Report Number 11/05

Solving Eigenvalue Problems on Curved Surfaces using the Closest Point Method

by

Colid B. MacDonald, Jeremy Brandman, and Steven J. Ruuth

Oxford Centre for Collaborative Applied Mathematics
Mathematical Institute
24 - 29 St Giles'
Oxford
OX1 3LB
England

SOLVING EIGENVALUE PROBLEMS ON CURVED SURFACES USING THE CLOSEST POINT METHOD

COLIN B. MACDONALD*, JEREMY BRANDMAN†, AND STEVEN J. RUUTH‡

Abstract. Eigenvalue problems are fundamental to mathematics and science. We present a simple algorithm for determining eigenvalues and eigenfunctions of the Laplace–Beltrami operator on rather general curved surfaces. Our algorithm, which is based on the Closest Point Method, relies on an embedding of the surface in a higher-dimensional space, where standard Cartesian finite difference and interpolation schemes can be easily applied. We show that there is a one-to-one correspondence between a problem defined in the embedding space and the original surface problem. For open surfaces, we present a simple way to impose Dirichlet and Neumann boundary conditions while maintaining second-order accuracy. Convergence studies and a series of examples demonstrate the effectiveness and generality of our approach.

Key words. eigenvalues, eigenfunctions, Laplace–Beltrami operator, Closest Point Method, surface computation, implicit surfaces.

1. Introduction. The study of eigenvalues and eigenfunctions of the Laplacian operator has long been a subject of interest in mathematics, physics, engineering, computer science and other disciplines. Of considerable importance is the case where the underlying domain is a curved surface, S, in which case the problem becomes one of finding eigenvalues and eigenfunctions of the corresponding Laplace–Beltrami operator

$$-\nabla_{\mathcal{S}} \cdot \nabla_{\mathcal{S}} u = \lambda u, \tag{1.1}$$

or, more generally, the elliptic operator

$$-\nabla_{\mathcal{S}} \cdot (a(x) \nabla_{\mathcal{S}} u) = \lambda u.$$

The Laplace–Beltrami eigenvalue problem has played a prominent role in recent years in data analysis. For example, in [23], eigenvalues of the Laplace–Beltrami operator were used to extract "fingerprints" which characterize surfaces and solid objects. In [2, 6], Laplace–Beltrami eigenvalues and eigenfunctions were used for dimensionality reduction and data representation. Other application areas include smoothing of surfaces [27] and the segmentation and registration of shape [22].

Analytical solutions to the Laplace–Beltrami eigenvalue problem are rarely available, so it is crucial to be able to numerically approximate them in an accurate and efficient manner. Partial differential equations on surfaces, including eigenvalue problems, have traditionally been approximated using either (a) discretizations based on a parameterization of the surface [9], (b) finite element discretizations on a triangulation of the surface [24], or (c) embedding techniques which solve some embedding PDE in a small region near the surface [5] (see also the related works [4, 12, 11, 19, 7]).

*Mathematical Institute, University of Oxford, OX1 3LB, UK (macdonald@maths.ox.ac.uk). The work of this author was supported by an NSERC postdoctoral fellowship, NSF grant number CCF-0321917, and by Award No KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST).

†Department of Mathematics, Courant Institute of Mathematical Sciences, New York University (brandman@cims.nyu.edu). The work of this author was supported by an NSF Mathematical Sciences Postdoctoral Research Fellowship.

‡Department of Mathematics, Simon Fraser University, Burnaby, British Columbia, V5A 1S6 Canada (sruuth@sfu.ca). The work of this author was partially supported by a grant from NSERC Canada.

Parameterization methods (a) are often effective for simple surfaces [9], but for more complicated geometries have the deficiency of introducing distortions and singularities into the method through the parameterization [8]. Approaches based on the finite element method can be deceptively difficult to implement; as described in [24], "even though this method seems to be very simple, it is quite tricky to implement". Embedding methods (c) have gained a considerable following because they permit PDEs on surfaces to be solving using standard finite differences.

This paper proposes a simple and effective embedding method for the Laplace–Beltrami eigenvalue problem based on the *Closest Point Method*. The Closest Point Method is a recent embedding method that has been used to compute the numerical solution to a variety of partial differential equations [25, 16, 15, 17], including in-surface heat flow, reaction-diffusion equations, and higher-order motions involving biharmonic and "surface diffusion" terms. Unlike traditional embedding methods, which are built around level set representatives of the surface, the Closest Point Method is built around a closest point representation of the surface. This allows for general smooth surfaces with boundaries and does not require the surface to have an inside/outside [25]. In addition, the method does not introduce artificial boundary conditions at the edge of the computation band. Such artificial boundary conditions typically lead to low-order accuracy [11].

Here we apply the Closest Point Method to the problem of determining the eigenvalues and eigenmodes of the Laplace–Beltrami operator on a surface. We begin by demonstrating that, for closed surfaces, there is a one-to-one correspondence between the eigenvalues of the embedding problem and the original surface problem. Later, we consider open surfaces and present simple techniques for achieving high-order accurate approximations to Dirichlet and homogeneous Neumann boundary conditions. Our proposed method retains the usual advantages of the Closest Point Method, namely generality with respect to the surface, high-order accuracy and simplicity of implementation.

The paper unfolds as follows. Section 2 provides key background on the Closest Point Method. Section 3 proposes an embedding problem used to solve the Laplace–Beltrami eigenvalue problem and explains why a similar embedding problem leads to spurious eigenvalues. Section 4 provides discretization details. In Section 5, a second-order discretization of boundary conditions is described for open surfaces. Section 6 validates the method with a number of convergence studies and examples on complex shapes. Finally, Section 7 gives a summary and conclusions.

2. The Closest Point Method. We now review the ideas behind the Closest Point Method [25] which are relevant to the problem of finding Laplace-Beltrami eigenvalues and eigenfunctions.

The representation of the underlying surface is fundamental to any numerical method for PDEs on surfaces. The Closest Point Method relies on a closest point representation of the underlying surface.

DEFINITION 1 (Closest point function). *Given a surface $\mathcal{S}$,* $\mathrm{cp}(\boldsymbol{x})$ *refers to a (possibly non-unique) point belonging to $\mathcal{S}$ which is closest to $\boldsymbol{x}$.*

The closest point function, defined in a neighborhood of a surface, gives a representation of the surface. This *closest point representation* allows for general surfaces with boundaries and does not require the surface to have an inside/outside. The surface can be of any codimension [25], or even of mixed codimension [17].

The goal of the Closest Point Method is to replace a surface PDE by a related PDE in the embedding space which can be solved using finite difference, finite element

or other standard methods. In the case of the Laplace-Beltrami eigenvalue problem, this approach relies on the following result, which states that the Laplace–Beltrami operator $\Delta_{\mathcal{S}}$ may be replaced by the standard Laplacian Δ in the embedding space $\mathbb{R}^d$ under certain conditions.

THEOREM 1. *Let $\mathcal{S}$ be a smooth closed surface in $\mathbb{R}^d$ and $u : \mathcal{S} \to \mathbb{R}$ be a smooth function. Assume the closest point function* $\mathrm{cp}(\boldsymbol{x})$ *is defined in a neighborhood $\Omega \subset \mathbb{R}^d$ of $\mathcal{S}$. Then*

$$\Delta_{\mathcal{S}} u(\boldsymbol{x}) = \Delta(u(\mathrm{cp}(\boldsymbol{x}))) \quad \textit{for } \boldsymbol{x} \in \mathcal{S}. \tag{2.1}$$

Note that the right-hand side is well-defined because $u(\mathrm{cp}(\cdot))$ can be evaluated at points both on and off the surface. This result follows from the principles in [25].

Because the function $u(\mathrm{cp}(\boldsymbol{x}))$, known as the closest point extension of u, is used throughout this paper, we make the following definition.

