via Reilly's, OP, OGr LO, and OGr HO methods using various numbers of points. The numerical results for other eigenvalues demonstrate similar behaviors. The condition numbers of metric matrices associated with different numbers of nodes are plotted in Figure 5.14.

Table 5.10: First non-zero eigenvalue of Laplace-Beltrami operator on the torus (5.7) with major and minor radii $R = {}^4/3$ and $r = 1$ respectively.

N	Reilly	Orthogonal Projection
956	0.480547661027474	0.443537322498211
1888	0.444102386920198	0.444035874910732
2668	0.444035418941456	0.444033284830513
3472	0.444033269948001	0.444033163244554
	OGr Low Order	OGr High Order
956	0.511197330555129	0.511348432193897
1888	0.444042590377088	0.444152957874043
2668	0.443952808901597	0.444037438724576
3472	0.443926606103339	0.444033442510456

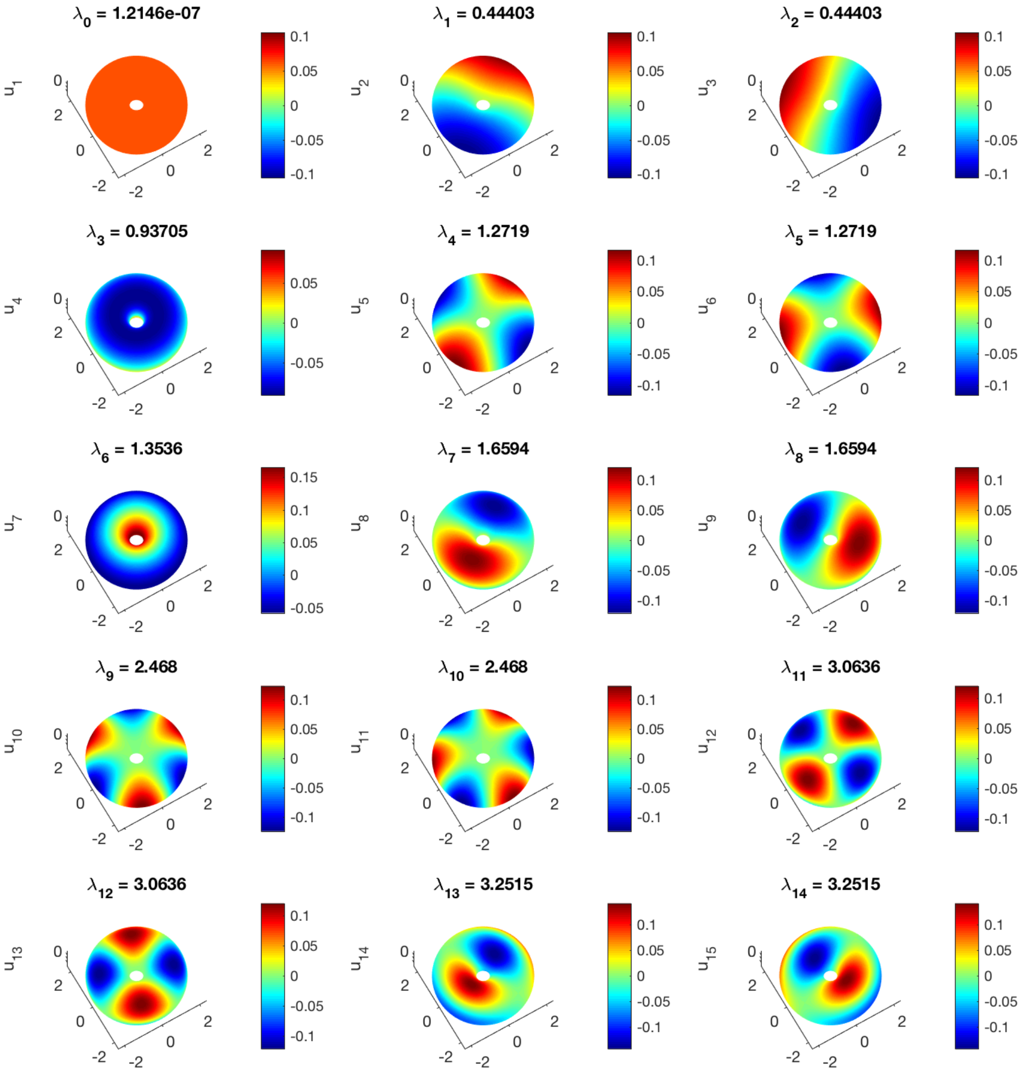


Figure 5.13: Laplace-Beltrami eigenfunctions corresponding to the first 15 eigenvalues on torus with $R = 4/3$ and $r = 1$ computed using Reilly's formula (2.10) on 3472 nodes.

5.2.3 Ellipsoid

Consider a two-dimensional ellipsoid given by the equation

$$\mathcal{M} = \left\{ (x, y, z) \mid 1 - \frac{x^2}{a^2} - \frac{y^2}{b^2} - \frac{z^2}{c^2} = 0 \right\}. \tag{5.8}$$

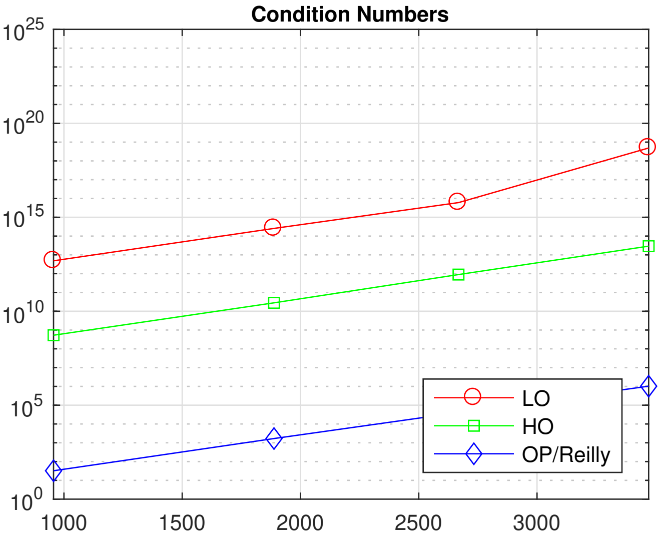


Figure 5.14: Condition numbers of metric matrices used to compute eigenvalues of Laplace-Beltrami operator on torus (5.7) with major radius $R = 4/3$ and minor radius $r = 1$ computed with LO, HO, and OP/Reilly's methods.

where parameters are chosen as $a = 2$, $b = 1$, and $c = 1.5$. Similarly to the case of unit sphere presented in subsection 5.2.1, the choice of the sign of the left-hand side of the equation (5.8) is motivated by the direction of the unit normal vector, which we want to point "inwards" i.e. towards the region of Euclidean space enclosed by the surface.

Table 5.11 contains eigenvalues of the Laplace-Beltrami operator on the ellipsoid (5.8) computed using Reilly's formula (4.2) for different numbers of points scattered on manifold. Corresponding eigenfunctions can be found in Figure 5.15.

The numerical approximations of the first non-zero eigenvalue computed via Reilly's, OP, OGr LO, and OGr HO methods using various numbers of points can be found in Table 5.12. The numerical results for other eigenvalues demonstrate similar behaviors. The condition numbers of metric matrices associated with different numbers of nodes are plotted in Figure 5.16.

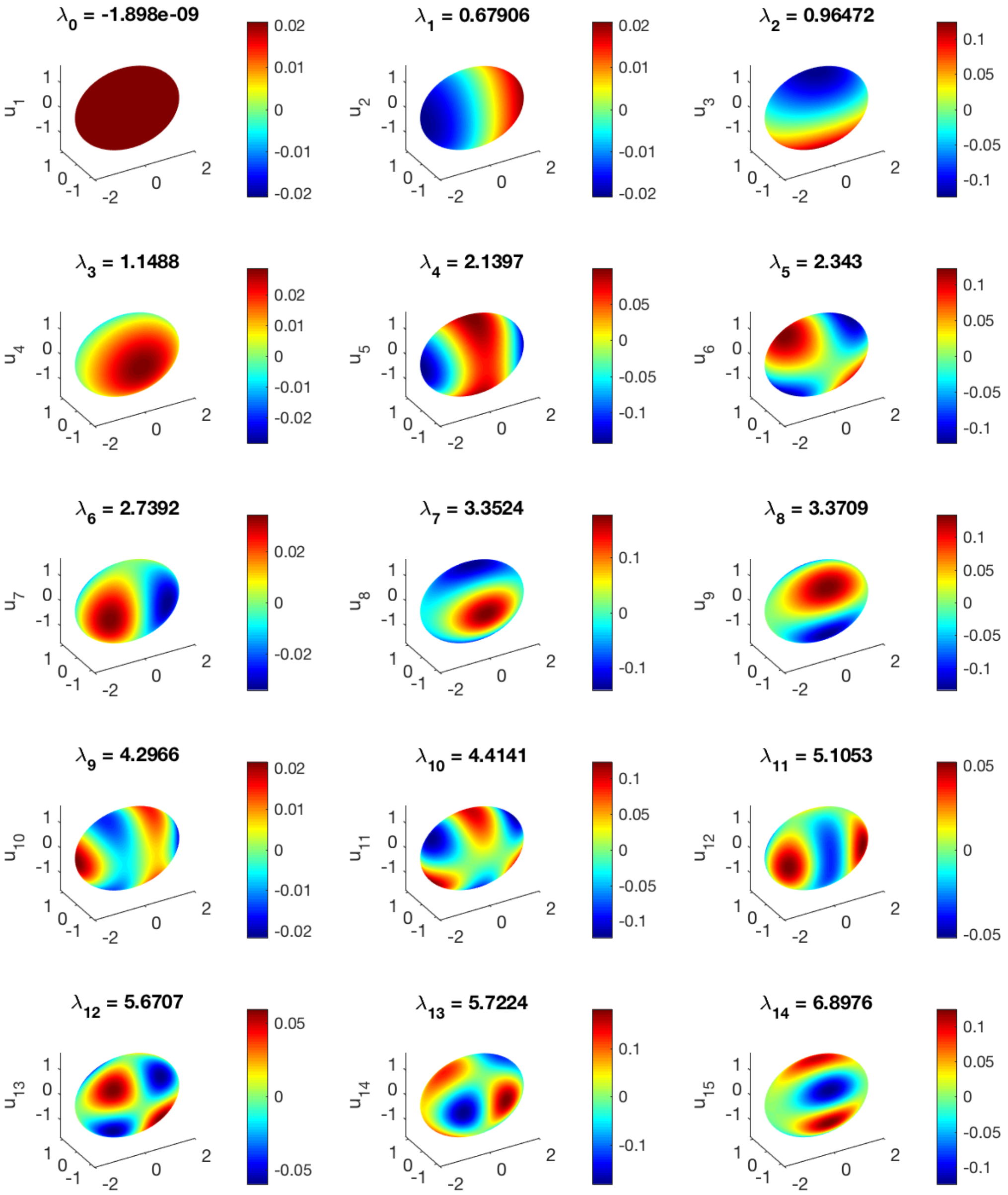


Figure 5.15: Eigenfunctions corresponding to the first 15 eigenvalues of Laplace-Beltrami operator on ellipsoid with semiaxis $a = 2$, $b = 1$, and $c = 1.5$ computed via Reilly's formula (2.10).

