A Closest Point Penalty Method for Evolution Equations on Surfaces

Ingrid von Glehn
Oriel College
University of Oxford

A thesis submitted for the degree of
Doctor of Philosophy

Trinity 2014

Abstract

This thesis introduces and analyses a numerical method for solving time-dependent partial differential equations (PDEs) on surfaces. This method is based on the closest point method, and solves the surface PDE by solving a suitably chosen equation in a band surrounding the surface. As it uses an implicit closest point representation of the surface, the method has the advantages of being simple to implement for very general surfaces, and amenable to discretization with a broad class of numerical schemes.

The method proposed in this work introduces a new equation in the embedding space, which satisfies a key consistency property with the surface PDE. Rather than alternating between explicit time-steps and re-extensions of the surface function as in the original closest point method, we investigate an alternative approach, in which a single equation can be solved throughout the embedding space, without separate extension steps. This is achieved by creating a modified embedding equation with a penalty term, which enforces a constraint on the solution.

The resulting equation admits a method of lines discretization, and can therefore be discretized with implicit or explicit time-stepping schemes, and analysed with standard techniques. The method can be formulated in a straightforward way for a large class of problems, including equations featuring variable coefficients, higher-order terms or nonlinearities.

The effectiveness of the method is demonstrated with a range of examples, drawing from applications involving curvature-dependent diffusion and systems of reaction-diffusion equations, as well as equations arising in PDE-based image processing on surfaces.

Acknowledgements

I would like to thank my supervisor, Colin Macdonald, for his guidance, encouragement, and especially for his contagious enthusiasm for scientific computing. My thanks also to Tom März for invaluable support and for many thought-provoking discussions. I have really enjoyed my time working as a member of this group.

I am grateful to Endre Süli and Andy Wathen for their feedback on my transfer and confirmation of status. Thank you to Almut Eisenträger and Tamara von Glehn for reading parts of the thesis, and in particular to Iain Smears for many helpful discussions and suggestions.

Thanks to everyone in OCCAM, the Numerical Analysis Group, and Oriel College, for being wonderful. Finally, thank you to my family for their support and encouragement.

This work of this thesis was partially supported by Award No KUK-C1-013-04 made by King Abdullah University of Science and Technology (KAUST), and by an Oriel College Graduate Scholarship.

Contents

Chapter 1

Introduction

Partial differential equations (PDEs) defined on curved surfaces appear in many diverse physical and biological systems and applications. Examples include fluid flow on surfaces [MCC02], the diffusion of chemicals in cell growth and morphogenesis [Mur03], and texture mapping in computer graphics [Tur91].

The numerical solution of these equations and treatment of surface differential operators is an area of active research. Some methods work directly on the surface, using either a parameterization (for a survey, see [FH05]), or a triangulation of the surface [DE07b, DMS13, LXZ08]. An alternative approach is to use an embedding (Eulerian) method, in which a related equation is solved in the space surrounding the surface. The solution of this equation is then restricted to the surface to obtain a solution of the original PDE. Eulerian methods which use a level set representation of the surface include the framework of [BCOS01, Gre06] for variational problems, and finite element methods on implicit surfaces [Bur09, DDEH10]. A different embedding approach is the closest point method [RM08, MR09], which instead represents the surface using a closest point map, making it more general with respect to surface geometry and codimension.

1.1 Overview

This thesis is based on the closest point method, a simple embedding method for the numerical solution of PDEs on surfaces. It is applicable to a large class of surface geometries, and is straightforward to implement using standard well-studied numerical techniques on Cartesian grids [RM08]. It has been applied to a variety of problems, including eigenvalue problems [MBR11], in-surface interface motion [MR08], and fluid effects on surfaces [AMT+12, KTT13].

The closest point method uses an implicit representation of the surface, and is based on formulating an embedding equation for a function in the ambient space surrounding the surface. It is required that the embedding equation should be consistent with the surface PDE, in the sense that a solution of this embedding equation, when restricted to the surface, is a solution of the original surface PDE. In this work, we propose a new embedding equation, for which we can show that this consistency property is satisfied, with the additional property of admitting a method of lines discretization, thus allowing the use of higher-order time-stepping schemes.

To specify the method, we first introduce a framework for mapping between functions on the surface and in the embedding space, based on linear extension and restriction operators. We then use these operators to derive the extended closest point equation for a given time-dependent PDE, and show that there is a one-to-one correspondence between solutions of the extended and original surface equations. This consistency is enforced by a constraint on the solution of the embedding equation. The new equation is simple to discretize and solve numerically using a standard method of lines approach. This approach generalizes the stabilized implicit closest point method of [MR09], and has the advantage that it can be adapted to a very general class of problems.

The closest point method has increasingly found applications in image processing and graphics, such as fire simulation [HZQW10] and image segmentation [TMR09]. As an application of the method introduced in this work, we turn to problems in PDE-based image processing. These PDE models generalize image processing methods for planar images to the setting of images or textures defined on surfaces. We consider standard examples from this field, such as image denoising and image inpainting, and demonstrate the simplicity and effectiveness of the method.

1.1.1 The closest point method

Suppose we want to solve a time-dependent PDE defined on a curved surface. The closest point method [RM08] uses the fact that the surface is embedded in $\mathbb{R}^n$, and represents the surface by a retraction based on Euclidean distance. For every point x in this surrounding space $\mathbb{R}^n$, the retraction returns a surface point which is closest to x (see Figure 1.1). We call this retraction a closest point function, denoted by cp.

A function defined on the surface can be extended off the surface into the surrounding space by assigning to each point x the value of the surface function at $\text{cp}(x)$. The key observation is that this function is now constant in the direction normal to the surface. It will be shown that surface gradients and surface divergences of the

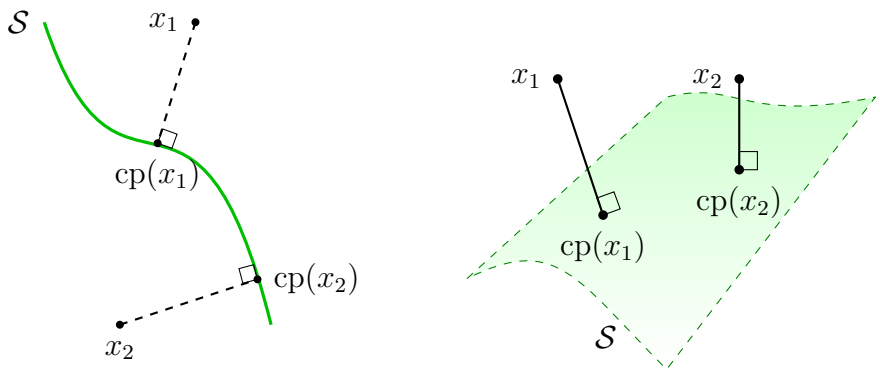


Figure 1.1: Closest point functions for surfaces embedded in $\mathbb{R}^2$ and $\mathbb{R}^3$.

original function agree with the standard Cartesian operators of the extended function at the surface [MM12]. We call these ideas the "gradient principle" and "divergence principle" [RM08]. These principles are used to derive a simpler analogous PDE problem in the embedding space (for example, replacing surface intrinsic diffusion with the bulk or *Cartesian* diffusion). More general closest point functions are introduced in [MM12], using notions other than Euclidean distance to determine the mapping between the surrounding space and the surface.