DEFINITION 2 (Closest point extension). *Let $\mathcal{S}$ be a smooth surface in $\mathbb{R}^d$. The* closest point extension *of a function $u : \mathcal{S} \to \mathbb{R}$ to a neighborhood Ω of $\mathcal{S}$ is the function $v : \Omega \to \mathbb{R}$ defined by*

$$v(\boldsymbol{x}) = u(\mathrm{cp}(\boldsymbol{x})). \tag{2.2}$$

3. The embedded eigenfunction problem. Our objective is to develop a simple, effective method for solving the following surface eigenvalue problem:

PROBLEM 1 (Laplace–Beltrami eigenvalue problem). *Given a surface $\mathcal{S}$, determine the* eigenfunctions $u : \mathcal{S} \to \mathbb{R}$ *and* eigenvalues λ *satisfying*

$$-\Delta_{\mathcal{S}}(u(\boldsymbol{x})) = \lambda u(\boldsymbol{x}), \quad \textit{for } \boldsymbol{x} \in \mathcal{S}. \tag{3.1}$$

If the surface $\mathcal{S}$ is open, then the problem will also have boundary conditions: we address this in Section 5.

In this section, we assume that a closest point representation of the surface is available and consider two associated embedding problems. We will see that the first, a direct extension of (2.1), leads to an ill-posed problem. However a straightforward modification of this problem leads to our second embedding problem, which we show is equivalent to (3.1).

3.1. A first try. We first consider the following embedded eigenvalue problem, which is directly motivated by (2.1).

PROBLEM 2 (Ill-posed embedded eigenvalue problem). *Determine the* eigenfunctions $v : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ *and* eigenvalues λ *satisfying*

$$-\Delta(v(\mathrm{cp}(\boldsymbol{x}))) = \lambda v(\boldsymbol{x}), \tag{3.2}$$

in a neighborhood $\Omega \subset \mathbb{R}^d$ *of $\mathcal{S}$.*

Solutions to Problem 1 and Problem 2 are closely related. Every solution to the embedding problem, restricted to the surface, is a solution to the surface problem. Conversely, except for the $\lambda = 0$ case, every surface eigenfunction corresponds to a unique solution of Problem 2. These results are established in Appendix A.

Notably, the one-to-one correspondence between solutions breaks down for the $\lambda = 0$ case (the null-eigenspace). Not only is this case significant in theory, in practice it makes Problem 2 ill-posed.

3.1.1. The null-eigenspace. The constant eigenfunction $u(\boldsymbol{x}) = c$ and $\lambda = 0$ is a solution to (3.1). Now consider a function on Ω which differs from c at a single point $\boldsymbol{y} \notin \mathcal{S}$

$$v(\boldsymbol{x}) = \begin{cases} c, & \text{for } \boldsymbol{x} \in \mathcal{S} \text{ and for } \boldsymbol{x} \in \Omega - \{\boldsymbol{y}\}, \\ c_2, & \text{for } \boldsymbol{x} = \boldsymbol{y}. \end{cases}$$

Note that $v(\boldsymbol{x})$ is a null-eigenfunction of (3.2). Because the point $\boldsymbol{y} \notin \mathcal{S}$ is arbitrary, the set of null-eigenfunctions for (3.2) is much larger than the set of null-eigenfunctions for (3.1). In fact, the set of null-eigenfunctions for (3.2) is infinite-dimensional: any change off the surface gives a new linearly independent eigenfunction.

This example demonstrates the essential flaw of Problem 2: when $\lambda = 0$, only the surface values of eigenfunctions are determined by (3.2) (i.e., eigenfunctions can take on arbitrary values elsewhere). Not surprisingly, the infinite-dimensionality of the null-eigenspace causes problems for numerical methods based on approximating (3.2). For example, a large number of eigenfunctions with near-zero eigenvalues are observed in the numerical experiment of Figure 4.2 in Section 4.3 below.

3.2. The fix: a modified embedded eigenvalue problem. To avoid the null-eigenspace found in Problem 2, we consider a modified embedded eigenvalue problem. Our approach uses the split operator introduced in [17]:

DEFINITION 3 (Operator $\Delta_\varepsilon^\#$). *Given $\Omega \subset \mathbb{R}^d$ containing a surface $\mathcal{S}$ and a function $v : \Omega \to \mathbb{R}$, the operator $\Delta_\varepsilon^\#$ is defined as*

$$\Delta_\varepsilon^\# v(\boldsymbol{x}) := \Delta(v(\mathrm{cp}(\boldsymbol{x}))) - \frac{2d}{\varepsilon^2}\Big[v(\boldsymbol{x}) - v(\mathrm{cp}(\boldsymbol{x}))\Big]. \tag{3.3}$$

where $0 < \varepsilon \ll 1$. The factor $2d$ (twice the dimension of the embedding space Ω) is for later notational convenience. We can view $\Delta_\varepsilon^\# v$ as $\Delta(v(\mathrm{cp}))$ plus a penalty for large change in the normal direction: specifically $\Delta_\varepsilon^\# v$ will be large if $|v(\boldsymbol{x}) - v(\mathrm{cp}(\boldsymbol{x}))|$ is not $O(\varepsilon^2)$. Using this operator, we pose another embedded eigenvalue problem:

PROBLEM 3 (Regularized embedded eigenvalue problem). *Determine all* eigenfunctions $v : \Omega \subset \mathbb{R}^d \to \mathbb{R}$ *and* eigenvalues λ *satisfying*

$$-\Delta_\varepsilon^\# v(\boldsymbol{x}) = \lambda v(\boldsymbol{x}), \tag{3.4}$$

in an embedding space $\Omega \subset \mathbb{R}^d$ *containing the surface $\mathcal{S}$.*

For eigenvalues $\lambda < \frac{2d}{\varepsilon^2}$, we can show a one-to-one correspondence between Problem 1 and Problem 3.

THEOREM 2 (Equivalence of two eigenvalue problems). *Suppose $\mathcal{S}$ is a smooth surface embedded in $\mathbb{R}^d$ and that $\Omega \subset \mathbb{R}^d$ is a neighborhood of the surface. Then, for every eigenfunction $u : C^2(\mathcal{S}) \to \mathbb{R}$ of (3.1) with eigenvalue $\lambda < \frac{2d}{\varepsilon^2}$, there exists a unique eigenfunction $v : \Omega \to \mathbb{R}$ of (3.4) with eigenvalue λ which agrees with u on $\mathcal{S}$. The eigenfunction v is given by*

$$v(\boldsymbol{x}) = \frac{\Delta(u(\mathrm{cp}(\boldsymbol{x}))) + \frac{2d}{\varepsilon^2} u(\mathrm{cp}(\boldsymbol{x}))}{-\lambda + \frac{2d}{\varepsilon^2}}. \tag{3.5}$$

Conversely, for every eigenfunction $v(\boldsymbol{x})$ of (3.4) with eigenvalue λ, the restriction of $v(x)$ to $\mathcal{S}$ is an eigenfunction of (3.1) with eigenvalue λ.

Proof. Existence and uniqueness of v follow directly from the condition that v agrees with u on $\mathcal{S}$ and the definition of $\Delta_\varepsilon^\# v$. The converse follows by choosing $x \in \mathcal{S}$ in (3.4) and applying Theorem 1. ∎

Remark 1. Note that when $\lambda = 0$ in (3.4), v is determined in Ω by its values on the surface, avoiding the null-eigenspace problem of (3.2).

Remark 2. If one tries to compute the eigenvalues $\lambda \geq \frac{2d}{\varepsilon^2}$ of Problem 3, one encounters ill-posedness similar to the ill-posed embedded eigenvalue problem (3.2). This is confirmed by the numerical results in Section 6.1.2. The shift of the null-space by $\frac{2d}{\varepsilon^2}$ is due to the additional term $\frac{2d}{\varepsilon^2}\big(v(\boldsymbol{x}) - v(\mathrm{cp}(\boldsymbol{x}))\big)$ in Problem 3. As we shall see in the next section and Section 6.1.2, this ill-posedness will not be an issue for practical purposes since the parameter ϵ can be chosen proportional to the grid spacing.