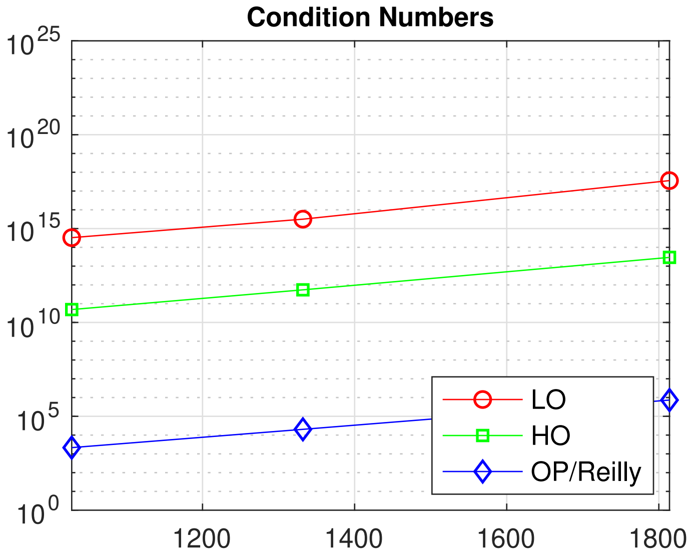


Figure 5.16: Condition numbers of metric matrices used to compute eigenvalues of Laplace-Beltrami operator on ellipsoid (5.8) with principle semiaxis $a = 2$, $b = 1$, and $c = 1.5$ computed via LO, HO, and OP/Reilly's methods.

5.2.4 Goursat Surface

The Goursat surface is given implicitly by the equation

$$\mathcal{M} = \Big\{ (x, y, z) \mid \; x^4 + y^4 + z^4 + a\big(x^2 + y^2 + z^2\big)^2 + b\big(x^2 + y^2 + z^2\big) + c = 0 \Big\}.$$

For our numerical experiment we choose parameters $a = 0$, $b = -1$, and $c = {}^9/_{20}$, thus obtaining a smooth shape of genus 6 described implicitly by the equation

$$\mathcal{M} = \Big\{ (x, y, z) \mid \; x^4 + y^4 + z^4 - x^2 - y^2 - z^2 + {}^9/_{20} = 0 \Big\}. \tag{5.9}$$

Figure 5.17 presents plots of eigenvalues of the Laplace-Beltrami operator, computed with Reilly's formula (4.2) with 3192 nodes distributed over the surface with the accuracy $\sim 10^{-5}$ by the DistMesh package [52] for MATLAB. The numerical approximation for eigenvalues themselves computed with different numbers of scattered nodes can be found in Table 5.13.

Next, Table 5.14 compares numerical approximations of the first non-zero eigenvalue of

Laplace-Beltrami operator on Goursat surface (5.9) computed with Reilly's, OP, OGr LO, and OGr HO methods using various numbers of points. The results for other eigenvalues demonstrate similar behaviors. The condition numbers of metric matrices associated with different numbers of nodes are plotted in Figure 5.18.

5.2.5 Pilz Surface

The Pilz surface in this section is given implicitly by the equation

$$\mathcal{M} = \left\{ (x, y, z) \mid \left(\left(x^2 + y^2 - 1\right)^2 + (z+1)^2 \right) \left(\left(x^2 + (z+0.7)^2 - 1\right)^2 + y^2 \right) - 0.1 \right\}, \tag{5.10}$$

and is used here as an example of a surface of genus 4. The eigenfunctions of Laplace-Beltrami operator on Pilz surface approximated using Reilly's formula (4.2) and $N = 3916$ scattered nodes are plotted in Figure 5.20. The nodes are distributed over the surface with the accuracy $\sim 10^{-5}$ by the DistMesh package [52] for MATLAB. The approximations of corresponding eigenvalues can be found in Table 5.15.

Table 5.16 compares numerical approximations of the first non-zero eigenvalue of Laplace-Beltrami operator on Pilz surface (5.10) computed with Reilly's, OP, OGr LO, and OGr HO methods using various numbers of points. The results for other eigenvalues demonstrate similar behaviors. The condition numbers of metric matrices associated with different numbers of nodes are plotted in Figure 5.19.

Table 5.11: The first fifteen eigenvalues of Laplace-Beltrami operator on ellipsoid (5.8) with semiaxis $a = 2$, $b = 1$, and $c = 1.5$ computed via Reilly's formula (5.2).

N	1028	1332
λ_0	0.000061589557452 - 1.181e-08i	0.000004454392027 - 1.6383e-10i
λ_1	0.679137898914703 - 3.1174e-08i	0.679069049187716 - 3.3897e-10i
λ_2	0.964780492647246 - 1.4222e-08i	0.964725424045463 - 3.2683e-09i
λ_3	1.148858957398043 - 2.3677e-09i	1.148799597009603 - 1.3872e-10i
λ_4	2.139809900780672 - 4.7725e-08i	2.139738973104914 - 2.8465e-10i
λ_5	2.343124667785336 - 5.4382e-08i	2.343046805356146 - 1.2698e-08i
λ_6	2.739298612831187 - 3.0951e-09i	2.739235000270296 - 5.2945e-10i
λ_7	3.352461240860305 - 3.2186e-09i	3.352385908356782 - 2.0821e-11i
λ_8	3.371014764510352 - 1.5979e-08i	3.370952402333703 - 3.7414e-09i
λ_9	4.296680176273985 - 6.3767e-08i	4.296591529243099 - 1.5778e-10i
λ_{10}	4.414206939208842 -1.2828e-07i	4.414128723094687 - 2.4454e-08i
λ_{11}	5.105355086705209 - 5.5335e-09i	5.105271219019692 - 9.5858e-10i
λ_{12}	5.670757085149417 - 3.0012e-08i	5.670673540555087 - 2.124e-08i
λ_{13}	5.722502408094088 - 7.2473e-09i	5.722415884539031 - 1.4978e-10i
λ_{14}	6.897740927074761 - 3.2624e-08i	6.897654869366301 - 1.0365e-09i
	1814	2612
λ_0	0.000000129264069 + 9.1008e-12i	-0.000000001898025 + 5.7879e-15i
λ_1	0.679063547189381 + 1.5286e-11i	0.679063466641899 + 7.5232e-15i
λ_2	0.964721361167485 + 3.8747e-09i	0.964721175588664 + 9.0229e-10i
λ_3	1.148795327983074 + 6.9991e-12i	1.148795155105693 + 6.1369e-15i
λ_4	2.139733610449238 + 1.8396e-12i	2.139733531840573 - 1.1368e-14i
λ_5	2.343041230679852 + 1.5101e-08i	2.343041021189079 + 2.6461e-09i
λ_6	2.739229175690334 + 2.4724e-11i	2.739229048104656 + 1.8301e-14i
λ_7	3.352380632999143 + 1.2769e-14i	3.352380407963443 + 2.8391e-14i
λ_8	3.370947939559696 + 1.4108e-09i	3.370947713654945 + 7.3985e-10i
λ_9	4.296584537449716 - 1.8595e-11i	4.296584452417493 - 3.9901e-15i
λ_{10}	4.414122998390591 + 2.8858e-08i	4.414122716119929 + 3.7251e-09i
λ_{11}	5.105265718187255 + 4.244e-11i	5.105265663911774 + 1.1457e-14i
λ_{12}	5.670667156216845 + 8.6947e-09i	5.670666936449536 + 1.3508e-09i
λ_{13}	5.722407698107498 + 1.3634e-12i	5.722407430511693 + 6.6444e-15i
λ_{14}	6.897648188756625 + 1.8644e-10i	6.897647871146280 + 3.7368e-10i

Table 5.12: First non-zero eigenvalue of Laplace-Beltrami operator on ellipsoid (5.8) with $a = 2$, $b = 1$, and $c = {}^{3}/_{2}$.

N	Reilly	Orthogonal Projection
1028	0.679137898914703 - 3.1174e-08i	0.679068861954066 - 3.65e-07i
1332	0.679069049187716 - 3.3897e-10i	0.679064025165072 - 6.1219e-09i
1814	0.679063547189381 + 1.5286e-11i	0.679063481368641 + 9.5876e-08i
2612	0.679063466641899 + 7.5232e-15i	0.679063469816814 + 3.0589e-08i
	OGr Low Order	OGr High Order
1028	0.679158292682947 - 3.9639e-07i	0.679238106758234 - 3.9604e-07i
1332	0.679003674354810 - 5.2091e-09i	0.679078204186933 - 5.332e-09i
1814	0.678989982752958 - 4.8777e-10i	0.679064136830598 + 4.6275e-11i
2612	0.678924107438544 - 2.985e-08i	0.679063478081916 + 9.6943e-13i

Table 5.13: The first fifteen eigenvalues of Laplace-Beltrami operator on Goursat Surface (5.9) computed via Reilly's formula (5.2).

N	840	1284
λ_0	0.006202361838503	0.001206377734761
λ_1	1.028635837537297	1.028475687579542
λ_2	1.034761410112030	1.029636199903901
λ_3	1.040470808733878	1.030495621413112
λ_4	2.295910275305883	2.293253253504226
λ_5	2.298939538488177	2.294910844282620
λ_6	2.304170378920344	2.295387061866696
λ_7	4.074616784383848	4.069765580273434
λ_8	8.258665259778596	8.259930038259338
λ_9	8.269946026960776	8.261525792073794
λ_{10}	8.455723417626279	8.460421753561626
λ_{11}	8.474441715600411 - 0.0010427i	8.463740899956404
λ_{12}	8.474441715600411 + 0.0010427i	8.464091335261026
λ_{13}	9.530133435374188	9.527651229823558
λ_{14}	9.530780640440209	9.528648375792299
	2052	3192
λ_0	-0.000075641611073	-0.000027198757798
λ_1	1.028283320391200	1.028374079404753
λ_2	1.028327812687807	1.028377655483791 - 3.4572e-05i
λ_3	1.028339502882462	1.028377655483791 + 3.4572e-05i
λ_4	2.293365978582924	2.293503151167956 - 3.1254e-05i
λ_5	2.293463989040739 - 7.5809e-05i	2.293503151167956 + 3.1254e-05i
λ_6	2.293463989040739 + 7.5809e-05i	2.293522013549082
λ_7	4.068946403512256	4.069019488428963
λ_8	8.260099824878390	8.260585945897564 - 2.8557e-05i
λ_9	8.260377871915415	8.260585945897564 + 2.8557e-05i
λ_{10}	8.462370295898214 - 0.000146i	8.462782505271223 - 1.2057e-05i
λ_{11}	8.462370295898214 + 0.000146i	8.462782505271223 + 1.2057e-05i
λ_{12}	8.462566383052636	8.462840692384855
λ_{13}	9.528465299836213	9.528800422429317 - 2.5182e-05i
λ_{14}	9.528610239918219 - 3.8617e-05i	9.528800422429317 + 2.5182e-05i

Table 5.14: First non-zero eigenvalue of Laplace-Beltrami operator on Goursat surface (5.9) computed via different RBF methods on $N = 840$, 1284, 2052, and 3192 scattered points.

N	Reilly	Orthogonal Projection
840	1.028635837537297	1.029530967645337
1284	1.028475687579542	1.028333480527482 - 0.00026259i
2052	1.028283320391200	1.028193063980084
3192	1.028374079404753	1.028242542724286
	OGr Low Order	OGr High Order
840	1.041756546797207 - 0.0010112i	1.041761664901879 - 0.00095534i
1284	1.024040672121686 - 0.0062862i	1.024551223541682 - 0.0068042i
2052	1.024361933770759 - 0.061546i	1.027999792841623 - 0.001777i
3192	-0.497862901116229	0.979801688646540

Table 5.15: The first fifteen eigenvalues of Laplace-Beltrami operator on Pilz surface (5.10) computed via Reilly's formula (4.2) with $N = 1120$, 2504, and 3916 scattered points.