We introduce the method of [RM08] with a simple example. Let $\mathcal{S}$ be a smooth compact surface without boundary embedded in $\mathbb{R}^n$, and let $u(t, y)$ be a C^2 scalar function on $\mathcal{S}$. Consider the surface diffusion equation,

$$u_t(t, y) = \Delta_{\mathcal{S}} u(t, y), \quad y \in \mathcal{S} \tag{1.1}$$

subject to an initial condition $u(0, y) = u_0(y)$.

For a suitably chosen neighbourhood of the surface $B(\mathcal{S})$ in $\mathbb{R}^n$ (referred to as the band, see Figure 1.2), we define a smooth *closest point function* $\mathrm{cp} : B(\mathcal{S}) \to \mathcal{S}$, which maps points in the band to points on the surface. Typically, this will be the point closest in Euclidean distance on the surface, but this may be made more general in certain cases [MM12], provided that the manifolds of constant cp intersect the surface orthogonally. Thus if we assign to each point x in $B(\mathcal{S})$ the value of the function u at the surface point $\mathrm{cp}(x)$, then the gradient of the resulting function $u(\mathrm{cp}(x))$ will be tangent at the surface.

The surface differential operator (Laplace–Beltrami operator) may then be replaced at the surface by a standard (Cartesian) Laplacian, so that

$$u_t(t, y) = \Delta[u(t, \mathrm{cp}(y))], \quad y \in \mathcal{S}. \tag{1.2}$$

We consider now a function $\bar{u}$, defined in the whole of $B(\mathcal{S})$, and write down an analogous PDE in the surrounding space:

$$\bar{u}_t(t,x) = \Delta[\bar{u}(t,x)], \quad x \in B(\mathcal{S}), \tag{1.3}$$

where $\bar{u}_0(x) = u_0(\mathrm{cp}(x))$ is obtained through an extension of the initial surface function u. This new equation involves no surface differential operators, and can be approximated using existing numerical discretizations. Note however, that the closest point principles hold only for points on the surface. If this analogous PDE (1.3) is solved throughout the embedding space, the restriction of the solution to the surface may no longer be a solution of the original equation. The approach of the explicit closest point formulation of [RM08] is to compute the numerical solution at the next time step using an explicit time-stepping scheme, before a re-extension of the function $\bar{u}$ is performed to correct the values away from the surface. The resulting semi-discrete scheme can then be expressed as a two-step explicit method in the form of [RM08, MR08]: for example, with forward Euler time-stepping, we compute the intermediate step

$$\tilde{\bar{u}}^{n+1}(x) = \bar{u}^n(x) + \Delta t \Delta \bar{u}^n(x), \quad x \in B(\mathcal{S}), \tag{1.4}$$

followed by a re-extension of the values on the surface,

$$\bar{u}^{n+1}(x) = \tilde{\bar{u}}^{n+1}(\mathrm{cp}(x)), \quad x \in B(\mathcal{S}), \tag{1.5}$$

where $\bar{u}^n$ denotes the approximation to the solution at time step $t^n = n\Delta t$ for all $n \geq 0$. We note that this approach is not a method of lines, so the standard analysis of time-stepping schemes does not apply.

An implicit version of the closest point method is described in [MR09], allowing application to stiff problems, such as those involving biharmonic or higher-order operators. To ensure stability, this formulation includes a stabilizing term, which can be related to the approach of this thesis (see Section 3.4.1).

1.1.2 A constrained formulation

The requirement that the extended PDE should be consistent with the surface equation can be enforced by a constraint on the solution of the equation. In this work, we propose a constrained formulation of the extended equation as follows.

The solution of a given time-dependent surface PDE is a function u, which is defined only for points on the surface. We consider again a function v which is

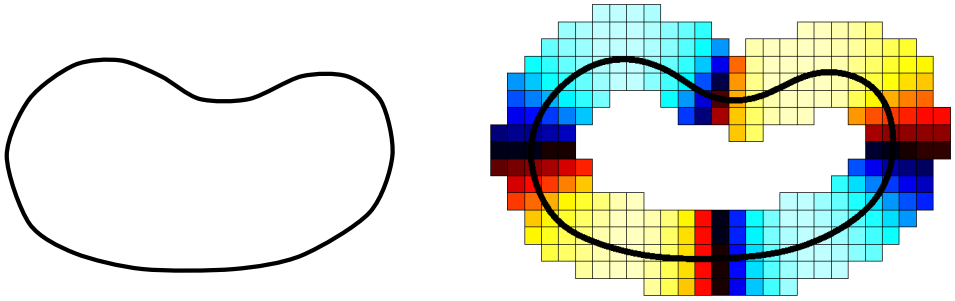


Figure 1.2: Calculations are performed in a narrow band surrounding the surface. The figure shows a surface embedded in $\mathbb{R}^2$, and values of a function defined on a Cartesian grid in the narrow band.

defined for all points in a band $B(\mathcal{S})$ surrounding the surface, and require $v|_{\mathcal{S}}$ to be a solution of the surface PDE. Based on the extension operators defined in Chapter 2, we formulate an equation for v which is now consistent with the original surface equation in all of $B(\mathcal{S})$,

$$v_t(t,x) = E\Delta v(t,x), \tag{1.6}$$

subject to the constraint

$$v(t,x) = Ev(t,x). \tag{1.7}$$

Here, the extension operator E is defined by $Ev := v(t, \mathrm{cp}(x))$ (see Section 2.3). The constraint or side condition that v is a closest point extension is enforced by adding a penalty term to the equation, resulting in a single equation

$$v_t(t,x) = E\Delta v(t,x) - \gamma(v(t,x) - Ev(t,x)), \quad x \in B(\mathcal{S}),\ t \in (0,T), \tag{1.8}$$

where $\gamma \in \mathbb{R}$ is a penalty parameter. We will show that the restriction to the surface of a solution of this equation is a solution of the original surface PDE.

If the resulting Cartesian differential operators and extension operators are discretized in space as in [MR09], we obtain an ordinary differential equation in the computational band. We therefore have a method of lines formulation of the closest point method, and thus can use higher-order time discretizations, such as Runge–Kutta methods.

We demonstrate the effectiveness of the method with examples of evolutionary PDEs on smooth surfaces such as the unit sphere in $\mathbb{R}^3$, as well as triangulated surfaces without parameterizations. We begin by implementing surface diffusion equations

based on the Laplace–Beltrami operator, and then generalize the method to higher-order operators, such as the biharmonic operator. More general cases of surface diffusion, in particular nonlinear and curvature-dependent diffusion, are shown to be simple to handle with this formulation.

Finally, we turn to applications in PDE-based image processing. Common techniques for enhancing images, such as denoising, deblurring, and inpainting (for a review, see [CS05]) can be applied to images defined on manifolds [Kim97, SK05]. The algorithm uses standard PDE models in image processing as a starting point, with small modifications to apply them to surface images. Applications such as surface image denoising [BvGMM13] are demonstrated, and results are shown on various triangulated and parameterized surfaces.

1.1.3 Contributions of this thesis

The main original contribution of this thesis is the penalty formulation of Chapter 3, based on a proof that a solution of the penalized embedding equation, when restricted to the surface, results in a solution of the surface PDE. The derivation of the modified embedding equation relies on the introduction of new extension and restriction operators, along with certain properties of these operators (Section 2.3 and onwards). The analysis of the resulting discrete scheme is presented in Chapter 4, along with numerical convergence studies. Certain results and numerical studies also appear in [vGMM14]. The thesis then contributes new applications of the closest point method, in particular image inpainting and curvature-dependent reaction-diffusion equations on surfaces. Some examples of Chapter 6 appear in [BvGMM13].