EXAMPLE 1. *Let $\mathcal{S}$ be a circle of radius R embedded in 2D and consider the eigenfunction $u = \cos(\sqrt{\lambda}R\theta)$ with eigenvalue λ. Applying Theorem 2, we get*

$$v(\boldsymbol{x}) = \left(\frac{\lambda\varepsilon^2\left(1 - R^2/r^2\right)}{-\lambda\varepsilon^2 + 4} + 1\right)\cos(\sqrt{\lambda}R\theta),$$

and we note that indeed $v(\boldsymbol{x}) = u(\boldsymbol{x})$ on the surface.

3.2.1. Choice of ε. In this work, we choose $\varepsilon = \Delta x$, where Δx is the underlying grid spacing. In two dimensions, with this choice of ε and a centered five-point discretization of the Laplacian operator, we obtain

$$\begin{aligned}\Delta_\varepsilon^{\#} u(x,y) \approx \frac{1}{\Delta x^2}\big[&-4u(x,y) + u(\mathrm{cp}(x+\Delta x, y)) + \\ &u(\mathrm{cp}((x-\Delta x, y)) + u(\mathrm{cp}(x, y+\Delta x)) + u(\mathrm{cp}(x, y-\Delta x))\big], \quad (3.6)\end{aligned}$$

and analogously in higher dimensions. Note that the diagonal term (i.e., the "center" of the finite difference stencil) is without a closest point extension: where diagonal terms appear in the discretization, consistency does not require an interpolation to the closest point on the surface. See [17] for details.

The approximation (3.6) is not completely discrete since, for example, $\mathrm{cp}(x + \Delta x, y)$ is generally not a grid point. In Section 4.3 we will complete this discretization and solve the resulting eigenvalue problem.

4. Numerical Discretization.

In this section, we discretize $\Delta_\varepsilon^{\#} u$ following the approach given in [17]. The eigenvalues of the resulting matrix, computed using standard techniques, approximate the eigenvalues of (3.1).

Following [17], our computational domain is a narrow band of m grid points $L = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m\}$ enveloping the surface $\mathcal{S}$. We approximate a function u at all points in L as the vector $\boldsymbol{u} \in \mathbb{R}^m$ where $u_i \approx u(\boldsymbol{x}_i)$ for each $\boldsymbol{x}_i \in L$.

4.1. Discrete closest point extensions.

To choose the list of points L, we consider the problem of evaluating $u(\mathrm{cp}(\boldsymbol{x}_j))$ for a particular point $\boldsymbol{x}_j$. As shown in Figure 4.1, $\mathrm{cp}(\boldsymbol{x}_j)$ is not typically a grid point, so we approximate the value of $u(\mathrm{cp}(\boldsymbol{x}_j))$ using interpolation of the values of u at neighboring grid points. This interpolation has a certain stencil associated with it [17] and we choose L such that it contains the union of all the required interpolation stencils (an algorithm for doing this is given in [17]).

In this work, we use Barycentric Lagrange polynomial interpolation [3] which is linear in the values of $\boldsymbol{u}$. This allows us to express the closest point extension to any chosen set of points as a multiplication by an *extension matrix* E, where each row of E represents one discrete closest point extension.

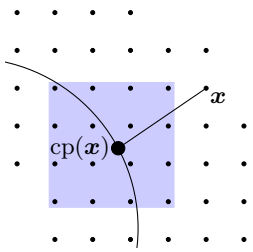
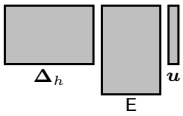


Fig. 4.1: A discrete closest point extension being applied to extend $u(\mathrm{cp}(\boldsymbol{x}))$ using degree $p = 3$ (bicubic) interpolation. The shaded region indicates the interpolation stencil for $\mathrm{cp}(\boldsymbol{x})$.

4.2. Discretizing the Laplace operator. Combining the interpolation step, given by the matrix E, with a standard, symmetric finite difference discretization of Δ (e.g., the standard five-point stencil in 2D or the standard nine-point stencil in 3D), yields a discretization of $\Delta\left(u(\mathrm{cp}(L))\right)$:

$$\Delta\left(u(\mathrm{cp}(L))\right) \approx \boldsymbol{\Delta}_h \mathsf{E}\boldsymbol{u} = \quad = \quad = \widetilde{\mathsf{M}}\boldsymbol{u},$$



where $\widetilde{\mathsf{M}} = \boldsymbol{\Delta}_h \mathsf{E}$ is an $m \times m$ matrix intended to approximate the Laplace–Beltrami operator.

However, the matrix $\widetilde{\mathsf{M}}$ is based on discretizing $\Delta(u(\mathrm{cp}(\cdot)))$ and we know the latter cannot capture the eigenmodes of $\Delta_{\mathcal{S}}$ correctly because of the issue of the null-eigenspace discussed in Section 3.1.1. Thus, it is not surprising to find that the discrete operator $\widetilde{\mathsf{M}}$ also does a poor job of approximating the Laplace–Beltrami operator. For example, Figure 4.2 shows that $\widetilde{\mathsf{M}}$ has many eigenvalues close to zero, including some with imaginary parts and negative real parts. Notably, the matrix $\widetilde{\mathsf{M}}$ is also ill-suited for time-dependent calculations; for example, [17, 15] show that implicit methods built on the matrix $\widetilde{\mathsf{M}}$ have very strict stability time-stepping restrictions and are not competitive with explicit schemes.

4.3. A modified discretization. We will approximate the operator $\Delta_\varepsilon^\#$ using a matrix M as described next. This approach will yield a convergent algorithm for the eigenfunctions and eigenvalues of the Laplace–Beltrami operator $\Delta_{\mathcal{S}}$.

Approximating $\Delta_\varepsilon^\#$ is straightforward and follows from the definition (3.3). When approximating $\Delta_{\mathcal{S}}$ at the point $\boldsymbol{x}_i$ with a finite difference scheme, we map only the neighboring points $\boldsymbol{x}_j$ of the stencil back to their closest points $\mathrm{cp}(\boldsymbol{x}_j)$ and use u_i itself instead of $u(\mathrm{cp}(\boldsymbol{x}_i))$ [17]. This special treatment of the diagonal elements (which corresponds to $\varepsilon = \Delta x$) yields a new $m \times m$ matrix [17]

$$\mathsf{M} := \operatorname{diag}\boldsymbol{\Delta}_h + (\boldsymbol{\Delta}_h - \operatorname{diag}\boldsymbol{\Delta}_h)\mathsf{E} = \quad = \quad + \quad .$$

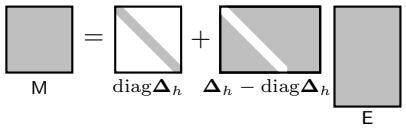


Figure 4.2 gives the results of an experiment which contrasts the behavior of M and $\widetilde{\mathsf{M}}$. We find that the spectrum of M matches that of $\Delta_{\mathcal{S}}$ on a semi-circular domain in $\mathbb{R}^2$, whereas the spectrum of $\widetilde{\mathsf{M}}$ does not.

Remark. The computational band L must be chosen so that it contains the interpolation stencil for $\mathrm{cp}(\boldsymbol{x})$ for all $\boldsymbol{x}$ needed in the calculation. Figure 4.3 shows the computational grid for an egg-shaped curve $\mathcal{S}$ illustrating the list of grid points L. An algorithm for the construction of the list L is given in [17].