N	1120	2504	3916
λ_0	-0.022962522672233	-0.000038307954002	-0.000023164370250
λ_1	0.953006490681902	0.958463600734575	0.958537163854013
λ_2	1.422821113878850	1.546008131923973	1.546279757232460
λ_3	1.792114335595091	1.803064509039900	1.803125795556601
λ_4	3.307337946497406	3.331184007630988	3.331790491870338
λ_5	3.970038876457923	3.998693132660278	3.998987524665552
λ_6	4.247350039555112	4.281121394342198	4.281134508541220
λ_7	4.389747913869662	4.390007533269901	4.390305131900825
λ_8	5.832209908602137	5.834982261687343	5.834786829022552
λ_9	6.304218609051948	6.360161057598139	6.360960744012081
λ_{10}	8.293791561266346	8.302592464330157	8.302964054623667
λ_{11}	8.335136703874854	8.367933169824601	8.368960463581057
λ_{12}	9.726398416008960	9.748929283335197	9.749614973949695
λ_{13}	10.553503929422545	10.588982538905453	10.590193880108448
λ_{14}	12.244767000894822	12.245906016397608	12.246516661341831

Table 5.16: First non-zero eigenvalue of Laplace-Beltrami operator on Pilz surface (5.10) computed via different RBF methods with $N = 1120$, 2504, and 3916 scattered points.

N	Reilly	Orthogonal Projection
1120	0.953006490681902	0.960309530939965
2504	0.958463600734575	0.958652633907363
3916	0.958537163854013	0.957433146507474
	OGr Low Order	OGr High Order
1120	0.923366423081284	0.966941282097027
2504	0.684289166761502	1.036019036426904
3916	1.221162566776748 - 0.45539i	1.064616786384905

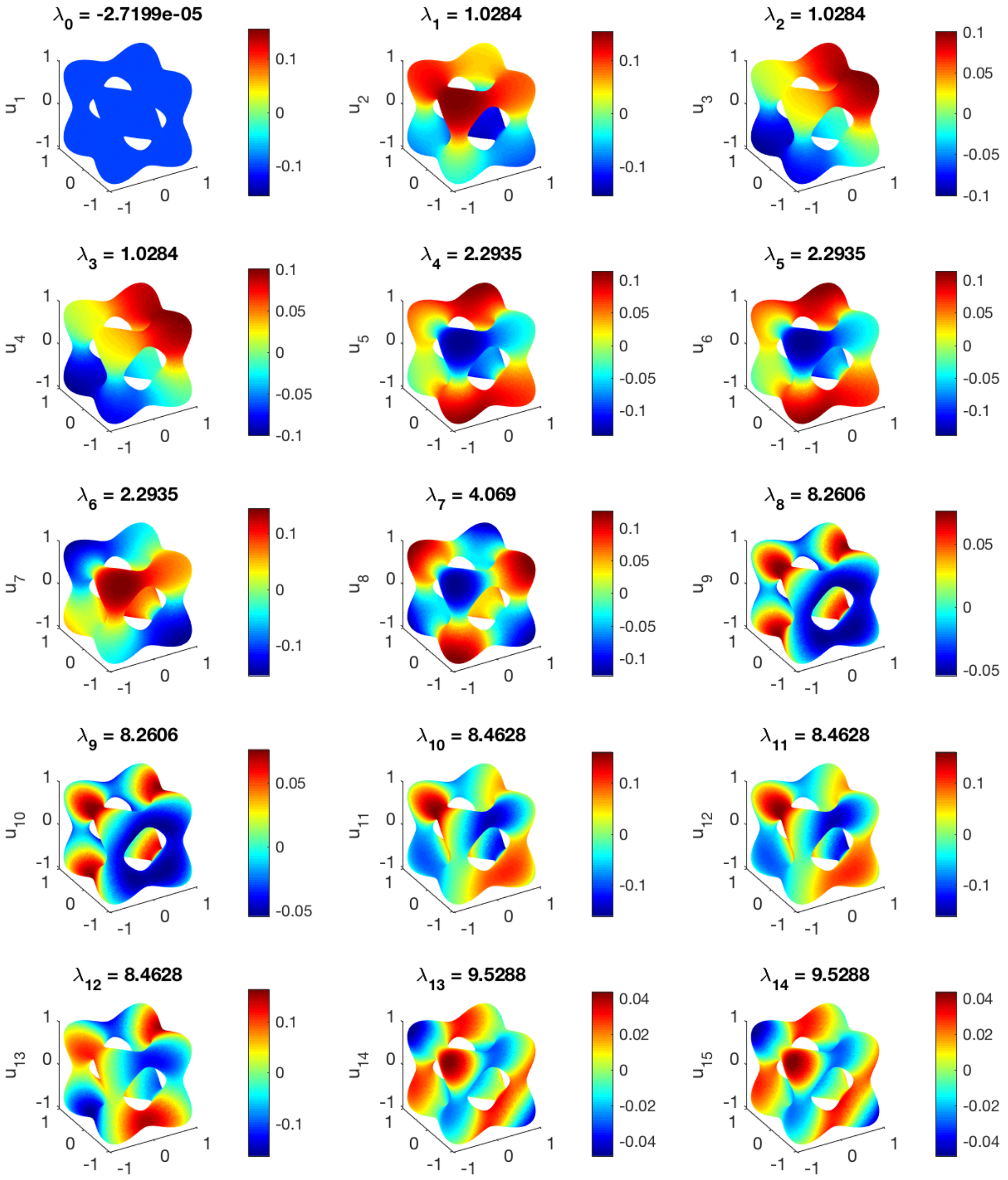


Figure 5.17: Laplace-Beltrami eigenfunctions corresponding to the first 15 eigenvalues on Goursat surface (5.9) computed via Reilly's formula (2.10) on 3192 nodes.

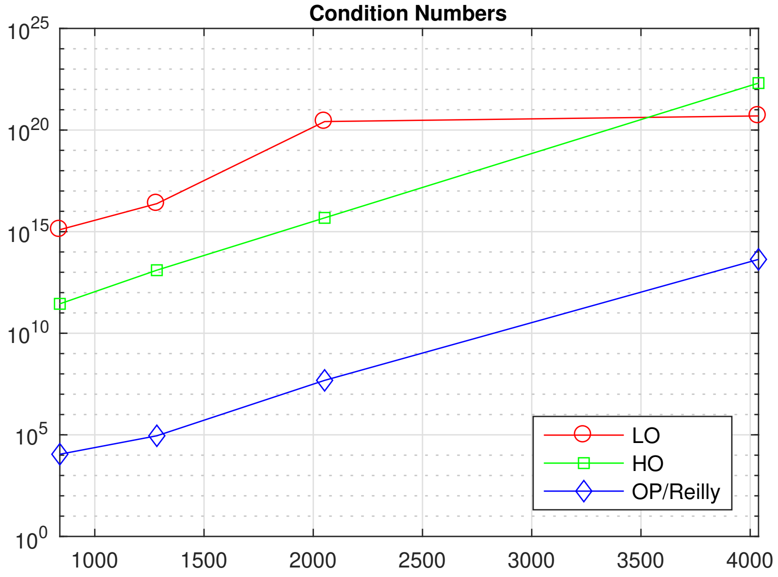


Figure 5.18: Condition numbers of metric matrix used to compute eigenvalues of Laplace-Beltrami operator on Goursat surface (5.9) via LO, HO, and OP/Reilly's methods.

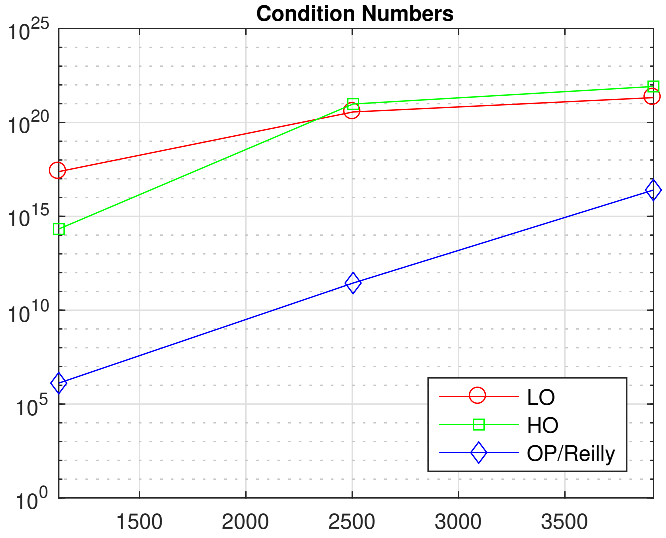
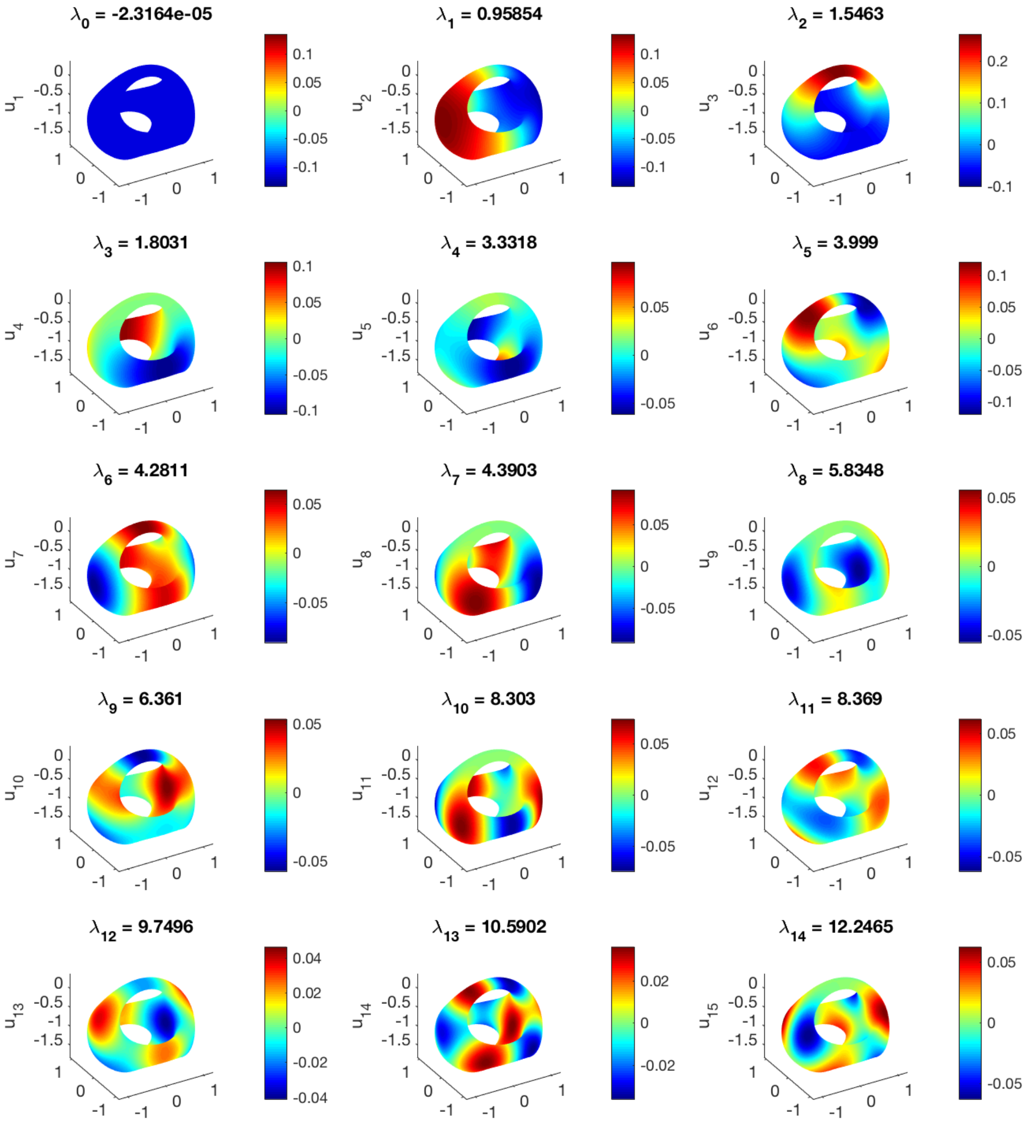


Figure 5.19: Condition numbers of metric matrix used to compute eigenvalues of Laplace-Beltrami operator on Pilz surface (5.10) computed via LO, HO, and OP/Reilly's methods.