1.2 Numerical approaches to surface PDEs

In this section, we review methods for computing numerical solutions of surface partial differential equations. These approaches can often be assigned to one of two types: those intrinsic to the surface, and those which embed the surface into the surrounding space.

1.2.1 Surface intrinsic methods

Intrinsic methods are those which use either a parameterization of the surface or create an approximation to it, and then solve the PDE directly on the parameterized

or approximated surface. Discrete versions of the differential operators are formulated using, for example, finite difference or finite element discretizations.

A tutorial and survey of surface parameterization is found in [FH05]. For a simple surface such as the sphere, a global parameterization can be constructed in terms of polar coordinates. However, many surfaces of interest do not admit such parameterizations, and indeed these polar coordinates result in singularities even on this smooth surface. Due to their importance in models of geophysical flows (e.g. [LR01]), partial differential equations on the sphere have been well-studied. Various alternative grids which are more regular have been constructed over the sphere, for example by using projections of inscribed polyhedra, or using mappings of Cartesian grids onto the surface [CHL08].

For non-parameterized surfaces, a discrete approximation can be constructed, such as a piecewise polygonal approximation of a surface embedded in $\mathbb{R}^3$. Partial differential equations on such surfaces are considered in [Dzi88], where a discrete version of the Laplace–Beltrami operator for triangulated surfaces is introduced, and surface finite element spaces constructed on the triangulation. The method is extended to parabolic equations on stationary surfaces in [DE07b], and further to the evolving surface finite element method for time-dependent surfaces [DE07a]. Applications in biological models such as cell motility [ESV12] have been studied using this formulation. A review of these and other methods is found in [DE13].

An alternative finite element method, using finite element spaces obtained by the projection of an outer triangulation, is described in [ORG09, OR10]. A framework for intrinsic image processing on surfaces is presented in [LC11], in which discrete differential geometry operators are formulated directly on triangulated surfaces.

1.2.2 Embedding methods

In embedding methods, the surface is considered as a subspace of an ambient space, and computations are carried out in this surrounding space. After an extended equation has been solved, the solution is restricted back to the surface. Note that the embedding formulation for moving surfaces is also referred to as an *Eulerian* formulation (e.g. [Gre06, DE08, XZ03]).

Many such methods use a level set representation of the surface [OS88, OF03], in which the surface is defined implicitly as the zero level set of a function in the ambient space. Many surface quantities, such as surface normals, can be expressed simply in terms of the level set function. These methods apply to hypersurfaces only, i.e., surfaces with codimension one.

An implicit framework for variational surface problems based on a level set formulation is introduced in [BCOS01]. Gradients in the embedding space are expressed in terms of projections onto the tangent spaces of the level sets, and finite difference schemes are then used to compute the solution of the embedding equation. A modified version appears in [Gre06], which takes into account effects due to curvature away from the surface. This is also adapted to fourth-order PDEs in [GBS06].

A finite element approximation for elliptic problems is introduced in [Bur09], in which the level set formulation above is used to define and analyse extended finite element spaces. An Eulerian finite element method is also used in [DE08] to treat parabolic problems, and extended to implement transport and diffusion on evolving surfaces [DE10]. A related narrow band method [DDEH10] uses unfitted finite element spaces in a narrow band surrounding the surface.

An alternative approach to surface problems, involving traits of both intrinsic and embedding methods, uses radial basis functions (RBFs) [FW13, SWKF14]. The surface is represented at scattered nodes, and surface derivatives are approximated by differentiating the RBF interpolant. Another RBF-based approach is the orthogonal gradients (OGr) method [Pir12], in which RBFs are used to reconstruct an implicit level surface in the embedding space. The gradients in the normal direction to the surface are forced to vanish, allowing a closest point approximation of the differential operators in the embedding space.

Surfaces are increasingly being represented in the graphics community as sets of sampled points, or point clouds, rather than triangulated meshes. The point clouds are defined explicitly in a surrounding space, making an embedding method a natural choice for solving equations on the surface (e.g. [LZ13, MMR13]). Differential operators can be represented in the surrounding space using convolution kernels [MMR13].

The surface could instead be stored implicity as a set of points which are the 'footprints' (closest points) of the grid points of an underlying Cartesian grid, as in the grid based particle method [LZ09, LLZ11] for moving surfaces. The surface is represented using meshless particles, which are moved at each time step according to the interface motion. The surface is locally reconstructed, and the closest point representation is resampled at each time step.

Embedding methods have also been used for solving eigenvalue problems using both level set [Bra08] and closest point [MBR11] formulations.

1.2.3 Discussion

The choice of an intrinsic or embedding method may depend on the problem considered. Intrinsic methods have an advantage in computational efficiency, since they do not increase the dimension of the problem. However, embedding methods can often be constructed in a narrow band surrounding the surface (e.g. [DDEH10, RM08]), where the width of the band scales with the mesh size of the computational grid, so that the number of degrees of freedom of the problem is only a small constant multiple of the number of degrees of freedom on the surface. The closest point method, in particular, does not require that the boundary of the computational band is a level set, and artificial boundary conditions are not specified at the boundaries of the band.

When considering moving surfaces, embedding methods can handle topological changes of the surface, such as droplets coalescing or splitting, which are difficult to formulate using the parameterizations or triangulations of intrinsic methods. In addition, level set and closest point methods do not require remeshing at each timestep, although local interpolation may be used to reconstruct the surface.

Level set-based embedding methods require that the surface is a closed hypersurface, while methods based on closest point [RM08, MR08] or grid based particle representations [LZ09, LLZ11] may be more flexible with respect to the topology and codimension of the surface. Problems involving surface to bulk coupling can be handled simply with the closest point method [MMR13].

In practice, the choice of method may depend on the given representation of the surface, for example an initial triangulation, point cloud, or implicit surface, although fast methods exist to convert between explicit and implicit representations (e.g. [Set96]).

1.3 Representations of surfaces

In the remainder of this chapter, we define notation used throughout the thesis, and specify properties of the sets we refer to as surfaces (in this case, embedded submanifolds of $\mathbb{R}^n$). In later chapters, the introduction of a closest point representation will encode the geometry of the surface into a closest point map. This will allow partial differential equations on surfaces to be defined without charts and atlases.

We first review definitions of surfaces, implicit surfaces, and the associated tangent and normal spaces. The embedding band, in which numerical computations will be performed, is then introduced. This is followed by definitions required to perform calculus on surfaces, such as functions on surfaces, as well as surface differential

operators. Finally, some geometric properties of surfaces relating to curvature are introduced. Most notation follows that of [MM12] or [DE13]. Further references for calculus on smooth manifolds are [Lee09] and [Sim84].

1.3.1 Submanifolds of $\mathbb{R}^n$

Define an *embedding space* to be $\mathbb{R}^n$, with $n \in \mathbb{N}$. We call a subset $\mathcal{M} \subset \mathbb{R}^n$ a k-dimensional *submanifold* of $\mathbb{R}^n$ if, for every point $y \in \mathcal{M}$, there is an open neighbourhood $U \subset \mathbb{R}^n$ of y and an open set $W \subset \mathbb{R}^n$, such that there exists a homeomorphism $\varphi : U \to W$ with

$$\varphi(U \cap \mathcal{M}) = W \cap \mathbb{R}^k_0, \tag{1.9}$$

where $\mathbb{R}^k_0 = \{x \in \mathbb{R}^n : x = (x_1, ..., x_k, 0, ..., 0)\}$ [BG88]. Informally, we say the submanifold looks locally like $\mathbb{R}^k$.