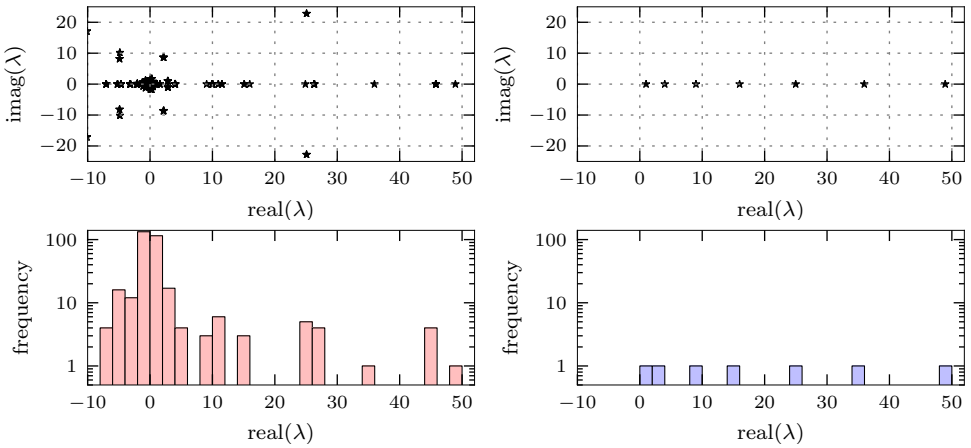


Fig. 4.2: Eigenvalues and histograms of their distribution for the matrices $\widetilde{\mathsf{M}}$ (left) and M (right). The geometry is a unit semi-circle in 2D with homogeneous Dirichlet boundary conditions, E with bicubic interpolation ($p = 3$), and a mesh spacing of $\Delta x = \frac{1}{32}$. Note the large number of eigenvalues near zero for $\widetilde{\mathsf{M}}$, whereas M correctly captures the spectrum of $1, 4, 9, 16, 25, \ldots$.

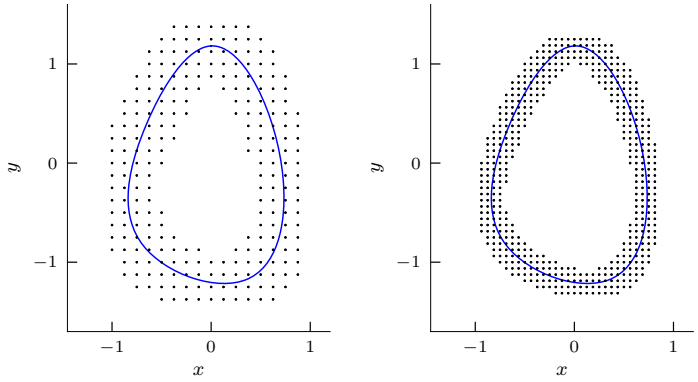


Fig. 4.3: Computational grids for the Closest Point Method for an egg-shaped surface $\mathcal{S}$ with $\Delta x = 0.125$ (left) and $\Delta x = 0.0625$ (right). Second-order finite differences and degree $p = 3$ interpolation.

4.4. Computing eigenvalues and eigenmodes. Given the discrete operator M which approximates the Laplace–Beltrami operator on a surface $\mathcal{S}$, we compute a spectral decomposition

$$\mathsf{M} = \mathsf{Q}\Lambda\mathsf{Q}^{-1}, \tag{4.1}$$

where diag Λ are the eigenvalues of M and the columns of Q are the eigenvectors. These eigenvalues and eigenvectors are the respective approximations of the eigenvalues and eigenfunctions of the Laplace–Beltrami operator on $\mathcal{S}$.

4.4.1. Implementation. The final step of our algorithm—computing the spectral decomposition of a matrix M—is a well-known problem in numerical linear algebra (e.g., [10, 28]). For example, in MATLAB we can use the function eig() to compute the complete decomposition or the function eigs() to compute only part of the spectrum. MATLAB's eig() calls LAPACK routines [1] and eigs() calls ARPACK [14] which makes use of the sparsity of the matrix M (in fact, it only requires a function which returns a matrix-vector product, although in this work we explicitly form the matrix M). Many of our calculations are performed in Python using SciPy [13] and NumPy [20] where we also make use of ARPACK via scipy.sparse.linalg.eigen.arpack.

The eigenfunctions are returned as vectors over the list of points L. In practice, the interpolation scheme from the closest point extension can be reused to evaluate the eigenfunctions at any desired points on the surface for plotting or other purposes.

4.4.2. Degree of interpolation. For the Closest Point Method to achieve the full order of accuracy of the underlying finite difference scheme (say q), the degree of interpolation p must be chosen large enough so that the interpolant itself can be differentiated sufficiently accurately. For the second-order Laplace–Beltrami problems in this work, we need $p \geq q + 1$ [25, 17].

5. Boundary conditions. When applied to an open surface, the Closest Point Method propagates boundary values into the embedding space along directions normal to the boundary, yielding homogeneous Neumann boundary conditions [25]. An analogous method for Dirichlet boundary conditions is similarly straightforward: instead of propagating out the interpolated values at boundary points the prescribed boundary conditions are propagated out [25]. While these methods are simple, they are only first-order accurate, which is lower-order than the discretizations of the Laplace–Beltrami operator discussed in this paper. Fortunately, a simple modification of the closest point function can be introduced to obtain a second-order accurate discretization for boundary conditions of Neumann or Dirichlet type.

Assume a smooth surface, and consider the following modification of the closest point function,

$$\bar{\text{cp}}(\boldsymbol{x}) := \text{cp}\big(\boldsymbol{x} + 2(\text{cp}(\boldsymbol{x}) - \boldsymbol{x})\big) = \text{cp}\big(2\text{cp}(\boldsymbol{x}) - \boldsymbol{x}\big). \tag{5.1}$$

As is illustrated in Figure 5.1, whenever $\text{cp}(\boldsymbol{x})$ is a point in the interior of the surface, the line between $2\text{cp}(\boldsymbol{x}) - \boldsymbol{x}$ and $\text{cp}(\boldsymbol{x})$ is orthogonal to the surface. This implies that $\bar{\text{cp}}(\boldsymbol{x}) = \text{cp}(2\text{cp}(\boldsymbol{x}) - \boldsymbol{x}) = \text{cp}(\boldsymbol{x})$, at least in a neighborhood of the surface. Conversely, if $\bar{\text{cp}}(\boldsymbol{x}_g) \neq \text{cp}(\boldsymbol{x}_g)$ for a point $\boldsymbol{x}_g$ then $\bar{\text{cp}}(\boldsymbol{x}_g)$ is an interior point of the surface[1] and $\text{cp}(\boldsymbol{x}_g)$ is a boundary point. For a straight line or a planar surface, $u(\bar{\text{cp}}(\boldsymbol{x}_g))$ gives the mirror value for $\boldsymbol{x}_g$, while for a general, curved surface it gives an approximate mirror value. In correspondence to the terminology for codimension-zero regions with boundaries, we call a point $\boldsymbol{x}_g$ a *ghost point* if $\text{cp}(\boldsymbol{x}_g) \neq \bar{\text{cp}}(\boldsymbol{x}_g)$ (and note this terminology differs slightly from [17]).

Thus, replacing $\text{cp}(\boldsymbol{x})$ by $\bar{\text{cp}}(\boldsymbol{x})$ in the Closest Point Method does not change the treatment of interior points. At ghost points, however, (approximate) mirror values are used. This yields a second-order approximation of homogeneous Neumann conditions; no other modification of the method is required. Homogeneous Dirichlet boundary conditions are obtained by extending the function at ghost points by

[1] At least for $\boldsymbol{x}_g$ in a neighborhood of a sufficiently well-behaved surface: for example, for $\boldsymbol{x}_g$ far from one boundary of a curve segment, $\text{cp}(2\text{cp}(\boldsymbol{x}_g) - \boldsymbol{x}_g)$ might be another boundary point instead of an interior point.