λ_0 = -2.3164e-05

λ_1 = 0.95854

λ_2 = 1.5463

λ_3 = 1.8031

λ_4 = 3.3318

λ_5 = 3.999

λ_6 = 4.2811

λ_7 = 4.3903

λ_8 = 5.8348

λ_9 = 6.361

λ_{10} = 8.303

λ_{11} = 8.369

λ_{12} = 9.7496

λ_{13} = 10.5902

λ_{14} = 12.2465

Figure 5.20: Laplace-Beltrami eigenfunctions corresponding to the first 15 eigenvalues on Pilz surface (5.10) computed via Reilly's formula (2.10) on 3916 nodes.

Chapter 6

Final Remarks

The field of kernel-based methods for approximating functions and solving PDEs has been experiencing explosive growth as evidenced by the wealth of publications in recent years. Potential applications include image processing [65], shape optimization [1], solving partial differential equations on manifolds [19, 20, 35], and shape analysis [57, 58].

Working on the project I learned how to notice, connect, and use parallel concepts from different areas of mathematics in research. I also developed geometrical intuition for abstract analytical objects, and most importantly acquired valuable research skills.

6.1 Future Work

The most obvious and immediate improvements for the present methods of computing eigenmodes of Laplace-Beltrami operator would be: (i) formulation of the numerical scheme based on Reilly's formulas for higher-order linear differential operators such as Bi-Laplace-Beltrami operator, and (ii) deeper investigation of relationship between RBF node density, curvature, and shape parameter ε on the one hand, and condition number for RBF interpolation matrix and convergence rate of eigenmodes on the other hand.

The slightly more far-fetched prospective goal would be to obtain analytical estimates for convergence rate of RBF OGr and Reilly's formula based methods similar to the work done by Fuselier and Wright in [21] for RBF projection methods. Meanwhile we propose a few concrete improvements for the techniques described above that are lacking numerical results at the moment.

Specifically, we propose modifications to RBF Orthogonal Gradient (OGr) method which was introduced by Piret in [54] aimed at incorporating boundary conditions for the case of open manifolds. We also describe a multi-layer OGr approach with the intent to introduce more control over the order of convergence of RBF-reconstructed linear differential operators on manifold. The last suggestion is inspired by the idea that eigenvalues of regular Laplace operator defined in the tubular neighborhood of manifold converge to the eigenvalues of Laplace-Beltrami operator on the manifold as the size of the tubular neighborhood goes to zero. The idea, in turn, is supported by asymptotic calculations in section 2.4.

6.1.1 RBF-OGr Method on Manifolds with Boundary

6.1.1.1 Approximation of a Function on Manifolds with Constant Normal Extension

The *key idea* in OGr approach is to expand a function $u : \mathcal{M} \to \mathbb{R}$ from $\mathcal{M}$ to the embedding space $\mathbb{E}$ so that it will be constant along the normal of $\mathcal{M}$ [54] and will satisfy boundary conditions.

Points Allocation

Assume that $\mathcal{M}$ is given implicitly as the zero level set of a distance function $d : \mathbb{E} \to \mathbb{R}$. Then for $k \in \mathbb{N}^+$ denote

$$\begin{aligned} \mathcal{M} &= \big\{ x \in \mathbb{E} : \quad d(x) = 0 \big\}, \\ \mathcal{M}^{\pm k} &= \big\{ x \in \mathbb{E} : \quad d(x) = \pm k \big\}, \end{aligned}$$

Given a set of N_I *interior nodes* $\{x_i\}_{i=1}^{N_I}$ in the *interior* of manifold $\mathcal{M}$ and N_B *boundary nodes* $\{z_i\}_{i=1}^{N_B}$ on the boundary $\partial\mathcal{M}$, we define

$$\begin{aligned} \vec{\boldsymbol{x}}^0 &= \big\{ x_i \big\}_{i=1}^{N_I} \in \ \ \mathcal{M}_I, & \vec{\boldsymbol{x}}^{\pm k} &= \big\{ x_i \pm \delta k \vec{\boldsymbol{n}}_{x_i} \big\}_{i=1}^{N_I} \in \ \ \mathcal{M}_I^{\pm k}, \\ \vec{\boldsymbol{z}}^0 &= \big\{ z_i \big\}_{i=1}^{N_B} \in \partial\mathcal{M} & \vec{\boldsymbol{z}}^{\pm k} &= \big\{ z_i \pm \delta k \vec{\boldsymbol{n}}_{x_i} \big\}_{i=1}^{N_B} \in \partial\mathcal{M}^{\pm k}. \end{aligned}$$

sets of *plus-* and *minus-layers interior* and *boundary nodes*, where $\vec{\boldsymbol{n}}_x$ is the normal vector of $\mathcal{M}$ at the point $x \in \mathcal{M}$ and δ is constant. Denote $N = N_I + N_B$ as the total number of nodes in the interior of $\mathcal{M}$ and on the boundary. Finally, let us denote

$$\vec{\boldsymbol{X}}^0 = \begin{bmatrix} \vec{\boldsymbol{x}}^0 \\ \vec{\boldsymbol{z}}^0 \end{bmatrix}, \quad \vec{\boldsymbol{X}}^{\pm k} = \begin{bmatrix} \vec{\boldsymbol{x}}^{\pm k} \\ \vec{\boldsymbol{z}}^{\pm k} \end{bmatrix}.$$

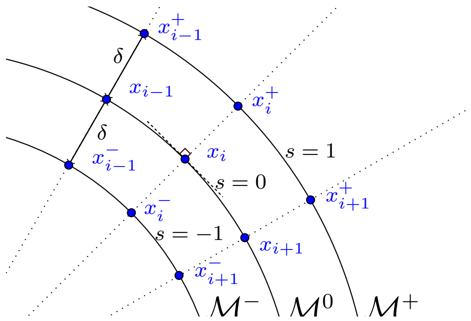


Figure 6.1: Illustration of *plus* and *minus* layers of one-dimensional manifold $\mathcal{M}$

Function Reconstruction

Given a smooth function $\phi : \mathbb{R} \to \mathbb{R}$ and a points ξ, the RBF is defined as $\varphi(x, \xi) := \phi(\|x - \xi\|)$, where $\|\cdot\|$ is a metric in $\mathbb{E}$. We say that φ is centered at ξ and estimated at x. Then, given a set $\vec{\boldsymbol{\xi}} = \{\xi_i\}$ of centers, an arbitrary function $u : \mathbb{E} \to \mathbb{R}$ can be approximated by linear combination of RBFs:

$$u(x) \approx \sum_{i=1}^{N_\xi} \alpha_i \varphi(x, \xi_i), \tag{6.1}$$

where $\vec{\boldsymbol{\alpha}} = \{\alpha_i\}_{i=1}^{N_\xi}$ is a vector of coefficients of RBF approximation of u. Choosing $\vec{\boldsymbol{\xi}}$ as

$$\vec{\boldsymbol{\xi}} = \left\{ \vec{\boldsymbol{X}}^0 \cup \vec{\boldsymbol{X}}^{+1} \cup \dots \cup \vec{\boldsymbol{X}}^{k_{\max}} \cup \vec{\boldsymbol{X}}^{-1} \cup \dots \cup \vec{\boldsymbol{X}}^{-k_{\min}} \right\},$$

we can write a RBF expansion of u for the nodes $\vec{\boldsymbol{x}}^k$ in the matrix form $u\left(\vec{\boldsymbol{x}}^k\right) = \varphi\big(\vec{\boldsymbol{x}}^k, \vec{\boldsymbol{\xi}}\big)\vec{\boldsymbol{\alpha}}$. The total number of nodes then is $N_\xi = (k_{\min} + k_{\max} + 1)\, N$.

Remark. It is common for *centers* $\vec{\boldsymbol{\xi}}$ to coincide with nodes $\vec{\boldsymbol{X}}$, however placing them on boundary $\partial\mathcal{M}$ proved to be inefficient [14]. Thus, we propose to place $(k_{\min} + k_{\max} + 1)N_I$ centers to the *interior* nodes $\vec{\boldsymbol{x}}$, and remaining $(k_{\min} + k_{\max} + 1)N_B$ centers outside but very close to $\partial\mathcal{M}^{\pm k}$, i.e. for any point $z \in \partial\mathcal{M}^{\pm k}$ there exists $\xi_i \in \vec{\boldsymbol{\xi}}$ s.t.$\|\xi_i - z\|_{\mathbb{E}} \le \delta$, where δ is the distance between layers.

Denote $\boldsymbol{M}$ the *metric matrix* of RBF functions centered at $\vec{\boldsymbol{\xi}}$ and estimated at $\vec{\boldsymbol{x}}^k$:

$$\boldsymbol{M} := \varphi\big(\vec{\boldsymbol{x}}^k, \vec{\boldsymbol{\xi}}\big) = \begin{bmatrix} \varphi\left(x_1^k, \xi_1\right), & \dots & \varphi\left(x_1^k, \xi_{N_\xi}\right) \\ \vdots & \ddots & \vdots \\ \varphi\left(x_{N_I}^k, \xi_1\right), & \dots & \varphi\left(x_{N_I}^k, \xi_{N_\xi}\right) \end{bmatrix},$$

then the size of $\boldsymbol{M}$ is $N_I \times N_\xi$ and

$$u\left(\vec{\boldsymbol{x}}^k\right) = \boldsymbol{M}\vec{\boldsymbol{\alpha}}. \tag{6.2}$$

Now we will enhance the approximation of u by enforcing constant normal extension and boundary conditions, so that u will have the same values on all layers, $u\big(\vec{\boldsymbol{X}}^0\big) = u\big(\vec{\boldsymbol{X}}^{\pm k}\big)$, and will satisfy $\mathcal{B}(u)(z) = 0$ for all $z \in \partial\mathcal{M}$. The former can be done in either Low Order or the High Order manner. The latter is done by applying operator $\mathcal{B}$ to (6.2), so that denoting $\boldsymbol{B} := \varphi_{\mathcal{B}}\big(\vec{\boldsymbol{z}}^k, \vec{\boldsymbol{\xi}}\big) = \left\{\mathcal{B}\left(\varphi\right)\left(\vec{\boldsymbol{z}}^k, \vec{\boldsymbol{\xi}}\right)\right\}$ and $\mathbf{0}, \mathbf{1}$ - zero and identity matrices, we write

$$\boldsymbol{B}\vec{\boldsymbol{\alpha}} = \mathbf{0}_{N_B \times 1}. \tag{6.3}$$

Low Order implies that values of u on different layers are the same as the ones on the zero layer:

$$\begin{bmatrix} \varphi\big(\vec{\boldsymbol{X}}^{0}\big) - \varphi\big(\vec{\boldsymbol{X}}^{+1}\big) \\ \vdots \\ \varphi\big(\vec{\boldsymbol{X}}^{0}\big) - \varphi\big(\vec{\boldsymbol{X}}^{k_{\max}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^{0}\big) - \varphi\big(\vec{\boldsymbol{X}}^{-1}\big) \\ \vdots \\ \varphi\big(\vec{\boldsymbol{X}}^{0}\big) - \varphi\big(\vec{\boldsymbol{X}}^{-k_{\min}}\big) \end{bmatrix} \vec{\boldsymbol{\alpha}} = \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \vdots \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \vdots \\ \mathbf{0}_{N\times 1} \end{bmatrix}. \tag{6.4}$$

High Order proposes to enforce normal derivatives $(\partial_{\vec{\boldsymbol{n}}})^i$ of u to be zero:

$$\begin{bmatrix} (\vec{\boldsymbol{n}} \cdot \nabla)\, \varphi\big(\vec{\boldsymbol{X}}^{0}, \vec{\boldsymbol{\xi}}\big) \\ (\vec{\boldsymbol{n}} \cdot \nabla)^{2}\, \varphi\big(\vec{\boldsymbol{X}}^{0}, \vec{\boldsymbol{\xi}}\big) \\ \vdots \\ (\vec{\boldsymbol{n}} \cdot \nabla)^{p}\, \varphi\big(\vec{\boldsymbol{X}}^{0}, \vec{\boldsymbol{\xi}}\big) \end{bmatrix} \vec{\boldsymbol{\alpha}} = \begin{bmatrix} \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \\ \vdots \\ \mathbf{0}_{N\times 1} \end{bmatrix} \tag{6.5}$$

where $p = k_{\max} + k_{\min}$.