In this work, we will call such a submanifold a *surface*, denoted by $\mathcal{S}$. Note that we use the term surface to mean a submanifold of any dimension $k \leq n$ in an embedding space $\mathbb{R}^n$, rather than using a common definition which is restricted to the case $n = 3$, $k = 2$. Note also that we allow the definition of a surface to include the codimension-0 case, where $k = n$ and $\mathcal{S}$ is an open set in $\mathbb{R}^n$.

The *smoothness* of the surface is determined by the smoothness of the maps φ. We say that $\mathcal{S}$ is a C^r surface if, for all points $y \in \mathcal{S}$, there exists a function φ as above which is a C^r diffeomorphism in U. If the surface is C^r for all $r \geq 0$, then we say $\mathcal{S}$ is a *smooth* or C^∞ surface.

The surface $\mathcal{S}$ is said to be a *closed* surface if it is closed as a set in $\mathbb{R}^n$, and it is *compact*, if it is closed and bounded. Under the definition of the submanifold above, a closed surface has no boundary.

1.3.2 Implicit surfaces and level set functions

Define the *codimension* of the surface to be $\operatorname{codim} \mathcal{S} = n - k$, the difference in dimensions of the embedding space and the surface. We call a surface with codimension one a *hypersurface*. A C^1 hypersurface $\mathcal{S}$ can be represented locally as the zero level set of a C^1 function $\phi : U \to \mathbb{R}$, satisfying $\nabla\phi(x) \neq 0$, $\forall x \in \mathcal{S} \cap U$. For example, we can choose ϕ to be the nth component of the map $\varphi : U \to W$. Locally, the surface is given implicitly by the zero set

$$\mathcal{S} \cap U = \{x \in U | \phi(x) = 0\}. \tag{1.10}$$

Note that the choice of ϕ is not unique.

A common choice is the *signed distance* (also called oriented distance) function. If the surface is the boundary of a bounded open set $\Omega \subset \mathbb{R}^n$, so that $\mathcal{S} = \partial\Omega \subset \mathbb{R}^n$, then the signed distance to the boundary is defined as

$$\mathrm{sd}(x) := \begin{cases} \inf_{y\in\mathcal{S}} |x-y| & x \in \bar{\Omega}, \\ -\inf_{y\in\mathcal{S}} |x-y| & x \in \mathbb{R}^n \backslash \bar{\Omega}. \end{cases} \tag{1.11}$$

The function sd is Lipschitz continuous on $\mathbb{R}^n$. If $\mathcal{S}$ is a C^r hypersurface with $r \geq 2$, then sd is C^r in a neighbourhood of the surface [GT01], and satisfies the Eikonal equation $|\nabla \mathrm{sd}| = 1$. Note that the convention here is chosen with $\mathrm{sd} > 0$ in the interior of Ω and $\mathrm{sd} < 0$ in the exterior of Ω.

We will use $\mathrm{sd}(x)$ to denote the signed distance function, and $\mathrm{d}(x)$ to denote the Euclidean (unsigned) distance function

$$\mathrm{d}(x) := \mathrm{dist}(x, \mathcal{S}), \quad x \in B(\mathcal{S}). \tag{1.12}$$

1.3.3 Tangent and normal spaces

In the following section we restrict our attention to C^1 surfaces. For a point $y \in \mathcal{S}$, we define a tangent vector at y to be a vector $\tau \in \mathbb{R}^n$, which is tangent to some differentiable curve $\gamma : (-\epsilon, \epsilon) \to \mathbb{R}^n$ passing through y, with $\gamma(0) = y$, $\gamma'(0) = \tau$, and $\gamma((-\epsilon, \epsilon)) \subset \mathcal{S}$. The tangent space $\mathcal{T}_y\mathcal{S}$ at y is a vector space of dimension k spanned by all tangent vectors at y.

The normal space $\mathcal{N}_y\mathcal{S}$ at y consists of all vectors $\nu \in \mathbb{R}^n$ perpendicular to the tangent space $\mathcal{T}_y\mathcal{S}$. We can choose a basis of orthonormal vectors spanning the normal space at each point.

In the case of a hypersurface, a unit normal vector at $y \in \mathcal{S}$ is given by

$$\nu(y) = \frac{\nabla\phi(y)}{|\nabla\phi(y)|}, \qquad y \in \mathcal{S}, \tag{1.13}$$

where $\phi(x)$ is a level set function representing $\mathcal{S}$ (1.10). This has a natural extension $\nu(x) = \frac{\nabla\phi(x)}{|\nabla\phi(x)|}$, defined in a neighbourhood U of the surface, where $\nu(x)$ is now the normal to the particular level set which goes through the point $x \in U$. We require that $\nabla\phi(x) \neq 0$ in U. If the level set function is chosen to be the signed distance function $\phi(x) = \mathrm{sd}(x)$, then the unit normal vector simplifies further to $\nu(x) = \nabla\mathrm{sd}(x)$.

A hypersurface is called *orientable* if there exists a continuous vector field $\nu : \mathcal{S} \to \mathbb{R}^n$ such that $\nu(y)$ is a unit normal vector to $\mathcal{S}$ for all $y \in \mathcal{S}$. For surfaces of higher codimension, in which the normal space has dimension greater than 1, we

define orientability as the existence of $n-k$ continuous vector fields ν_i which span the normal space.

We now introduce the *orthogonal projection* operator $P_y : \mathbb{R}^n \to \mathcal{T}_y\mathcal{S}$. For hypersurfaces, we can represent P_y as

$$P_y = I - \nu(y)\nu(y)^T = I - \frac{(\nabla\phi(y))(\nabla\phi(y))^T}{|\nabla\phi(y)|^2}, \qquad y \in \mathcal{S}. \tag{1.14}$$

In general, for surfaces of higher codimension, we have

$$P_y = I - \sum_{i=1}^{n-k} \nu_i(y)\nu_i(y)^T, \qquad y \in \mathcal{S}, \tag{1.15}$$

where the $n-k$ orthonormal vectors $\nu_i(y)$ span the normal space $\mathcal{N}_y\mathcal{S}$.

The operator P_y above is only defined for surface points $y \in \mathcal{S}$. For hypersurfaces, we define a natural generalization of the projection operator in a neighbourhood U of the surface. Let $P_x : \mathbb{R}^n \to \mathcal{T}_x\mathcal{S}_\phi$, where $\mathcal{T}_x\mathcal{S}_\phi$ is the span of all vectors which are tangent at x to the level set of ϕ passing through $x \in U$, and

$$P_x := I - \nu(x)\nu(x)^T = I - \frac{(\nabla\phi(x))(\nabla\phi(x))^T}{|\nabla\phi(x)|^2}, \qquad x \in U. \tag{1.16}$$

Since $\nu(x)$ is normal to the level set of ϕ intersecting x, the operator P_x projects vectors $v \in \mathbb{R}^n$ onto the tangent space of the level set at the point $x \in U$.