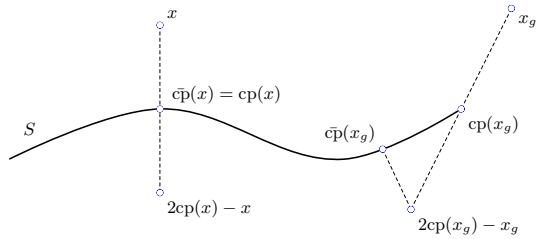


Fig. 5.1: The function $\bar{\text{cp}}$. For the point $\boldsymbol{x}$, the closest point $\text{cp}(\boldsymbol{x})$ lies in the interior of the surface and for such a point, $\bar{\text{cp}}(\boldsymbol{x}) = \text{cp}(\boldsymbol{x})$. However, for $\boldsymbol{x}_g$, the closest point $\text{cp}(\boldsymbol{x}_g)$ lies on the boundary of the surface and $\bar{\text{cp}}(\boldsymbol{x}_g)$ lies in the interior, with $\bar{\text{cp}}(\boldsymbol{x}_g)$ and $\boldsymbol{x}_g$ roughly equidistant from $\text{cp}(\boldsymbol{x}_g)$.

$-u(\bar{\text{cp}}(\boldsymbol{x}_g))$. By analogy to second-order boundary conditions for codimension-zero regions, we observe that an approximation of non-homogeneous Dirichlet conditions is obtained by extending at ghost points by $u(\boldsymbol{x}_g) = 2u(\text{cp}(\boldsymbol{x}_g)) - u(\bar{\text{cp}}(\boldsymbol{x}_g))$, where $u(\text{cp}(\boldsymbol{x}_g))$ is a prescribed value on the boundary.

We emphasize that the previous discussion relates to boundaries of the *surface* and the treatment of boundary conditions imposed thereat, not to the boundary of the narrow band of points (e.g., in Figure 4.3) where no artificial boundary condition need be applied [25, 17].

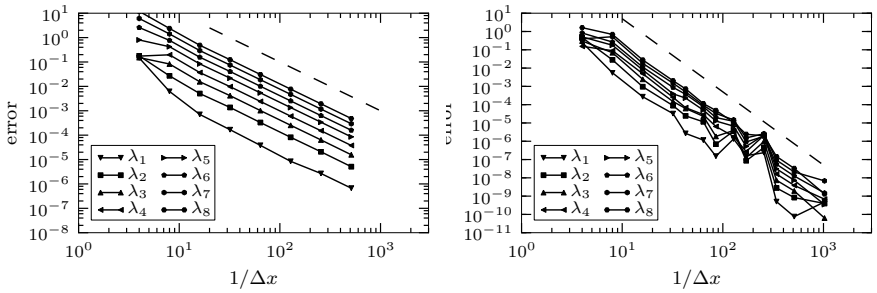
6. Numerical Results. We present a series of numerical examples to demonstrate the effectiveness of our approach.

6.1. Numerical convergence studies. We consider the egg-shaped curve in Figure 4.3 with arclength 2π as a test case. The Laplace–Beltrami eigenfunctions and eigenvalues in this case are the same as those of $u'' = \lambda u$, where u is 2π periodic, namely $u = \cos(ns + \beta)$ and $\lambda = n^2$ for $n \in \mathbb{Z} \geq 0$ where s represents arclength and β is a phase shift. The closest point function for this curve was determined using a numerical optimization procedure based on Newton's method.

Figure 6.1 shows convergence studies in Δx for the first few eigenvalues on the egg-shaped domain. For second-order finite differences and degree-three interpolation we observe second-order convergence. That is, the Closest Point Method approximates the eigenvalues with error $O(\Delta x^2)$. Note that the error increases (and this is true even if one measures relative error) for the larger eigenvalues, but still shows second-order convergence. Figure 6.1 also shows that using fourth-order finite differences and degree-five interpolation, we get fourth-order accurate approximation to the eigenvalues.

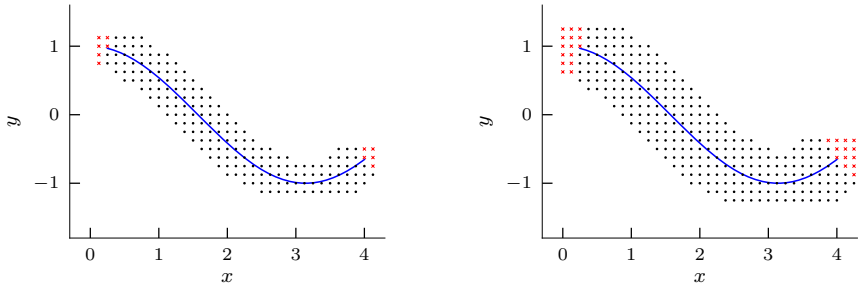
In further numerical tests (not included), we found that degree $p = 2$ interpolation with second-order finite differences, and degree $p = 4$ interpolation with fourth-order finite differences, give approximately second- and fourth-order convergence, respectively. These are better than expected by one order of accuracy, as originally observed in [17].

6.1.1. Boundary Conditions. In this section we verify using convergence studies that our treatment of boundary conditions introduced in Section 5 achieves the expected order of accuracy. For our tests, we use the curve shown in Figure 6.2 parameterized as $(x(t), y(t)) = (t, \cos t)$ for $t \in [1/4, 4]$. We apply homogeneous Neumann and Dirichlet boundary conditions to the ends. Again, the exact eigenvalues and eigenfunctions can be determined analytically by considering the corresponding problem on an interval. For the former, the arclength of the curve is required and can be determined in terms of elliptic integrals.



(a) 2nd-order FD, $p = 3$ interp. (b) 4th-order FD, $p = 5$ interp.

Fig. 6.1: Numerical convergence studies for eight of the Laplace–Beltrami eigenvalues ($\lambda_n = n^2$) of a closed egg-shaped curve in 2D. The dashed reference lines have slope two (left) and four (right). Note second-order accuracy using second-order finite differences with degree $p = 3$ interpolation (left) and fourth-order accuracy using fourth-order finite differences with degree $p = 5$ interpolation (right). The lack of smoothness in the fourth-order results plot may be because the underlying spline curve itself is only C^2 smooth; results on a circle (not included) are smoother.



(a) 2nd-order FD, $p = 3$ interp. (b) 4th-order FD, $p = 5$ interp.

Fig. 6.2: A curve and the corresponding computational grids for the Closest Point Method with second-order (left) and fourth-order (right) finite difference schemes with $\Delta x = 0.125$. Grid points marked with small crosses are the ghost points involved in implementing boundary conditions.

The original Closest Point Method, which does not explicitly treat boundaries, gives a first-order approximation to homogeneous boundary conditions [25]. See, for example, Figure 6.3 where it is observed that this trivial treatment of the boundaries limits the overall accuracy in the eigenvalues to first-order. Figure 6.3 also gives results using the new "$\bar{\text{cp}}$" approach for Neumann boundary conditions described in Section 5; we find that this method attains the second-order accuracy of the underlying Cartesian finite difference scheme. Note that the $\bar{\text{cp}}$ approach is also effective for Dirichlet boundary conditions. For example, in Figure 6.4 we find second-order results for homogeneous Dirichlet boundary conditions using $-u(\bar{\text{cp}}(\boldsymbol{x}_g)$ for ghost points (as in Section 5).

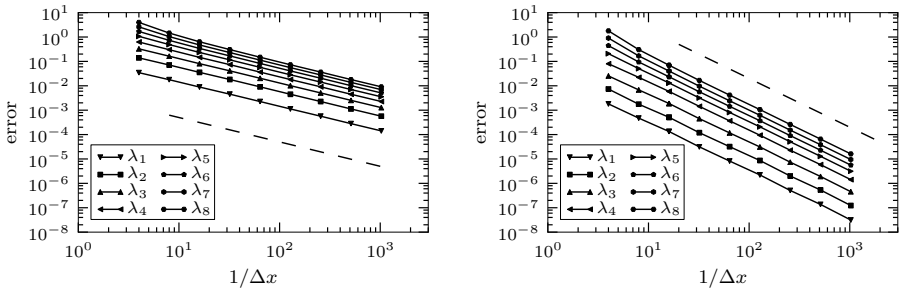


Fig. 6.3: Numerical convergence studies for the first eight non-zero Laplace–Beltrami eigenvalues of a curve in 2D with Neumann boundary conditions applied at both ends. The left figure uses the original closest point function to impose a homogeneous Neumann boundary condition, and we note the results are only first-order accuracy (dashed reference line has slope one). The right figure uses the modified "c̄p" approach and exhibits second-order accuracy (dashed reference line has slope two). Both computations use second-order finite differences and degree $p = 3$ interpolation.