If we denote matrices on the left-hand side of (6.4) and (6.5) by $\boldsymbol{N}_{LO}$ and $\boldsymbol{N}_{HO}$ respectively and agree that $\boldsymbol{N}$ stands for either $\boldsymbol{N}_{LO}$ or $\boldsymbol{N}_{HO}$, we can write a RBF expansion of u and compute corresponding coefficients $\vec{\boldsymbol{\alpha}}$ as following:

$$\begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{B} \\ \boldsymbol{N} \end{bmatrix} \vec{\boldsymbol{\alpha}} = \begin{bmatrix} \mathbf{1}_{N_I\times N_I} \\ \mathbf{0}_{N_B\times N_I} \\ \mathbf{0}_{pN\times N_I} \end{bmatrix} u\left(\vec{\boldsymbol{x}}^{0}\right) \implies \boxed{\vec{\boldsymbol{\alpha}} = \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{B} \\ \boldsymbol{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{1}_{N_I\times N_I} \\ \mathbf{0}_{N_B\times N_I} \\ \mathbf{0}_{pN\times N_I} \end{bmatrix} u\left(\vec{\boldsymbol{x}}^{0}\right)}. \tag{6.6}$$

Note. To verify the invertibility of the leftmost matrix in (6.6) we break down sizes as follows: (i) the size of the "interpolation matrix" $\boldsymbol{M}$ is $N_I \times N_\xi$, (ii) the size of the "boundary conditions" matrix $\boldsymbol{B}$ is $N_B \times N_\xi$, and (iii) the size of the "constant normal extension" matrix $\boldsymbol{N}$ is $(k_{\min} + k_{\max})\, N \times N_\xi = pN \times N_\xi$. Thus we ensure that the composite matrix is square.

Remark. The vector of coefficients $\vec{\boldsymbol{\alpha}}$ inherits notation from $\boldsymbol{N}$, i.e. we get $\vec{\boldsymbol{\alpha}}_{LO}$ and $\vec{\boldsymbol{\alpha}}_{HO}$ respectively if we substitute $\boldsymbol{N}_{LO}$ from (6.4) or $\boldsymbol{N}_{HO}$ from (6.5) into (6.6) .

6.1.1.2 Operator Reconstruction

Here we derive *Operator Matrix* $\boldsymbol{D}$ such that given a vector $u\left(\vec{\boldsymbol{x}}^0\right)$, we could obtain values of $\mathcal{L}(u)\left(\vec{\boldsymbol{x}}^0\right)$:

$$\boldsymbol{D}u\left(\vec{\boldsymbol{x}}^0\right) = \mathcal{L}\left(u\right)\left(\vec{\boldsymbol{x}}^0\right).$$

Applying operator $\mathcal{L}$ to (6.1), we get $\mathcal{L}\left(u\right)\left(\vec{\boldsymbol{x}}^0\right) = \mathcal{L}\left(\varphi\right)\left(\vec{\boldsymbol{x}}^0\right) := \varphi_{\mathcal{L}}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big)$. Denoting $\boldsymbol{M}_{\mathcal{L}}$ the *operator metric* matrix $\varphi_{\mathcal{L}}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big)$:

$$\boldsymbol{M}_{\mathcal{L}} := \varphi_{\mathcal{L}}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) = \begin{bmatrix} \mathcal{L}(\varphi)\left(x_1, \xi_1\right), & \dots & \mathcal{L}(\varphi)\left(x_1, \xi_{N_\xi}\right) \\ \vdots & \ddots & \vdots \\ \mathcal{L}(\varphi)\left(x_{N_I}, \xi_1\right), & \dots & \mathcal{L}(\varphi)\left(x_{N_I}, \xi_{N_\xi}\right) \end{bmatrix},$$

we turn our target expression for differentiation matrix D into

$$\boldsymbol{D}u\left(\vec{\boldsymbol{x}}^0\right) = \boldsymbol{M}_{\mathcal{L}}\, \vec{\boldsymbol{\alpha}}.$$

Substituting $\vec{\boldsymbol{\alpha}}$ from (6.6) and solving for D, we get

$$\boldsymbol{D} = \boldsymbol{M}_{\mathcal{L}} \begin{bmatrix} \boldsymbol{M} \\ \boldsymbol{B} \\ \boldsymbol{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{1}_{N_I \times N_I} \\ \mathbf{0}_{N_B \times N_I} \\ \mathbf{0}_{pN \times N_I} \end{bmatrix} \tag{6.7}$$

Remark. For $\boldsymbol{N} = \boldsymbol{N}_{LO}$ from (6.4), we get D_{LO} representing *Low Order* reconstruction of $\mathcal{L}$, and for $\boldsymbol{N} = \boldsymbol{N}_{HO}$ from (6.5), we get D_{HO} representing *High Order* differentiation matrix.

In order to estimate eigenmodes of (3.1), we solve the eigenvalue problem

$$\boldsymbol{D}u\left(\vec{\boldsymbol{x}}^0\right) = \lambda u\left(\vec{\boldsymbol{x}}^0\right).$$

Example 6.1 (*Laplace-Beltrami Operator with Neumann BC*)**.**

For the first problem in (3.2) we have the operator $\mathcal{L} = -\Delta$ which is of second order, so we need two extra layers (one positive and one negative) for Low Order and normal components of derivatives $\left(\vec{\boldsymbol{n}} \cdot \nabla\right)^i$ of up to the second order: $i \in \{1, 2\}$. Thus we require $3N$ RBF centers $\vec{\boldsymbol{\xi}}$ in order to make the middle matrix in (6.7) invertible. Boundary Operator $\mathcal{B}\left(u\right) = \big(\vec{\boldsymbol{b}} \cdot \nabla\big)u$, where $\vec{\boldsymbol{b}}$ is the binormal unit vector representing flux through the boundary, which is defined as $\vec{\boldsymbol{b}} = \vec{\boldsymbol{t}}_{\partial\mathcal{M}} \times \vec{\boldsymbol{n}}$ so that

$$D_{LO} = \varphi_{\Delta}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \big(\vec{\boldsymbol{b}} \cdot \nabla\big)\varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0\big) - \varphi\big(\vec{\boldsymbol{X}}^{+1}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0\big) - \varphi\big(\vec{\boldsymbol{X}}^{-1}\big) \end{bmatrix} \begin{bmatrix} \mathbf{1}_{N_I \times N_I} \\ \mathbf{0}_{N_B \times N_I} \\ \mathbf{0}_{N \times N_I} \\ \mathbf{0}_{N \times N_I} \end{bmatrix},$$

and

$$D_{HO} = \varphi_{\Delta}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \big(\vec{\boldsymbol{b}} \cdot \nabla\big)\varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \big(\vec{\boldsymbol{n}} \cdot \nabla\big)\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ \big(\vec{\boldsymbol{n}} \cdot \nabla\big)^2\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \end{bmatrix} \begin{bmatrix} \mathbf{1}_{N_I \times N_I} \\ \mathbf{0}_{N_B \times N_I} \\ \mathbf{0}_{N \times N_I} \\ \mathbf{0}_{N \times N_I} \end{bmatrix},$$

where $\varphi_{\Delta}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big)$ is the matrix with entries $\Big\{\varphi_{\Delta}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big)\Big\}_{i,j} = \Delta\varphi\left(\|x_i - \xi_j\|\right)$ which are the Laplacians of basis functions φ centered at $\xi_j \in \vec{\boldsymbol{\xi}}$ and estimated at $x_i \in \vec{\boldsymbol{x}}^0$.

Example 6.2 (*Biharmonic Operator with Dirichlet and Pinned Boundary Conditions.*)**.** The second problem in (3.2) implies operator $\mathcal{L} = \Delta^2$ which is of order 4, so we need four extra layers for Low Order and normal derivatives $\left(\vec{\boldsymbol{n}} \cdot \nabla\right)^i$ up to fourth order: $i \in \{1, 2, 3, 4\}$. Thus we require $5N + N_B$ RBF centers $\vec{\boldsymbol{\xi}}$, where extra N_B centers are required to match the additional N_B equations coming from the second set of boundary conditions, since $\mathcal{B}$ actually stands for a system of two boundary conditions operators $\mathcal{B}_1$ - Dirichlet, and $\mathcal{B}_2$ – Pinned, and

$$\mathcal{B}\left(u\right) := \begin{cases} \mathcal{B}_1\left(u\right) = u \\ \mathcal{B}_2\left(u\right) = \left(\Delta - \left(\boldsymbol{N} \cdot \nabla\right) - \left(\boldsymbol{N} \cdot \nabla\right)^2\right)\left(u\right) \end{cases}$$

Then the differentiation matrices for operator Δ^2 with such BCs can be computed as

$$D_{LO} = \varphi_{\Delta^2}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \big(\Delta - (\boldsymbol{N}\cdot\nabla) - (\boldsymbol{N}\cdot\nabla)^2\big)\varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0\big) - \varphi\big(\vec{\boldsymbol{X}}^{+1}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0\big) - \varphi\big(\vec{\boldsymbol{X}}^{+2}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0\big) - \varphi\big(\vec{\boldsymbol{X}}^{-1}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0\big) - \varphi\big(\vec{\boldsymbol{X}}^{-2}\big) \end{bmatrix} \begin{bmatrix} \mathbf{1}_{N_I\times N_I} \\ \mathbf{0}_{N_B\times N_I} \\ \mathbf{0}_{N_B\times N_I} \\ \mathbf{0}_{N\times N_I} \\ \mathbf{0}_{N\times N_I} \\ \mathbf{0}_{N\times N_I} \\ \mathbf{0}_{N\times N_I} \end{bmatrix},$$

and

$$D_{HO} = \varphi_{\Delta^2}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \big(\Delta - (\boldsymbol{N}\cdot\nabla) - (\boldsymbol{N}\cdot\nabla)^2\big)\varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ (\vec{\boldsymbol{n}}\cdot\nabla)\,\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ (\vec{\boldsymbol{n}}\cdot\nabla)^2\,\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ (\vec{\boldsymbol{n}}\cdot\nabla)^3\,\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ (\vec{\boldsymbol{n}}\cdot\nabla)^4\,\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \end{bmatrix} \begin{bmatrix} \mathbf{1}_{N_I\times N_I} \\ \mathbf{0}_{N_B\times N_I} \\ \mathbf{0}_{N_B\times N_I} \\ \mathbf{0}_{N\times N_I} \\ \mathbf{0}_{N\times N_I} \\ \mathbf{0}_{N\times N_I} \\ \mathbf{0}_{N\times N_I} \end{bmatrix},$$

where $\varphi_{\Delta^2}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big)$ is defined similar to $\varphi_{\Delta}\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big)$ in Example 6.1.