1.3.4 Embedding band

We will perform computations in a neighbourhood $B(\mathcal{S}) \subset \mathbb{R}^n$ of the surface $\mathcal{S}$. We refer to an open set $B(\mathcal{S})$ with $\mathcal{S} \subset B(\mathcal{S}) \subset \mathbb{R}^n$ as a *band* or *embedding band* around $\mathcal{S}$. The existence of this neighbourhood is sufficient for the existence of retractions to the surface (see Lemma 2.1.2), which will later be used in the definition of closest point functions.

The boundary $\partial B(\mathcal{S})$ of the embedding band is not required in general to be a level set, however in certain cases we will use a band of fixed width. We write $B_\alpha(\mathcal{S})$ for a band of width α about $\mathcal{S}$, i.e.,

$$B_\alpha(\mathcal{S}) := \{x \in \mathbb{R}^n : \operatorname{dist}(x, \mathcal{S}) < \alpha\}. \tag{1.17}$$

1.4 Calculus on surfaces

Surface partial differential equations are formulated in terms of differential operators that are intrinsic to the surface. We introduce some of these operators below.

1.4.1 Functions on surfaces

With the notation of Section 1.3.1, we will say that a function $u : \mathcal{S} \to \mathbb{R}$ is in $C^l(\mathcal{S})$ if $u \circ \varphi^{-1} : W \cap \mathbb{R}^k_0 \to \mathbb{R}$ is l-times continuously differentiable for any pair (U, φ) satisfying (1.9). This requires that the surface is smooth enough, i.e., that $\mathcal{S}$ is at least C^l. If $u \in C^l(\mathcal{S})$ for all $l \geq 0$, then we say u is *smooth*, or $C^\infty(\mathcal{S})$.

We say that a vector-valued surface function $f : \mathcal{S} \to \mathbb{R}^m$ is in $C^l(\mathcal{S}; \mathbb{R}^m)$ if each component of f is $C^l(\mathcal{S})$. A surface vector field is a function $g : \mathcal{S} \to \mathbb{R}^n$, with n the dimension of the embedding space. If $g(y) \in \mathcal{T}_y\mathcal{S}$ for all points $y \in \mathcal{S}$, then g is called a tangential vector field.

We can introduce norms on functions over the surface. For a bounded continuous function u, let the maximum and L^2 norms be defined by

$$||u||_{\infty,\mathcal{S}} := \sup_{y \in \mathcal{S}} |u(y)|, \tag{1.18}$$

$$||u||_{2,\mathcal{S}} := \left(\int_{\mathcal{S}} |u(y)|^2 ds(y) \right)^{\frac{1}{2}}, \tag{1.19}$$

where $ds(y)$ is the surface Lebesgue measure (the k-dimensional Hausdorff measure).

The function u above is defined only on the surface. In order to work in the embedding space, we consider a related function in a neighbourhood of the surface. We call a function $\bar{u} : B(\mathcal{S}) \to \mathbb{R}$ a *regular extension* of u into $B(\mathcal{S})$, if $\bar{u}$ is as smooth as u, and agrees with u on the surface, so that $\bar{u}(y) = u(y)$ for $y \in \mathcal{S}$. In Chapter 2, we show that such regular extensions exist for sufficiently smooth surfaces.

1.4.2 Surface differential operators

Differential operators on surfaces can often be defined through projections. Let u be in $C^1(\mathcal{S})$, and $\bar{u}$ be a regular extension of u into $B(\mathcal{S})$. The surface gradient of u can be expressed as

$$\nabla_{\mathcal{S}} u(y) = P_y \nabla \bar{u}(y), \tag{1.20}$$

where P_y is the projection operator onto the tangent space $\mathcal{T}_y\mathcal{S}$. The gradient is independent of the choice of extension $\bar{u}$ of the surface function u (e.g. [DE13, Lemma 2.4]).

For $f \in C^1(\mathcal{S};\mathbb{R}^m)$, and $g \in C^1(\mathcal{S};\mathbb{R}^n)$, we express the surface Jacobian and divergence operators as

$$D_{\mathcal{S}} f(y) = D\bar{f}(y)P_y, \tag{1.21}$$

$$\operatorname{div}_{\mathcal{S}} g(y) = \operatorname{tr}(D\bar{g}(y)P_y). \tag{1.22}$$

Many higher-order operators can be constructed as compositions of the above differential operators. For example, the Laplace–Beltrami operator (surface Laplacian) is defined by

$$\Delta_{\mathcal{S}} u := \operatorname{div}_{\mathcal{S}} \nabla_{\mathcal{S}} u, \quad u \in C^2(\mathcal{S}). \tag{1.23}$$

1.4.3 Mean curvature

In this section, we use the signed distance function to define the curvature of the surface. For a C^2 hypersurface $\mathcal{S}$, the *Weingarten map* or shape operator is given by the Hessian of the signed distance map

$$\mathcal{H}(x) = D^2 \mathrm{sd}(x). \tag{1.24}$$

The eigenvalues of this matrix, denoted κ_i, are called the principal curvatures of the surface. The trace of the Weingarten map is called the *mean curvature*, denoted κ,

$$\kappa(x) := \operatorname{tr}\left(\mathcal{H}(x)\right) = \sum_{i=1}^{k} \kappa_i(x). \tag{1.25}$$

The function $\kappa(x)$ gives the mean curvature of the level set at the point x. If κ is evaluated at a point $y \in \mathcal{S}$, this is equal to the mean curvature of the surface $\mathcal{S}$ at y. Note that there are different definitions of mean curvature; here, as is common in the level set community, we have not divided by the dimension k.

Since the normal vector can be expressed in terms of the signed distance function as $\nu(x) = \nabla \mathrm{sd}(x)$, we also have that

$$\kappa(x) = \operatorname{tr}\left(D^2 \mathrm{sd}(x)\right) = \operatorname{div} \nu(x). \tag{1.26}$$

1.5 Outline of thesis

The thesis proceeds as follows. Chapter 2 begins with a review of calculus on surfaces with closest point functions [MM12]. We then introduce new extension operators based on these closest point functions, and describe their properties. These operators are used to derive the gradient and divergence principles of [RM08, MM12]. In

Chapter 3, we formulate a new embedding equation, which is consistent with the original surface PDE, and includes a constraint enforced through a penalty term. The discretization of this equation, using a method of lines, is then described in Chapter 4, and the consistency and stability properties of the numerical scheme are investigated. Chapter 5 presents applications of the method to diffusion equations, including reaction-diffusion and curvature-dependent diffusion. Finally, applications in image processing on surfaces, such as image denoising and inpainting, are demonstrated in Chapter 6.

Chapter 2

Closest Point Representations and Extensions

The closest point method is formulated in terms of an implicit representation of the surface, using a *closest point map*. For each point x in the surrounding space $B(\mathcal{S})$, a point on the surface $\text{cp}(x)$ is defined, and the surface is then represented implicitly as the set of fixed points $\mathcal{S} := \{x \in B(\mathcal{S}) \mid x = \text{cp}(x)\}$. This representation allows us to compute gradients of functions on the surface, avoiding the manipulation of charts and atlases. In addition, it is defined for surfaces of general codimension and is not restrictive with respect to the topology of the surface.