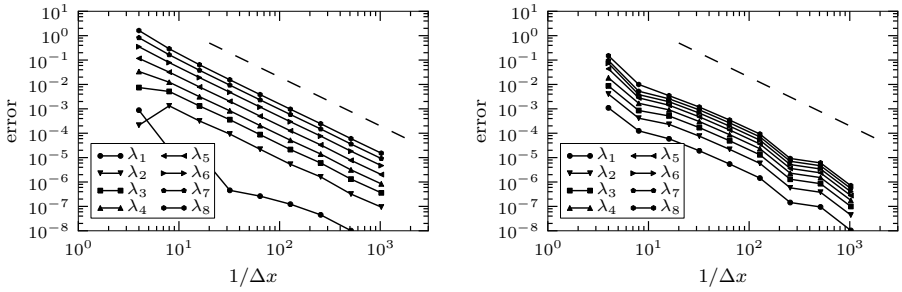


Fig. 6.4: Numerical convergence study for the first few eigenvalues on a curve with Dirichlet boundary conditions applied at both ends. The boundaries are dealt with using the "c̄p" and $-u$ discretization with degree p interpolation. The left figure uses second-order finite differences (and degree $p = 3$ interpolation) and achieves second-order accuracy (the dashed reference lines have slope two). The right figure uses fourth-order finite differences (and degree $p = 5$ interpolation) but the overall accuracy is limited by the second-order treatment of the boundary conditions.

The c̄p approach is designed to give second-order approximations to boundary conditions. Thus it is expected, and observed in Figure 6.4, that even with higher-order finite difference schemes and high-degree interpolation, the results will generally be second-order accurate in the presence of boundary conditions. Third- and higher-order approximation of boundary conditions may also be contemplated; while we do not investigate such methods here we note that approximations of this type will require a replacement for c̄p that incorporates the curvature of the surface near the boundary.

6.1.2. Behavior for high-frequency modes. The large number of spurious eigenvalues near $\frac{2d}{\Delta x^2}$ in Figure 6.5 reflect the singular behavior of Equation (3.5) at $\lambda = \frac{2d}{\Delta x^2}$ (c.f., Figure 4.2 where the problem instead occurs near the origin).

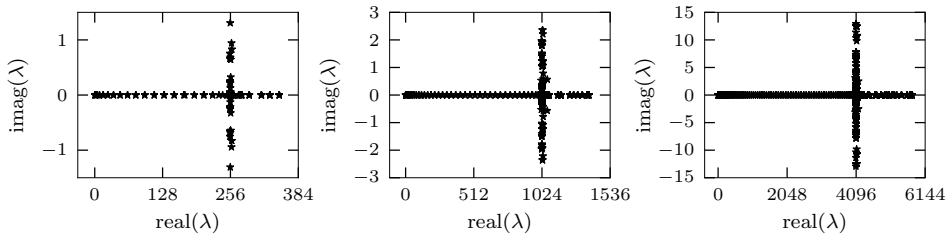


Fig. 6.5: Computed spectra of the Laplace–Beltrami operator on a unit circle using $\Delta x = \frac{1}{8}, \frac{1}{16}$, and $\frac{1}{32}$ (from left-to-right). The method uses second-order finite differences and degree $p = 3$ interpolation. Note a large number of extraneous complex eigenvalues near $\frac{4}{\Delta x^2}$ (i.e., 256, 1024, and 4096): these correspond to the singularity in (3.5) and can be controlled by further resolving Δx.

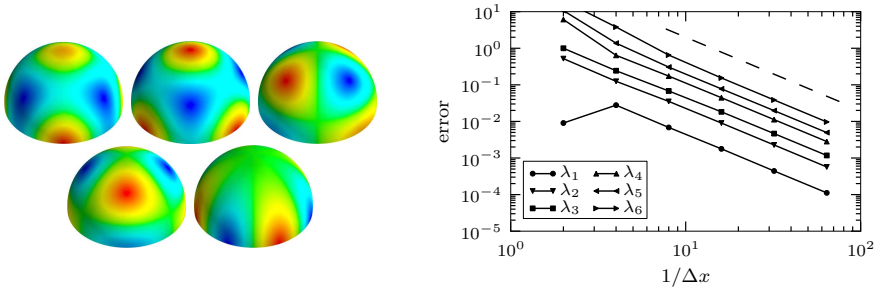


Fig. 6.6: The left image shows several hemispherical harmonics with eigenvalue 20 computed with second-order finite differences, degree $p = 3$ interpolation, and $\Delta x = \frac{1}{64}$. The right image is a convergence study for the first six eigenvalues ($\lambda_n = n(n+1)$), also using second-order finite differences and degree $p = 3$ interpolation (the dashed reference line has slope two).

Notably, it is these spurious eigenvalues which possess nonzero imaginary components, reflecting the fact that the eigenvalue problem (3.1) is not self-adjoint. Because these spurious eigenvalues correspond to highly oscillatory eigenfunctions which are quite close to the Nyquist frequency (i.e., with eigenvalues near $\frac{\pi^2}{\Delta x^2}$), it is not surprising that numerical difficulties arise for such modes. Nonetheless, any particular higher-frequency modes may be resolved by appropriately refining Δx.

6.2. Eigenfunction computations. The following computations were performed in Matlab and SciPy [13]. Visualizations were carried out with Matlab, VTK [26] and MayaVi [21].

6.2.1. Hemispherical harmonics. Figure 6.6 shows several Laplace–Beltrami eigenmodes of a unit hemisphere with homogeneous Neumann boundary conditions on the equator.

6.2.2. Eigenkaninchen. Figure 6.7 shows Laplace–Beltrami eigenmodes of the surface of the Stanford Bunny [29]. The closest point function for this and other triangulated surfaces can be computed in an efficient, straightforward manner [16].

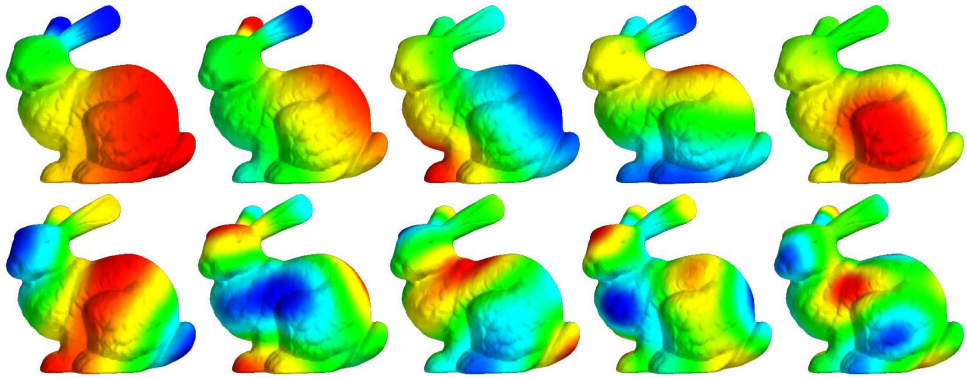


Fig. 6.7: A selection of eigenkaninchen: Laplace–Beltrami eigenmodes on the surface of the Stanford Bunny [29]. The results are computed via the Closest Point Method using second-order finite differences, degree $p = 3$ interpolation, and $\Delta x = 0.1$, where the bunny is roughly two units long.