6.1.1.3 Inverse Operator Reconstruction

It is possible to use the routine described in section 6.1.1.2 to reconstruct *inverse operator* $\mathcal{L}^{-1} : \mathcal{C}^\infty(\mathcal{M}_I) \to \mathcal{C}^\infty(\mathcal{M}_I)$, such that $\mathcal{L}^{-1}\mathcal{L} = \mathcal{L}\mathcal{L}^{-1} = I$ – identity operator. Let us introduce $v := \mathcal{L}(u)$, so that $u = \mathcal{L}^{-1}(v)$, and substitute it to the problem (3.1):

$$\begin{cases} \mathcal{L}^{-1}(v)(x) = {}^1\!/\!{}_\lambda\, v(x), & x \in \mathcal{M}_I \\ \mathcal{B}(u)(x) = 0, & x \in \partial\mathcal{M} \end{cases} \tag{6.8}$$

Then we approximate u by radial basis functions similarly to (6.2) denoting coefficients $\vec{\boldsymbol{\beta}}$:

$$u\left(\vec{\boldsymbol{x}}^0\right) = \boldsymbol{M}\vec{\boldsymbol{\beta}} \implies v\left(\vec{\boldsymbol{x}}^0\right) = \boldsymbol{M}_{\mathcal{L}}\vec{\boldsymbol{\beta}}. \tag{6.9}$$

In order to solve (6.8), we will derive *Inverse Operator Matrix* R corresponding to $\mathcal{L}^{-1}$:

$$Rv\left(\vec{\boldsymbol{x}}^0\right) = \mathcal{L}^{-1}(v)\left(\vec{\boldsymbol{x}}^0\right) = u\left(\vec{\boldsymbol{x}}^0\right) = \boldsymbol{M}\vec{\boldsymbol{\beta}}$$

Now, using matrices $\boldsymbol{B}$ from (6.3) and $\boldsymbol{N} = \boldsymbol{N}_{LO/HO}$ from (6.4) or (6.5), we get $\vec{\boldsymbol{\beta}}$:

$$\vec{\boldsymbol{\beta}} = \begin{bmatrix} \boldsymbol{M}_{\mathcal{L}} \\ \boldsymbol{B} \\ \boldsymbol{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{1}_{N_I \times N_I} \\ \mathbf{0}_{N_B \times N_I} \\ \mathbf{0}_{pN \times N_I} \end{bmatrix} v\left(\vec{\boldsymbol{x}}^0\right) \tag{6.10}$$

Remark. As before, $\vec{\boldsymbol{\beta}}_{LO}$ corresponds for $\boldsymbol{N} = \boldsymbol{N}_{LO}$ and $\vec{\boldsymbol{\beta}}_{HO}$ for $\boldsymbol{N} = \boldsymbol{N}_{HO}$ in (6.10).

Substituting coefficients $\vec{\boldsymbol{\beta}}$ in (6.9), we get

$$R = \boldsymbol{M} \begin{bmatrix} \boldsymbol{M}_{\mathcal{L}} \\ \boldsymbol{B} \\ \boldsymbol{N} \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{1}_{N_I \times N_I} \\ \mathbf{0}_{N_B \times N_I} \\ \mathbf{0}_{pN \times N_I} \end{bmatrix}$$

with R_{LO} standing for Low Order and R_{HO} for High Order for corresponding matrix $\boldsymbol{N}$.

To estimate eigenmodes of (3.1) we solve the following eigenproblem:

$$Rv\left(\vec{\boldsymbol{x}}^0\right) = {}^1\!/\!_\lambda\, v\left(\vec{\boldsymbol{x}}^0\right).$$

6.1.2 Extended Multiple Layers RBF Orthogonal Gradient Approaches

The meshless approach based on radial basis functions is to approximate a solution by a linear superposition of radial symmetric functions, e.g., given a set of RBF centers $\vec{\boldsymbol{\xi}} = \{\xi_i\}$, and a function $\varphi(r) : \mathbb{R} \to \mathbb{R}$ (called *basic function*), the solution is approximated by

$$u(x) \approx \sum_{i=1}^{N_\xi} \alpha_i \varphi\left(\|x - \xi_i\|\right) \tag{6.11}$$

where $\xi_i \in \mathcal{M},\ i = 1, \dots, N_\xi$, and N_ξ is the number of points in $\vec{\boldsymbol{\xi}}$.

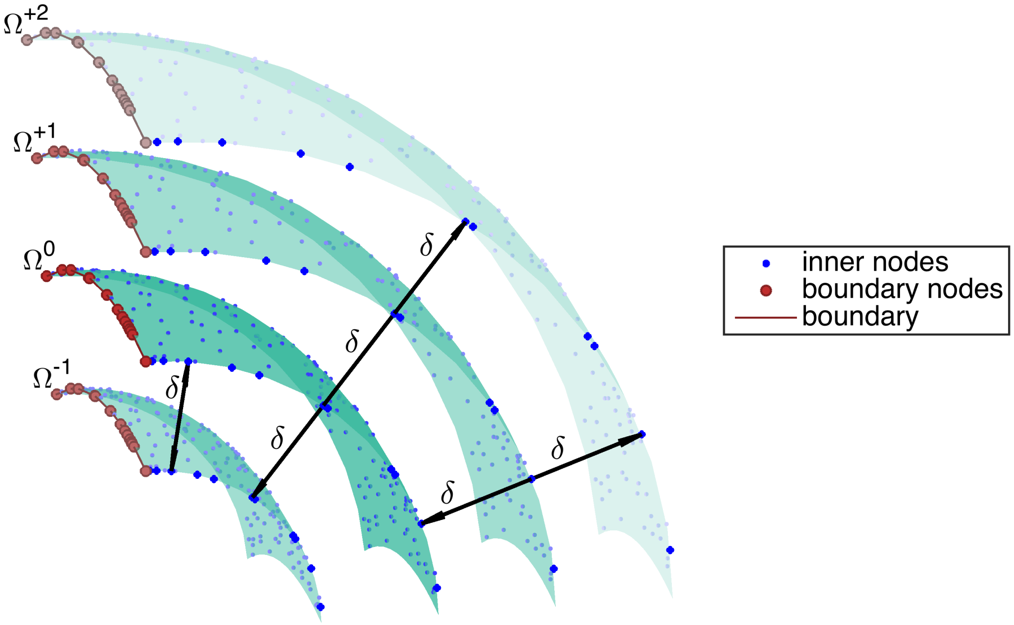


Figure 6.2: Illustration of points on two plus-layers ($\mathcal{M}^{+1}$ and $\mathcal{M}^{+2}$) and a minus-layer ($\mathcal{M}^{-1}$) of a two-dimensional manifold $\mathcal{M}$.

In order to solve the surface problem, the locations of RBF centers and the collocation points are chosen in a special way. Denote $\partial\mathcal{M}$ as the boundary of the manifold $\mathcal{M}$. Like in the previous subsection, suppose we have a point cloud sampling consisting of a set of N_I interior nodes $\{x_i\}_{i=1}^{N_I} \in \mathcal{M}$ and N_B boundary nodes $\{z_i\}_{i=1}^{N_B} \in \partial\mathcal{M}$ with $N = N_I + N_B$ the total number of collocation points in the interior of $\mathcal{M}$ and on the boundary of $\mathcal{M}$. Define the sets of plus- and minus-layers interior and boundary nodes:

$$\vec{\boldsymbol{x}}^{\,\pm k} = \{x_i \pm \delta k \vec{\boldsymbol{n}}(x_i)\}_{i=1}^{N_I}, \quad \vec{\boldsymbol{z}}^{\,\pm k} = \{z_i \pm \delta k \vec{\boldsymbol{n}}(x_i)\}_{i=1}^{N_B}.$$

where δ is a fixed parameter called *distance between layers*. See Fig. 6.2 for an illustration of these points. In general, the point cloud datasets can be scattering points.

Finally, let us denote

$$\vec{\boldsymbol{X}}^{\pm k} = \begin{bmatrix} \vec{\boldsymbol{x}}^{\pm k} \\ \vec{\boldsymbol{z}}^{\pm k} \end{bmatrix}.$$

Suppose we want to solve the Laplace-Beltrami eigenvalue problem

$$\begin{cases} -\Delta_{\mathcal{M}} u = \lambda u, & x \in \mathcal{M}, \\ \qquad\quad u = 0, & x \in \partial\mathcal{M} \end{cases} \tag{6.12}$$

Choosing

$$\vec{\boldsymbol{\xi}} = \left\{ \vec{\boldsymbol{X}}^0 \cup \vec{\boldsymbol{X}}^1 \cup \cdots \cup \vec{\boldsymbol{X}}^{k_{max}} \cup \vec{\boldsymbol{X}}^{-1} \cup \cdots \cup \vec{\boldsymbol{X}}^{-k_{min}} \right\}, \qquad N_\xi = (k_{max} + k_{min} + 1)\, N,$$

with given k_{max} and k_{min}, we can build RBF approximation of u by enforcing the boundary conditions and the derivatives along the normal direction to vanish. For example, if $k_{max} = k_{min} = 1$, we use three layers to construct the approximation and require both $u_{\boldsymbol{n}} = 0$ and $u_{\boldsymbol{nn}} = 0$. One way to implement this is through the finite difference estimation in the normal direction, i.e.,

$$u_{\boldsymbol{n}}\big(\vec{\boldsymbol{X}}^0\big) \approx \frac{u\big(\vec{\boldsymbol{X}}^1\big) - u\big(\vec{\boldsymbol{X}}^{-1}\big)}{2\delta} = 0, \quad \text{and} \quad u_{\boldsymbol{nn}}\big(\vec{\boldsymbol{X}}^0\big) \approx \frac{u\big(\vec{\boldsymbol{X}}^1\big) - 2u\big(\vec{\boldsymbol{X}}^0\big) + u\big(\vec{\boldsymbol{X}}^{-1}\big)}{\delta^2} = 0,$$

are both second-order accurate. Next, construct an approximation of u in the form of (6.11) satisfying

$$\begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) - \varphi\big(\vec{\boldsymbol{X}}^1, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^1, \vec{\boldsymbol{\xi}}\big) - 2\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) + \varphi\big(\vec{\boldsymbol{X}}^{-1}, \vec{\boldsymbol{\xi}}\big) \end{bmatrix} \vec{\boldsymbol{\alpha}} = \begin{bmatrix} u\big(\vec{\boldsymbol{x}}^0\big) \\ u\big(\vec{\boldsymbol{z}}^0\big) \\ \mathbf{0}_{N\times 1} \\ \mathbf{0}_{N\times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{N_I\times N} \\ \mathbf{0}_{N_B\times N} \\ \mathbf{0}_{N\times N} \\ \mathbf{0}_{N\times N} \end{bmatrix} u\big(\vec{\boldsymbol{x}}^0\big).$$

where $\vec{\boldsymbol{\alpha}} = \{\alpha_1, \ldots, \alpha_N\}$ is the vector of expansion coefficients $\alpha_i,\ i = 1, \ldots, N_\xi = 3N$ from (6.11), and $\varphi(\vec{\boldsymbol{X}}^k, \vec{\boldsymbol{\xi}})$ is N by $3N$ matrix with the RBF functions centered at $\vec{\boldsymbol{\xi}}$ and estimated at $\vec{\boldsymbol{X}}^k$, i.e.,

$$\varphi\big(\vec{\boldsymbol{X}}^k, \vec{\boldsymbol{\xi}}\big) = \begin{bmatrix} \varphi(\|x_1 - \xi_1\|) & \cdots & \varphi(\|x_1 - \xi_{N_\xi}\|) \\ \vdots & \ddots & \vdots \\ \varphi(\|z_{N_B} - \xi_1\|) & \cdots & \varphi(\|z_{N_B} - \xi_{N_\xi}\|) \end{bmatrix}$$

and it leads to

$$\vec{\boldsymbol{\alpha}} = \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) - \varphi\big(\vec{\boldsymbol{X}}^1, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^1, \vec{\boldsymbol{\xi}}\big) - 2\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) + \varphi\big(\vec{\boldsymbol{X}}^{-1}, \vec{\boldsymbol{\xi}}\big) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_{N_I \times N} \\ \mathbf{0}_{N_B \times N} \\ \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix} u\big(\vec{\boldsymbol{x}}^0\big). \tag{6.13}$$

where $\mathbf{I}_{N_I \times N}$ is the N_I by N identity matrix and $\mathbf{0}_{N_B \times N}$ is the N_B by N zero matrix.