In the closest point method introduced in [RM08], the point $\text{cp}(x)$ was defined to be the point on the surface closest in Euclidean distance to x. In this work, we use a more general definition from [MM12] for a closest point function, as a retraction from $B(\mathcal{S})$ to $\mathcal{S}$, with the additional property that the manifolds of constant cp intersect the surface orthogonally. When composed with the map cp, a function $u \circ \text{cp}$ will therefore be locally constant in the direction orthogonal to the surface.

This chapter begins with a discussion of regular extensions and retractions. We show that, given a continuous function u on a sufficiently smooth surface $\mathcal{S}$, we can construct a regular extension $\bar{u}$ in a band around the surface, which has the same smoothness as u, and agrees with u when restricted to $\mathcal{S}$. We then state the definition of a closest point function as a retraction [MM12], and give an alternative proof that such functions exist and are well-defined in a certain band.

The operators underlying the contributions of this thesis are then introduced in Section 2.3. The closest point function cp is used to construct a new linear extension operator E_S, which maps a function u on the surface to a function v in the embedding space, with $v = E_S u := u \circ \text{cp}$. We analyse properties of the continuous operator E_S

and the corresponding restriction operator, and give examples for various surfaces. These operators are then used to rederive principles from [RM08, MM12], relating the gradient and divergence operators of the function u on the surface to the corresponding Cartesian differential operators of the function v. Finally, we show that the explicit and implicit closest point methods of [RM08, MR09] can be expressed in a simple way using this new formulation.

2.1 Surfaces and embedding bands

In the following, let $\mathcal{S}$ be a surface of dimension k, embedded in $\mathbb{R}^n, n \geq k$, with a neighbourhood or *embedding band* $B(\mathcal{S}) \subset \mathbb{R}^n$ containing the surface $\mathcal{S}$.

2.1.1 Retractions and extensions

In order to solve a differential equation given on the surface, we formulate and solve a related equation in the surrounding embedding band. We aim to construct an equation which is consistent with the original problem on the surface, i.e., such that the solution of the new embedding equation, when restricted to the surface, is a solution of the original equation. We then wish to show that the solution of the new equation will be more amenable to discretization with standard numerical schemes, compared to previous formulations of the closest point method.

With a view to relating solutions of these equations, we review here some simple concepts from differential geometry (e.g. [Hir76, Lee09]). In particular, we discuss extensions of functions on the surface into the surrounding band, and retractions from the embedding space onto the surface. Starting with a $C^l(\mathcal{S})$ function u, we define an extended function v in $B(\mathcal{S})$, where u and v agree on $\mathcal{S}$. If we can choose the function v to be in $C^l(B(\mathcal{S}))$, then we say that v is a regular extension of u. We will show in later sections that this property is satisfied for smooth surfaces if we choose an extension based on a closest point map. In this section, we first show that regular extensions exist, constructed from general retractions onto the surface.

Definition 2.1.1. *A* retraction *from a set U onto a subset $V \subset U$ is a continuous map $\rho : U \to V$ such that $\rho|_V$ is the identity map on V.*

Lemma 2.1.2 ([MM12])**.** *Let $\mathcal{S}$ be a C^r surface, with $r \geq 0$. Then for each $y \in \mathcal{S}$, there exists a neighbourhood U of y and a C^r retraction $\rho : U \to \mathcal{S} \cap U$.*

Proof. Recall that since $\mathcal{S}$ is an embedded submanifold of $\mathbb{R}^n$, for each point $y \in \mathcal{S}$ there exists a neighbourhood $U \subset \mathbb{R}^n$ of y and a C^r diffeomorphism $\varphi : U \to W \subset \mathbb{R}^n$, such that $\varphi(\mathcal{S} \cap U) = W \cap \mathbb{R}^k_0$, where $\mathbb{R}^k_0 = \{x \in \mathbb{R}^n : x = (x_1, ..., x_k, 0, ..., 0)\}$. Let $\pi : \mathbb{R}^n \to \mathbb{R}^k_0$ be the orthogonal projector onto $\mathbb{R}^k_0$. Then define $\rho : U \to \mathcal{S} \cap U$ by $\rho = \varphi^{-1} \circ \pi \circ \varphi$, which is also a C^r map. Since $\rho(y) = \varphi^{-1} \circ \pi \circ \varphi(y) = \varphi^{-1} \circ \varphi(y) = y$ for $y \in \mathcal{S} \cap U$, it follows that ρ is a retraction. □

Note that Lemma 2.1.2 gives the existence of local retractions, used in [MM12]. In this work we use a partition of unity to extend this proof to the existence of a global regular extension.

Lemma 2.1.3. *Let $\mathcal{S}$ be a compact C^r surface, with $r \geq 0$. Then there exists a neighbourhood $U \subset \mathbb{R}^n$ of $\mathcal{S}$, such that for any function $u \in C^l(\mathcal{S})$ with $l \leq r$, there exists a regular extension $\bar{u} \in C^l(U)$ with $\bar{u}|_{\mathcal{S}} = u$.*

Proof. For each $y \in \mathcal{S}$, let (U_y, ρ_y) be as in Lemma 2.1.2. Then $\{U_y\}_{y \in \mathcal{S}}$ is an open cover of $\mathcal{S}$. Since $\mathcal{S}$ is compact, there exists a finite subcover $\{U_k\}_{k=1}^N$ of $\mathcal{S}$ with corresponding retractions ρ_k. By [AF03, Theorem 3.15], there exists a partition of unity for $\mathcal{S}$ subordinate to $\{U_k\}_{k=1}^N$ given by $\{\psi_k\}_{k=1}^N$, with $\psi_k \in C_0^\infty(U_k)$, and $\sum_{k=1}^N \psi_k(y) = 1$ for all $y \in \mathcal{S}$.

Define the local extension $\bar{u}_k : U_k \to \mathbb{R}$ by $\bar{u}_k(x) = u(\rho_k(x))$, with $\rho_k(x)$ a local retraction on U_k as in Lemma 2.1.2. Each $\bar{u}_k$ is in $C^l(U_k)$, since ρ_k is a C^r retraction, u is a $C^l(\mathcal{S})$ function, and $l \leq r$. Then for the neighbourhood $U := \bigcup_{k=1}^N U_k$ of $\mathcal{S}$, the global extension $\bar{u} : U \to \mathbb{R}$ can be defined as $\bar{u}(x) := \sum_{k=1}^N \psi_k(x)\bar{u}_k(x)$. For $y \in \mathcal{S}$, we have

$$\psi_k(y)\bar{u}_k(y) = \begin{cases} \psi_k(y)u(y) & \text{if } y \in U_k, \\ 0 & \text{if } y \notin U_k, \end{cases} \tag{2.1}$$

so that

$$\bar{u}(y) = \sum_{k=1}^N \psi_k(y)\bar{u}_k(y) = \left(\sum_{k=1}^N \psi_k(y)\right) u(y) = u(y), \tag{2.2}$$

as required. □

2.2 Closest point functions

In the previous section, a regular extension $\bar{u}$ of u was defined though the use of local retractions. We now make use of a particular global retraction that has additional useful properties. The following general class of retractions $\mathrm{cp} : B(\mathcal{S}) \to \mathcal{S}$ that map points in a neighbourhood to the surface was introduced in [MM12].