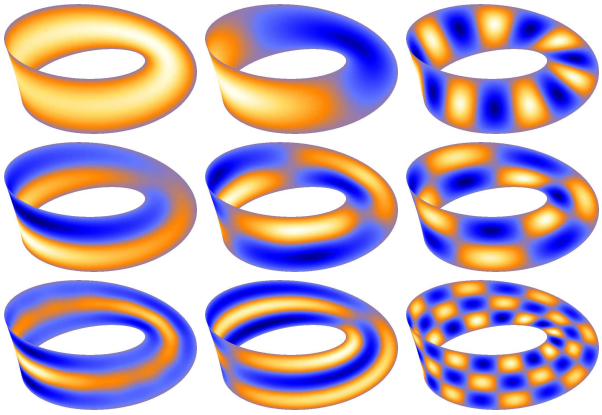


Fig. 6.8: A selection of Laplace–Beltrami eigenmodes of a Möbius strip computed with the Closest Point Method. Calculations use second-order finite differences, $p = 3$, $\Delta x = 0.1$, and the Möbius strip is about 2 units in "diameter".

6.2.3. Möbius strip. Figure 6.8 shows some Laplace–Beltrami eigenmodes of a Möbius strip. Dirichlet boundary conditions are applied at the boundary of the Möbius strip. The surface is embedded in $\mathbb{R}^3$ and the closest point function for each grid point is computed from a parameterization using a numerical optimization procedure minimizing the square of the distance function. Note that the non-orientable nature of the Möbius strip poses no difficultly for the Closest Point Method.

6.2.4. L-shaped Domain. The Closest Point Method works on surfaces of various codimension [25, 17] and indeed solid shapes in $\mathbb{R}^2$ or $\mathbb{R}^3$ are surfaces of codimension-zero. Figure 6.9 shows an eigenmode computation on an L-shaped domain, where zero Dirichlet boundary conditions are imposed using the "$\bar{\text{cp}}$" technique described in Section 5. In the interior of a solid, $\text{cp}(\boldsymbol{x}) = \boldsymbol{x}$ and so no interpolations are needed. Furthermore, for a grid-aligned L-shaped domain, for any $\boldsymbol{x}_i$ outside the domain, $\bar{\text{cp}}(\boldsymbol{x}_i)$ turns out to be another grid point $\boldsymbol{x}_j$ (located inside the domain as if the perimeter were a mirror) so no interpolation step is necessary. Thus in this special "corner case", the Closest Point Method reduces to a standard finite difference computation using ghost points to mirror the values along the perimeter. Interestingly,

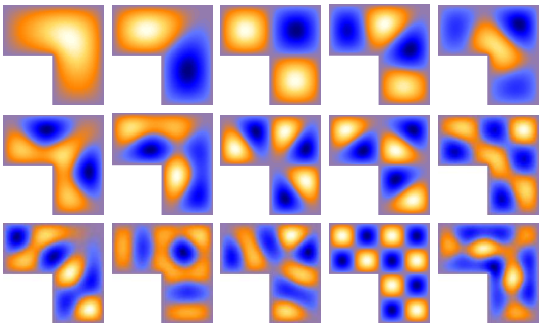


Fig. 6.9: The first 15 Laplacian eigenmodes of an L-shaped domain with Dirichlet boundary conditions. The results are computed using the Closest Point Method with second-order finite differences using $\Delta x = 1/40$ (the domain is two units wide).

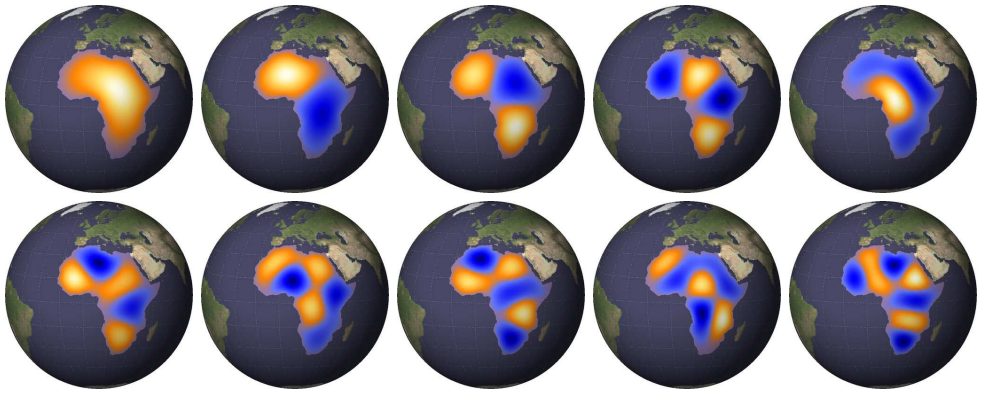


Fig. 6.10: First ten eigenmodes of continental Africa. The Earth has unit diameter and the computation uses second-order finite differences with $\Delta x = 1/40$ and degree $p = 3$ interpolation.

these reductions happen automatically: no change in the code is needed.

6.2.5. Continental Africa. Some Laplace–Beltrami eigenmodes of continental Africa are shown in Figure 6.10. The results are computed directly on the surface of the Earth (assumed to be a sphere). Homogeneous Dirichlet boundary conditions are applied to the coastline. Finding the closest point function involves projecting onto a bitmapped image of the Earth [18] where the continent was first manually segmented. It is interesting to note that these eigenmodes match very closely the first ten eigenmodes of the L-shaped domain in Figure 6.9.

7. Summary and Conclusions. Through a series of convergence studies and computational examples, we have shown that the Closest Point Method is an effective method for computing the spectra and eigenfunctions of the Laplace–Beltrami operator on rather general surfaces. The basis of our approach is the embedded eigenvalue problem (3.4) which, when discretized using standard, centered finite difference methods and Lagrange interpolation, yields a nonsymmetric matrix eigenvalue problem which can be solved using standard software. Fortunately, this lack of symmetry is not a concern in many practical situations since only the highest frequency modes have a significant imaginary component. We are currently investigating a finite element Closest Point Method which would lead to symmetric matrices.

For eigenvalue problems on open surfaces with Dirichlet or homogeneous Neu-

mann boundary conditions, we have introduced second-order accurate schemes. These methods require only a simple change to the closest point extension and are straightforward to implement.

Although we have focused on the Laplace–Beltrami operator in this work, the Closest Point Method is applicable to many other surface differential operators [25, 16, 15, 17]. This suggests that the approach presented here may be applicable to a larger class of eigenvalue problems.

Acknowledgments. The authors thank Bin Dong (UCSD), who motivated this work by asking if the Closest Point Method could be used for surface eigenvalue calculations.

Appendix A. Additional theorems. As mentioned in Section 3.1, every solution to the ill-posed embedding problem (3.2), restricted to the surface, is a solution to the surface eigenvalue problem. Conversely, except for the $\lambda = 0$ case, every surface eigenfunction corresponds to a unique solution of the embedding problem. These results are established by the following theorems.

THEOREM 3. *Suppose $v(\boldsymbol{x})$ and λ are a solution to the embedding eigenvalue problem (3.2) and $\mathcal{S}$ is a smooth surface. Then $u : \mathcal{S} \to \mathbb{R}$ defined by $u(\boldsymbol{x}) = v(\boldsymbol{x})$ for $\boldsymbol{x} \in \mathcal{S}$ is an eigenfunction of (3.1) with eigenvalue λ.*

Proof. This is a direct consequence of Theorem 1. □

THEOREM 4. *Let $\mathcal{S}$ be a smooth surface. For every non-constant solution u and λ of (3.1), there exists a unique (up to a multiplicative constant) eigenfunction v of (3.2) with eigenvalue λ which agrees with u on $\mathcal{S}$. This eigenfunction is given by $v(\boldsymbol{x}) = -\frac{1}{\lambda}\Delta(u(\text{cp}(\boldsymbol{x})))$.*

Proof. Using our hypothesis that v agrees with u on $\mathcal{S}$, we may solve for v in (3.2) to obtain the result. □

REFERENCES

[1] E. ANDERSON, Z. BAI, C. BISCHOF, S. BLACKFORD, J. DEMMEL, J. DONGARRA, J. DU CROZ, A. GREENBAUM, S. HAMMARLING, A. MCKENNEY, AND D. SORENSEN, *LAPACK Users' Guide*, SIAM, third ed., 1999.