Similar to the idea to create higher order scheme in finite difference framework by using wider stencils, one can use multiple layers to generate more accurate approximation. For example, one can use fourth order approximation for $u_n = 0$ and $u_{nn} = 0$ and second order approximation for $u_{nnn} = 0$ and $u_{nnnn} = 0$ for 5 layers computation.

Another way to construct the approximation is based on the differentiability of RBF, i.e.

$$\begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi_n\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi_{nn}\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \end{bmatrix} \vec{\boldsymbol{\alpha}} = \begin{bmatrix} u\big(\vec{\boldsymbol{x}}^0\big) \\ u\big(\vec{\boldsymbol{z}}^0\big) \\ \mathbf{0}_{N \times 1} \\ \mathbf{0}_{N \times 1} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_{N_I \times N} \\ \mathbf{0}_{N_B \times N} \\ \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix} u\big(\vec{\boldsymbol{x}}^0\big).$$

where $\varphi_n\big(\vec{\boldsymbol{X}}^k, \vec{\boldsymbol{\xi}}\big)$ and $\varphi_{nn}\big(\vec{\boldsymbol{X}}^k, \vec{\boldsymbol{\xi}}\big)$ are the $N \times 3N$ matrices with $\varphi, \varphi_n, \varphi_{nn}$ centered at $\vec{\boldsymbol{\xi}}$ and estimated at $\vec{\boldsymbol{X}}^k$. Thus we have an expression for vector of coefficients α_i:

$$\vec{\boldsymbol{\alpha}} = \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi_n\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi_{nn}\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_{N_I \times N} \\ \mathbf{0}_{N_B \times N} \\ \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix} u\big(\vec{\boldsymbol{x}}^0\big). \tag{6.14}$$

Now we can formulate the RBF representation for (6.12). With the vanishing derivatives approximation along the normal directions, we can then replace Laplace-Beltrami operator in (6.12) by regular Laplace operator. Thus

$$-\Big[\Delta\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big)\Big]\,\vec{\boldsymbol{\alpha}} = \lambda u\big(\vec{\boldsymbol{X}}^0\big).$$

Combining expression above with (6.13) yields approximation matrix for Laplace-Beltrami operator:

$$\Delta_{\mathcal{M}} \approx \Delta_{\mathcal{M}}^{D_1} = \left[\Delta\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big)\right] \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) - \varphi\big(\vec{\boldsymbol{X}}^1, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{X}}^1, \vec{\boldsymbol{\xi}}\big) - 2\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) + \varphi\big(\vec{\boldsymbol{X}}^{-1}, \vec{\boldsymbol{\xi}}\big) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_{N_I \times N} \\ \mathbf{0}_{N_B \times N} \\ \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix}$$

Using formula (6.14) to represent $\vec{\boldsymbol{\alpha}}$ in the matrix approximation of Laplace-Beltrami operator we get:

$$\Delta_{\mathcal{M}} \approx \Delta_{\mathcal{M}}^{D_2} = \left[\Delta\varphi\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big)\right] \begin{bmatrix} \varphi\big(\vec{\boldsymbol{x}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi\big(\vec{\boldsymbol{z}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi_n\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \\ \varphi_{nn}\big(\vec{\boldsymbol{X}}^0, \vec{\boldsymbol{\xi}}\big) \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{I}_{N_I \times N} \\ \mathbf{0}_{N_B \times N} \\ \mathbf{0}_{N \times N} \\ \mathbf{0}_{N \times N} \end{bmatrix}$$

6.1.3 Improvements for Orthogonal Gradient Method

Inspired by Piret's High Order Orthogonal Gradient method [54] described in subsection 3.3.5, we propose to reduce the size of differentiation matrix (3.5) by combining two conditions $\big(\vec{\boldsymbol{n}} \cdot \nabla\big)\Phi\big(\vec{\boldsymbol{\xi}}\big) = \vec{\mathbf{0}}$ and $\big(\vec{\boldsymbol{n}} \cdot \nabla\big)^2\Phi\big(\vec{\boldsymbol{\xi}}\big) = \vec{\mathbf{0}}$ in (3.4) $\mathcal{S}_{\boldsymbol{n}}\big(\vec{\boldsymbol{X}}^0\big) = \mathbf{0}$ and $\mathcal{S}_{\boldsymbol{nn}}\big(\vec{\boldsymbol{X}}^0\big) = \mathbf{0}$ into a single one $\big|\big(\vec{\boldsymbol{n}} \cdot \nabla\big)\Phi\big(\vec{\boldsymbol{\xi}}\big)\big|^2 + \big|\big(\vec{\boldsymbol{n}} \cdot \nabla\big)^2\Phi\big(\vec{\boldsymbol{\xi}}\big)\big|^2 = \vec{\mathbf{0}}$. Thus we reduce the size of the matrix from $3N$ to $2N$. This significantly decreases computational complexity as normally one would need to invert that matrix in order to compute coefficients of interpolation.

6.2 Conclusion

In this work we present analytical justification (chapter 2) and numerical results (section 5.1 and section 5.2) in an attempt to demonstrate improved performance of RBF-based method utilizing Reilly's formulas for the Laplace-Beltrami operator over other commonly suggested methods.

The superior performance of the proposed method based on Reilly's formulas can be attributed to reduced size of the metric matrix and the smaller condition number, as well as generalized eigenproblem formulation that allows one to reduce or eliminate spurious modes with non-zero imaginary part.

Additionally, the comparative numerical study of performance of various related methods is presented in chapter 5. Examples include planar curves and smooth compact surfaces of various genus.

Potential future work section contains multiple suggestions on improvement of algorithms, as well as extending them to the cases of manifolds with boundary. One of the proposed techniques is in fact an amalgamation of Orthogonal Gradient High Order and Low Order methods, and is analytically justified by asymptotic analysis of eigenvalues of Laplace operator on a thin ring and thin shell and comparison with Laplace-Beltrami eigenvalues of a circle and sphere respectively.

Appendix A

Notion of Curvature

Here we provide formal definitions of different types of curvature on curves and surface used in the examples below.

Definition 7 (*Curvature for planar curves*)**.** Given a smooth planar curve Γ, it *curvature* can be defined in one of the several ways listed below, all of which are equivalent [23]:

- the inverse of the radius of *osculating circle*;
- rate of change of *tangent direction*;
- amount of deviation of the curve from its *tangent line*;
- element of area of circular component of *arclength*.

The basic idea of curvature boils down to quantifying amount of deviation of the manifold from its tangent space for an infinitesimal neghborhood of a given point. Although pretty straightforward for planar curves, formalizing this idea for surfaces is slightly more ambiguous, as there could be more than one way to implement it. Definitions below provide analytical as well as geometric insight into the concept.

Definition 8 (*Normal and principal curvatures*)**.** Given two-dimensional smooth manifold

$\mathcal{M}$ embedded into 3D Euclidean space with unit normal $\vec{\boldsymbol{n}}_p$ and some tangent vector $\vec{\boldsymbol{t}}_p$ originating at the same point $p \in \mathcal{M}$, the *normal curvature* $\kappa_{\vec{\boldsymbol{n}}}(\vec{\boldsymbol{t}})$ in the direction $\vec{\boldsymbol{t}}$ is the curvature of the curve obtained by intersecting the manifold with the plane spanning vectors $\vec{\boldsymbol{t}}_p$ and $\vec{\boldsymbol{t}}_p$. Then the *principal curvatures* κ_1 and κ_2 are the *minimum* and *maximum* values of normal curvature $\kappa_{\vec{\boldsymbol{n}}}$ varying $\vec{\boldsymbol{t}}$ respectively.

Definition 9 (*Gaussian and mean curvature*)**.** Given two-dimensional smooth manifold $\mathcal{M}$ embedded into 3D Euclidean space, the *Gaussian curvature* can be defined in one of the two equivalent ways:

- product of *principle curvatures*
- element of area of spherical component of *surface area.*

Similarly, the *mean curvature* can be defined as

- the average of *principle curvatures*
- rate of change of *surface area* under the infinitesimal change of manifold in the normal direction.

References

[1] Eldar Akhmetgaliyev, Chiu-Yen Kao, and Braxton Osting. “Computational methods for extremal Steklov problems”. In: *SIAM Journal on Control and Optimization* 55.2 (2017), pp. 1226–1240.

[2] Diego Álvarez, Pedro González-Rodríguez, and Miguel Moscoso. “A Closed-Form Formula for the RBF-Based Approximation of the Laplace–Beltrami Operator”. In: *Journal of Scientific Computing* (2018), pp. 1–18.

[3] Marcelo Bertalmıo, Li-Tien Cheng, Stanley Osher, and Guillermo Sapiro. “Variational problems and partial differential equations on implicit surfaces”. In: *Journal of Computational Physics* 174.2 (2001), pp. 759–780.

[4] Herbert Buchholz. “Besondere Reihenentwicklungen für eine häufig vorkommende zweireihige Determinante mit Zylinderfunktionen und ihre Nullstellen”. In: *Z Angew Math Mech* 29 (1949), pp. 356–367.

[5] M Buhmann, S De Marchi, and E Perracchione. “Analysis of a new class of rational RBF expansions”. In: *IMA J. Numer. Anal* (2018).

[6] Martin D Buhmann. “Radial basis functions”. In: *Acta Numerica* 9 (2000), pp. 1–38.

[7] Mark AJ Chaplain, Mahadevan Ganesh, and Ivan G Graham. “Spatio-temporal pattern formation on spherical surfaces: numerical simulation and application to solid tumour growth”. In: *Journal of mathematical biology* 42.5 (2001), pp. 387–423.

[8] Isaac Chavel. *Eigenvalues in Riemannian geometry*. Vol. 115. Academic press, 1984.

[9] Yujia Chen and Colin B Macdonald. "The Closest Point Method and multigrid solvers for elliptic equations on surfaces". In: *SIAM Journal on Scientific Computing* 37.1 (2015), A134–A155.

[10] Ka Chun Cheung and Leevan Ling. "A kernel-based embedding method and convergence analysis for surfaces PDEs". In: *SIAM Journal on Scientific Computing* 40.1 (2018), A266–A287.

[11] Ka Chun Cheung, Leevan Ling, and Steven J Ruuth. "A localized meshless method for diffusion on folded surfaces". In: *Journal of Computational Physics* 297 (2015), pp. 194–206.

[12] M Costabel and C Safa. "A boundary integral formulation of antenna problems suitable for nodal-based wavelet approximations". In: *Numerical Mathematics and Advanced Applications*. Springer, 2003, pp. 265–271.

[13] VA Davydov, N Manz, O Steinbock, VS Zykov, and SC Müller. "Excitation fronts on a periodically modulated curved surface". In: *Physical review letters* 85.4 (2000), p. 868.