Definition 2.2.1 (Closest Point Function [MM12]). *Let $\mathcal{S}$ be a surface, and $B(\mathcal{S})$ a neighbourhood of $\mathcal{S}$. The map* $\mathrm{cp} : B(\mathcal{S}) \to \mathcal{S}$ *is called a* closest point function, *if* cp *is a C^1-retraction and $D\mathrm{cp}(y) = P_y$ for all $y \in \mathcal{S}$.*

Recall that P_y is the projection operator that maps vectors in $\mathbb{R}^n$ onto the tangent space $\mathcal{T}_y\mathcal{S}$ at a point y on the surface. A continuously differentiable retraction cp is therefore a closest point function if the Jacobian matrix $D\mathrm{cp}$ is a projection at the surface. This key property is required for the proof of identities relating surface differential operators to differential operators in the embedding space, and is equivalent to a local characterization of the pre-image of the function cp at the boundary; we state the following result from [MM12, Theorem 3.7] without proof.

Lemma 2.2.2 (Geometric characterization of closest point functions [MM12]). *Let* cp *be a closest point function, and $y \in \mathcal{S}$. The pre-image* $\mathrm{cp}^{-1}(y)$ *is locally a C^1 manifold, and intersects the surface orthogonally at y.*

Since the manifold of constant cp is locally orthogonal to the surface, the composition $u \circ \mathrm{cp} : B(\mathcal{S}) \to \mathbb{R}$ is constant in any direction in the normal space $\mathcal{N}_y S$. We say that the map cp propagates data perpendicularly off the surface.

The name 'closest point' originally referred to the function introduced in [RM08], which returns the point on the surface that is closest to x in Euclidean distance. However, Definition 2.2.1 is more general, and includes for example functions constructed from steepest descent trajectories of non-Euclidean distance functions [MM12].

In the following section, we show the existence of cp functions using the example of the original Euclidean closest point function, and relate the smoothness of these functions to that of the surface. We will concentrate on the case of hypersurfaces, as in e.g. [DE13, Gre06, ORG09].

2.2.1 Euclidean closest point function

A special case of a closest point function $\mathrm{ecp} : B(\mathcal{S}) \to \mathcal{S}$ was first introduced in [RM08], based on Euclidean distance to the surface.

Definition 2.2.3 (Euclidean Closest Point Function). *Let $\mathcal{S}$ be a compact surface, and $B(\mathcal{S})$ a neighbourhood of $\mathcal{S}$. Under certain assumptions on $\mathcal{S}$, the Euclidean closest point function* $\mathrm{ecp} : B(\mathcal{S}) \to \mathcal{S}$ *is defined by*

$$\mathrm{ecp}(x) := \mathrm{argmin}_{y \in \mathcal{S}} |x - y|, \quad x \in B(\mathcal{S}). \tag{2.3}$$

It is clear that $\mathrm{ecp}(y) = y$ for points $y \in \mathcal{S}$, so this function is a retraction onto $\mathcal{S}$, provided that it is continuous. Additional assumptions on the smoothness of $\mathcal{S}$ are required to ensure that this definition is well-defined. We now show that, at least for a particular class of surfaces, the value of $\mathrm{ecp}(x)$ is unique inside a band surrounding the surface, and the function ecp is one order less smooth than $\mathcal{S}$.

Following [GT01], we define a local coordinate system in a neighbourhood of the surface, which involves a closest point type representation (see Figure 2.1).

Lemma 2.2.4. *Let $\mathcal{S}$ be a compact C^2 hypersurface, such that $\mathcal{S}$ is the boundary of a connected open set Ω. Then there exists $\alpha > 0$ such that each point x in a band $B_\alpha(\mathcal{S})$ of width α about $\mathcal{S}$ can be written uniquely as*

$$x = a(x) + \mathrm{sd}(x)\nu(a(x)), \tag{2.4}$$

for some $a(x) \in \mathcal{S}$, where $\nu(a(x))$ is the normal to the surface at this point, and $\mathrm{sd}(x)$ is the signed distance function to the surface.

Proof. By hypothesis, the compact hypersurface $\mathcal{S}$ is the boundary of a region $\Omega \subset \mathbb{R}^n$, where $\mathcal{S} = \partial\Omega$ is C^2. Therefore, the boundary $\partial\Omega$ satifies a uniform interior sphere condition [GT01], i.e., there exists $\delta_1 > 0$, such that for each point $y \in \mathcal{S}$, there exists $z \in \Omega$ such that the ball $B = B(z, \delta_1) \subset \Omega$ with $|y - z| = \delta_1$. The surface also satisfies a uniform exterior sphere condition, with minimum radius δ_2.

Now for any $\epsilon < \delta_1$, if a point x satisfies $\mathrm{dist}(x, \mathcal{S}) = \epsilon$, then the ball $B(x, \epsilon)$ lies inside a ball $B(z, \delta_1) \subset \Omega$. There exists therefore a unique $y \in \mathcal{S}$, with $\bar{B}(x, \epsilon) \cap \mathcal{S} = \{y\}$. The point y is the unique closest point on the surface to x, and $x - y$ is perpendicular to the tangent space at y, so that $x - y = |x - y|\nu(y)$, with a suitable choice of the normal orientation.

Then we can write $a(x) = y$, so that $x = a(x) + \mathrm{sd}(x)\nu(a(x))$, where $\mathrm{sd}(x) = \inf_{y\in\mathcal{S}}|x-y|$ inside Ω. Using the exterior sphere condition, we find that a similar result holds outside of Ω, with an opposite choice of sign for the normal. This representation is therefore unique in a band of width $\alpha := \min(\delta_1, \delta_2)$ about the surface. □

We now define $\mathrm{ecp}(x) := a(x) = \mathrm{argmin}_{y\in\mathcal{S}}|x - y|$, as in Definition 2.2.3. From Lemma 2.2.4, this closest point is unique inside $B_\alpha(\mathcal{S})$.

For a C^r hypersurface $\mathcal{S}$, with $r \geq 2$, the signed distance map sd is C^r in some neighbourhood $B(\mathcal{S})$ [GT01, Lemma 14.16]. From (2.4), $\nabla\mathrm{sd}(x) = \nu(a(x))$, and $|\nabla\mathrm{sd}(x)| = 1$. It follows that the Euclidean closest point map can be expressed in terms of the signed distance function as

$$\mathrm{ecp}(x) = x - \mathrm{sd}(x)\nabla\mathrm{sd}(x). \tag{2.5}$$

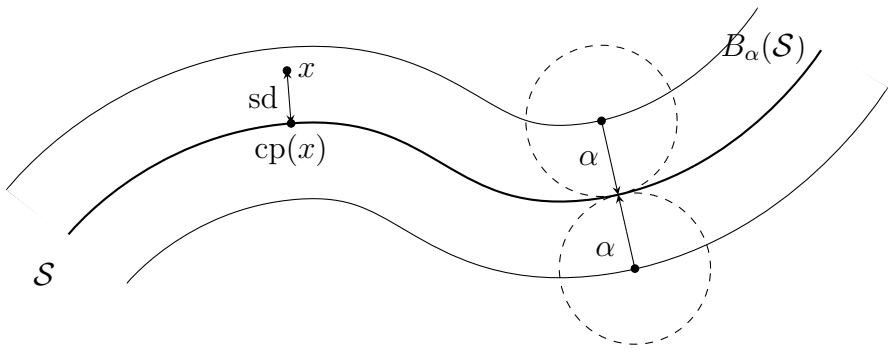


Figure 2.1: Each point x in the band $B_\alpha(\mathcal{S})$ of width α about $\mathcal{S}$ can be written uniquely as $x = \mathrm{cp}(x) + \mathrm{sd}(x)\nu(\mathrm{cp}(x))$.