[2] M. BELKIN AND P. NIYOGI, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Computation, 15 (2003), pp. 1373–1396.

[3] J.-P. BERRUT AND L. N. TREFETHEN, *Barycentric Lagrange interpolation*, SIAM Rev., 46 (2004), pp. 501–517.

[4] M. BERTALMÍO, L.-T. CHENG, S. OSHER, AND G. SAPIRO, *Variational problems and partial differential equations on implicit surfaces*, J. Comput. Phys., 174 (2001), pp. 759–780.

[5] J. BRANDMAN, *A level-set method for computing the eigenvalues of elliptic operators defined on compact hypersurfaces*, J. Sci. Comput., 37 (2008), pp. 282–315.

[6] R. R. COIFMAN AND S. LAFON, *Diffusion maps*, Applied and Computational Harmonic Analysis, 21 (2006), pp. 5–30.

[7] G. DZIUK AND C. M. ELLIOTT, *Eulerian finite element method for parabolic PDEs on implicit surfaces*, Interf. Free Bound., 10 (2008), pp. 119–138.

[8] M. FLOATER AND K. HORMANN, *Surface parameterization: a tutorial and survey*, in Advances in Multiresolution for Geometric Modelling, Springer, 2005, pp. 157–186.

[9] R. GLOWINSKI AND D. C. SORENSEN, *Computing the eigenvalues of the Laplace–Beltrami operator on the surface of a torus: A numerical approach*, in Partial Differential Equations, E. Oñate, R. Glowinski, and P. Neittaanmäki, eds., vol. 16 of Computational Methods in Applied Sciences, Springer Netherlands, 2008, pp. 225–232.

[10] G. H. GOLUB AND C. F. VAN LOAN, *Matrix Computations*, Johns Hopkins University Press, 1996.

[11] J. B. GREER, *An improvement of a recent Eulerian method for solving PDEs on general geometries*, J. Sci. Comput., 29 (2006), pp. 321–352.

[12] J. B. Greer, A. L. Bertozzi, and G. Sapiro, *Fourth order partial differential equations on general geometries*, J. Comput. Phys., 216 (2006), pp. 216–246.
[13] E. Jones, T. Oliphant, P. Peterson, et al., *SciPy: Open source scientific tools for Python*. http://www.scipy.org, 2001. Accessed 2008-07-15.
[14] R. Lehoucq, D. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, 1998.
[15] C. B. Macdonald, *The Closest Point Method for time-dependent processes on surfaces*, PhD thesis, Simon Fraser University, August 2008.
[16] C. B. Macdonald and S. J. Ruuth, *Level set equations on surfaces via the Closest Point Method*, J. Sci. Comput., 35 (2008), pp. 219–240.
[17] ———, *The implicit Closest Point Method for the numerical solution of partial differential equations on surfaces*, SIAM J. Sci. Comput., 31 (2009), pp. 4330–4350.
[18] NASA, *Visible Earth website*. http://visibleearth.nasa.gov, 2010. Accessed 2010-12-05.
[19] O. Nemitz, M. B. Nielsen, M. Rumpf, and R. Whitaker, *Finite element methods on very large, dynamic tubular grid encoded implicit surfaces*, SIAM J. Sci. Comput., (submitted).
[20] T. E. Oliphant, *Python for scientific computing*, Computing in Science & Engineering, 9 (2007), pp. 10–20.
[21] P. Ramachandran and G. Varoquaux, *Mayavi: Making 3D data visualization reusable*, in Proceedings of the 7th Python in Science Conference, G. Varoquaux, T. Vaught, and J. Millman, eds., 2008, pp. 51–56. http://code.enthought.com/projects/mayavi/.
[22] M. Reuter, S. Biasotti, D. Giorgi, G. Patanè, and M. Spagnuolo, *Discrete Laplace–Beltrami operators for shape analysis and segmentation*, Computers & Graphics, 33 (2009), pp. 381–390. IEEE International Conference on Shape Modelling and Applications 2009.
[23] M. Reuter, F.-E. Wolter, and N. Peinecke, *Laplace-spectra as fingerprints for shape matching*, in Proceedings of the 2005 ACM symposium on Solid and physical modeling, SPM '05, 2005, pp. 101–106.
[24] ———, *Laplace–Beltrami spectra as 'Shape-DNA' of surfaces and solids*, Computer-Aided Design, 38 (2006), pp. 342–366.
[25] S. J. Ruuth and B. Merriman, *A simple embedding method for solving partial differential equations on surfaces*, J. Comput. Phys., 227 (2008), pp. 1943–1961.
[26] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit: an object-oriented approach to 3D graphics*, Prentice Hall, 1998.
[27] S. Seo, M. Chung, and H. Vorperian, *Heat kernel smoothing using Laplace–Beltrami eigenfunctions*, in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2010, T. Jiang, N. Navab, J. Pluim, and M. Viergever, eds., vol. 6363 of Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2010, pp. 505–512.
[28] L. N. Trefethen and D. Bau, *Numerical Linear Algebra*, SIAM, 1997.
[29] G. Turk and M. Levoy, *The Stanford Bunny, the Stanford 3D scanning repository*. http://www-graphics.stanford.edu/data/3Dscanrep, 1994. Accessed 2008-06-18.

RECENT REPORTS

48/10	Possible role of differential growth in airway wall remodeling in asthma	Moulton Goriely
49/10	Variational Data Assimilation Using Targetted Random Walks	Cotter Dashti Robinson Stuart
50/10	A model for the anisotropic response of fibrous soft tissues using six discrete fibre bundles	Flynn Rubin Nielsen
51/10	STOCHSIMGPU Parallel stochastic simulation for the Systems Biology Toolbox 2 for MATLAB	Klingbeil Erban Giles Maini
52/10	Order parameters in the Landau-de Gennes theory - the static and dynamic scenarios	Majumdar
53/10	Liquid Crystal Theory and Modelling Discussion Meeting	Majumdar Mottram
54/10	Modeling the growth of multicellular cancer spheroids in a bioengineered 3D microenvironment and their treatment with an anticancer drug	Loessner Flegg Byrne Hall Moroney Clements Hutmacher McElwain
55/10	Scalar Z, ZK, KZK, and KP equations for shear waves in incompressible solids	Destrade Goriely Saccomandi
56/10	The Influence of Bioreactor Geometry and the Mechanical Environment on Engineered Tissues	Osborne ODea Whiteley Byrne Waters
57/10	A numerical guide to the solution of the bidomain equations of cardiac electrophysiology	Pathmanathan Bernabeu Bordas Cooper Garny Pitt-Francis Whiteley Gavaghan
58/10	Particle-scale structure in frozen colloidal suspensions from small angle X-ray scattering	Spannuth Mochrie Peppin Wettlaufer
59/10	Spin coating of an evaporating polymer solution	Munch Please Wagner

62/10	Adsorption and desorption dynamics of citric acid anions in soil	Oburger Leitner Jones Zygalakis Schnepf Roose
63/10	A dual porosity model of nutrient uptake by root hairs soil	Zygalakis Kirk Jones Roose Wissuwa
64/10	Hot Charge Pairs and Charge Generation in Donor Acceptor Blends	Kirkpatrick
65/10	Excluded-volume effects in the diffusion of hard spheres	Bruna Chapman
66/10	Dynamics of colloidal particles in ice	Spannuth Mochrie Peppin Wettlaufer
01/11	Improving the efficiency of optical coherence tomography by using the non-ideal behaviour of a polarising beam splitter	Lippok Nielsen Vanholsbeeck
02/11	Self-diffusion in remodelling and growth	Epstein Goriely
03/11	Spontaneous rotational inversion in Phycomyces	Goriely Tabor
04/11	From individual to collective behaviour of coupled velocity jump processes: a locust example	Erban Haskovec

Copies of these, and any other OCCAM reports can be obtained from:

Oxford Centre for Collaborative Applied Mathematics
Mathematical Institute
24 - 29 St Giles'
Oxford
OX1 3LB
England
www.maths.ox.ac.uk/occam