[14] Tobin A. Driscoll and Rodrigo B. Platte. *Computing Eigenmodes of Elliptical Operators Using Radial Basis Functions*. Tech. rep. Department of Mathematical Sciences, 2003.

[15] Gerhard Dziuk. "Finite elements for the Beltrami operator on arbitrary surfaces". In: *Partial differential equations and calculus of variations* 1357 (1988), pp. 142–155.

[16] Gerhard Dziuk and Charles M Elliott. "Finite element methods for surface PDEs". In: *Acta Numerica* 22 (2013), pp. 289–396.

[17] Gregory Fasshauer and Michael McCourt. *Kernel-based approximation methods using matlab*. Vol. 19. World Scientific Publishing Company, 2015.

[18] Natasha Flyer and Bengt Fornberg. “Radial basis functions: Developments and applications to planetary scale flows”. In: *Computers & Fluids* 46.1 (2011), pp. 23–32.

[19] Natasha Flyer and Grady B Wright. “A radial basis function method for the shallow water equations on a sphere”. In: *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*. The Royal Society. 2009, rspa–2009.

[20] Edward J Fuselier and Grady B Wright. “A high-order kernel method for diffusion and reaction-diffusion equations on surfaces”. In: *Journal of Scientific Computing* 56.3 (2013), pp. 535–565.

[21] Edward Fuselier and Grady B Wright. “Scattered data interpolation on embedded submanifolds with restricted positive definite kernels: Sobolev error estimates”. In: *SIAM Journal on Numerical Analysis* 50.3 (2012), pp. 1753–1776.

[22] Roland Glowinski and Danny C Sorensen. “Computing the eigenvalues of the Laplace-Beltrami operator on the surface of a torus: A numerical approach”. In: *Partial Differential Equations*. Springer, 2008, pp. 225–232.

[23] Ron Goldman. “Curvature formulas for implicit curves and surfaces”. In: *Computer Aided Geometric Design* 22.7 (2005), pp. 632–658.

[24] Jagannathan Gomatam and Faridon Amdjadi. “Reaction-diffusion equations on a sphere: Meandering of spiral waves”. In: *Physical Review E* 56.4 (1997), p. 3913.

[25] HPW Gottlieb. “Eigenvalues of the Laplacian with Neumann boundary conditions”. In: *The ANZIAM Journal* 26.3 (1985), pp. 293–309.

[26] John B Greer, Andrea L Bertozzi, and Guillermo Sapiro. “Fourth order partial differential equations on general geometries”. In: *Journal of Computational Physics* 216.1 (2006), pp. 216–246.

[27] P Grindrod, MA Lewis, and James Dickson Murray. “A geometrical approach to wave-type solutions of excitable reaction-diffusion systems”. In: *Proc. R. Soc. Lond. A* 433.1887 (1991), pp. 151–164.

[28] Lise-Marie Imbert-Gérard and Leslie Greengard. “Pseudo-Spectral Methods for the Laplace-Beltrami Equation and the Hodge Decomposition on Surfaces of Genus One”. In: *Numerical Methods for Partial Differential Equations* 33.3 (2017), pp. 941–955.

[29] Jürgen Jost and Jèurgen Jost. *Riemannian geometry and geometric analysis*. Vol. 42005. Springer, 2008.

[30] Di Kang and Chiu-Yen Kao. “Minimization of inhomogeneous biharmonic eigenvalue problems”. In: *Applied Mathematical Modelling* 51 (2017), pp. 587–604.

[31] Edward J Kansa. “Multiquadrics—A scattered data approximation scheme with applications to computational fluid-dynamics—II surface approximations and partial derivative estimates”. In: *Computers & Mathematics with applications* 19.8-9 (1990), pp. 127–145.

[32] Chiu-Yen Kao, Rongjie Lai, and Braxton Osting. “Maximization of Laplace- Beltrami eigenvalues on closed Riemannian surfaces”. In: *ESAIM: Control, Optimisation and Calculus of Variations* 23.2 (2017), pp. 685–720.

[33] Rongjie Lai, Jiang Liang, and Hongkai Zhao. “A Local Mesh Method for Solving PDEs on Point Clouds.” In: *Inverse Problems & Imaging* 7.3 (2013).

[34] Christoph Landsberg and Axel Voigt. “A multigrid finite element method for reaction-diffusion systems on surfaces”. In: *Computing and visualization in science* 13.4 (2010), pp. 177–185.

[35] Erik Lehto, Varun Shankar, and Grady B Wright. “A radial basis function (RBF) compact finite difference (FD) scheme for reaction-diffusion equations on surfaces”. In: *SIAM Journal on Scientific Computing* 39.5 (2017), A2129–A2151.

[36] Shingyu Leung, John Lowengrub, and Hongkai Zhao. “A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion”. In: *Journal of Computational Physics* 230.7 (2011), pp. 2540–2561.

[37] Shingyu Leung and Hongkai Zhao. “A grid based particle method for evolution of open curves and surfaces”. In: *Journal of Computational Physics* 228.20 (2009), pp. 7706–7728.

[38] Shingyu Leung and Hongkai Zhao. “A grid based particle method for moving interface problems”. In: *Journal of Computational Physics* 228.8 (2009), pp. 2993–3024.

[39] Jian Liang and Hongkai Zhao. “Solving partial differential equations on point clouds”. In: *SIAM Journal on Scientific Computing* 35.3 (2013), A1461–A1486.

[40] Dan Liu, Guoliang Xu, and Qin Zhang. “A discrete scheme of Laplace–Beltrami operator and its convergence over quadrilateral meshes”. In: *Computers & Mathematics with Applications* 55.6 (2008), pp. 1081–1093.

[41] Lok Ming Lui, Xianfeng Gu, Tony F Chan, Shing-Tung Yau, et al. “Variational method on riemann surfaces using conformal parameterization and its applications to image processing”. In: *Methods and Applications of Analysis* 15.4 (2008), pp. 513–538.

[42] Lok Lui, Yalin Wang, and Tony Chan. “Solving PDEs on manifolds with global conformal parametrization”. In: *Variational, Geometric, and Level Set Methods in Computer Vision* (2005), pp. 307–319.

[43] Colin B Macdonald, Jeremy Brandman, and Steven J Ruuth. “Solving eigenvalue problems on curved surfaces using the Closest Point Method”. In: *Journal of Computational Physics* 230.22 (2011), pp. 7944–7956.

[44] Colin B Macdonald, Barry Merriman, and Steven J Ruuth. “Simple computation of reaction–diffusion processes on point clouds”. In: *Proceedings of the National Academy of Sciences* 110.23 (2013), pp. 9209–9214.

[45] Colin B Macdonald and Steven J Ruuth. “Level set equations on surfaces via the Closest Point Method”. In: *Journal of Scientific Computing* 35.2 (2008), pp. 219–240.

[46] Colin B Macdonald and Steven J Ruuth. “The implicit closest point method for the numerical solution of partial differential equations on surfaces”. In: *SIAM Journal on Scientific Computing* 31.6 (2009), pp. 4330–4350.

[47] Nam Mai-Duy, Deepak Dalal, Thi Thuy Van Le, Duc Ngo-Cong, and Thanh Tran-Cong. “A symmetric integrated radial basis function method for solving differential equations”. In: *Numerical Methods for Partial Differential Equations* 34.3 (2018), pp. 959–981.

[48] Niklas Manz, VA Davydov, Stefan C Müller, and Markus Bär. “Dependence of the spiral rotation frequency on the surface curvature of reaction–diffusion systems”. In: *Physics Letters A* 316.5 (2003), pp. 311–316.

[49] James McMahon. “On the roots of the Bessel and certain related functions”. In: *The Annals of Mathematics* 9.1/6 (1894), pp. 23–30.

[50] Emma Naden, Thomas März, and Colin B Macdonald. “Anisotropic Diffusion on Curved Surfaces”. In: *arXiv preprint arXiv:1403.2131* (2014).

[51] Michael O’Neil. “Second-kind integral equations for the Laplace-Beltrami problem on surfaces in three dimensions”. In: *arXiv preprint arXiv:1705.00069* (2017).

[52] Per-Olof Persson and Gilbert Strang. “A simple mesh generator in MATLAB”. In: *SIAM review* 46.2 (2004), pp. 329–345.

[53] Argyrios Petras, Leevan Ling, and Steven J Ruuth. “An RBF-FD closest point method for solving PDEs on surfaces”. In: *Journal of Computational Physics* 370 (2018), pp. 43–57.

[54] Cécile Piret. “The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces”. In: *Journal of Computational Physics* 231.14 (2012), pp. 4662–4675.

[55] Robert C Reilly. “Mean curvature, the Laplacian, and soap bubbles”. In: *The American Mathematical Monthly* 89.3 (1982), pp. 180–198.

[56] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. “Laplace-spectra as fingerprints for shape matching”. In: *Proceedings of the 2005 ACM symposium on Solid and physical modeling*. ACM. 2005, pp. 101–106.

[57] Martin Reuter, Franz-Erich Wolter, and Niklas Peinecke. “Laplace-Beltrami spectra as ‘Shape-DNA’ of surfaces and solids”. In: *Computer-Aided Design* 38.4 (2006), pp. 342–366.

[58] Martin Reuter, Franz-Erich Wolter, Martha Shenton, and Marc Niethammer. “Laplace-Beltrami eigenvalues and topological features of eigenfunctions for statistical shape analysis”. In: *Computer-Aided Design* 41.10 (2009). Selected Papers from the 2007 New Advances in Shape Analysis and Geometric Modeling Workshop, pp. 739–755. ISSN: 0010-4485.

[59] Walter Rudin et al. *Principles of mathematical analysis*. Vol. 3. 4.2. McGraw-hill New York, 1976.

[60] Raif M Rustamov. “Laplace-Beltrami eigenfunctions for deformation invariant shape representation”. In: *Proceedings of the fifth Eurographics symposium on Geometry processing*. Eurographics Association. 2007, pp. 225–233.

[61] Ivo F Sbalzarini, Arnold Hayer, Ari Helenius, and Petros Koumoutsakos. “Simulations of (an) isotropic diffusion on curved biological surfaces”. In: *Biophysical Journal* 90.3 (2006), pp. 878–885.

[62] Peter Schwartz, David Adalsteinsson, Phillip Colella, Adam Paul Arkin, and Matthew Onsum. “Numerical computation of diffusion on a surface”. In: *Proceedings of the National Academy of Sciences of the United States of America* 102.32 (2005), pp. 11151–11156.

[63] Varun Shankar. “The overlapped radial basis function-finite difference (RBF-FD) method: A generalization of RBF-FD”. In: *Journal of Computational Physics* 342 (2017), pp. 211–228.

[64] Pratik Suchde and Joerg Kuhnert. “A Meshfree Generalized Finite Difference Method for Surface PDEs”. In: *arXiv preprint arXiv:1806.07193* (2018).

[65] Greg Turk. “Generating textures on arbitrary surfaces using reaction-diffusion”. In: *ACM SIGGRAPH Computer Graphics*. Vol. 25. 4. ACM. 1991, pp. 289–298.

[66] C Varea, JL Aragon, and RA Barrio. “Turing patterns on a sphere”. In: *Physical Review E* 60.4 (1999), p. 4588.

[67] Guoliang Xu. “Discrete Laplace–Beltrami operators and their convergence”. In: *Computer aided geometric design* 21.8 (2004), pp. 767–784.

[68] Hiroki Yagisita, Masayasu Mimura, and Michio Yamada. “Spiral wave behaviors in an excitable reaction-diffusion system on a sphere”. In: *Physica D: Nonlinear Phenomena* 124.1-3 (1998), pp. 126–136.