Since sd is in $C^r(B(\mathcal{S}))$, we have that ecp is in $C^{r-1}(B(\mathcal{S}))$.

The construction above can be extended to surfaces of higher codimension, using the fact that the normal space can be spanned by a smooth vector field $(\nu_1, ..., \nu_{n-k})$, and writing the normal coordinate decomposition as [AS96]

$$x = a(x) + \sum_{i=1}^{n-k} \mathrm{d}_i(x)\nu_i(a(x)). \tag{2.6}$$

Note that the properties of the Euclidean distance closest point function can also be derived from the characteristics of the Eikonal equation.

Theorem 2.2.5. *For a C^r hypersurface $\mathcal{S}$ with $r \geq 2$, the Euclidean closest point map* ecp *satisfies the definition of a closest point function (Definition 2.2.1).*

Proof. [MM12] As above, the Euclidean closest point is given by

$$\mathrm{ecp}(x) = x - \mathrm{sd}(x)\nabla\mathrm{sd}(x), \tag{2.7}$$

where $\mathrm{sd}(x)$ is the signed distance function to the surface $\mathcal{S}$, so that $\mathrm{sd}(y) = 0$ at the surface, and $|\nabla\mathrm{sd}| = 1$. Since the map $\mathrm{ecp}(x)$ is C^{r-1} in a neighbourhood $B(\mathcal{S})$, and $r \geq 2$, we may take the derivative of this function to give

$$D\mathrm{ecp}(x) \;=\; I - \nabla\mathrm{sd}(x)\nabla\mathrm{sd}(x)^T - \mathrm{sd}(x)D^2\mathrm{sd}(x). \tag{2.8}$$

The normal to the surface at y is $\nu(y) = \frac{\nabla\mathrm{sd}(y)}{|\nabla\mathrm{sd}(y)|} = \nabla\mathrm{sd}(y)$, so that the projection is given by

$$P_y = I - \nu(y)\nu(y)^T = I - \nabla\mathrm{sd}(y)\nabla\mathrm{sd}(y)^T. \tag{2.9}$$

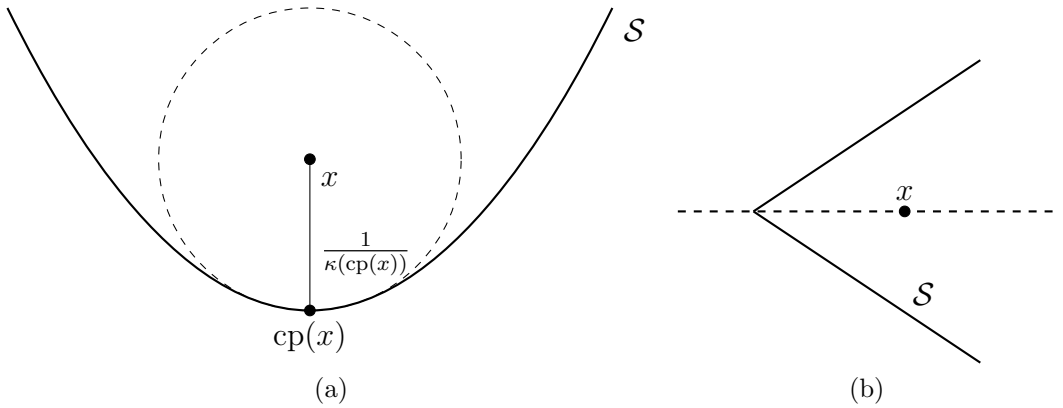


Figure 2.2: For a smooth curve $\mathcal{S}$, the Euclidean closest point is unique inside a band of width $\frac{1}{\kappa}$ (a). For C^0 curves with corners, the closest point is non-unique along the medial axis at the corner (b).

At the surface, since $\mathrm{sd}(y) = 0$,

$$D\mathrm{ecp}(y) = I - \nabla \mathrm{sd}(y) \nabla \mathrm{sd}(y)^T = P_y, \tag{2.10}$$

so the definition of a closest point function is satisfied. □

In [MM12], this is extended to general codimension for smooth surfaces, using the Euclidean distance map $\mathrm{d}(x)$ rather than the signed distance function.

2.2.2 Embedding band

For a C^2 hypersurface $\mathcal{S}$, the width of the band $B_\alpha(\mathcal{S})$ in which ecp is defined is given by α, the lower bound for the radius in the uniform interior and exterior sphere conditions in Lemma 2.2.4. Recall that for each $y \in \mathcal{S}$, there exists a ball $B(z, \delta) \subset \Omega$, and we choose $0 < \alpha < \delta(y)$ for all $y \in \mathcal{S}$. We can bound α by

$$\alpha \leq \mu := \left(\max_{i=1,..,k} ||\kappa_i||_{\infty,\mathcal{S}} \right)^{-1}, \tag{2.11}$$

where κ_i are the principal curvatures of the surface (see Section 1.4.3). For example, in the case of smooth curves in $\mathbb{R}^2$, the Euclidean closest point function ecp is well-defined in an open band of width $\alpha < \frac{1}{\kappa(s)}$, where $\kappa(s)$ is the curvature of the surface at a point s, or the inverse of the radius of the osculating circle (see Figure 2.2a).

If the surface is less smooth, ecp may not be uniquely defined in a band of any size $\alpha > 0$. For example, if $\mathcal{S}$ is a C^0 curve in $\mathbb{R}^2$ with a corner (as in Figure 2.2b), then for points on the medial axis at the corner, the global coordinates are not unique.

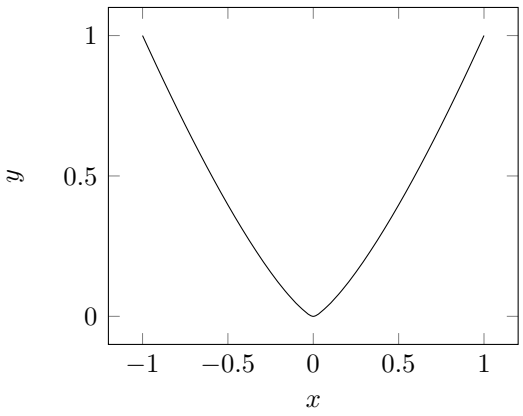


Figure 2.3: The curve $y = |x|^{4/3}$ is C^1 at $x = 0$, and does not have a unique Euclidean closest point in any neighbourhood of the surface.

Since the surface is only C^0, the projection P_y is discontinuous, so we cannot define any C^1 cp function satisfying the key property (Definition 2.2.1).

We now show an example of a C^1 curve for which the Euclidean closest point map is not unique in any neighbourhood. Consider the function $y = |x|^{4/3}$, near $x = 0$. Since $y''(x) = \frac{4}{9}|x|^{-2/3}$, the second derivative is infinite at $x = 0$, and the function is only C^1 here. Now

$$\lim_{x\to 0} \frac{y(x)}{x^2} = \lim_{x\to 0} |x|^{-2/3} = \infty, \tag{2.12}$$

so for any $c > 0$, there exists x_0 such that $y(x) \geq cx^2$, $\forall |x| < x_0$. Given a ball of radius c touching the curve at $x = 0$, we can choose an interval about $x = 0$ small enough so that the ball does not lie above the function in this interval. The surface therefore does not satisfy a uniform interior sphere condition. No point on the positive y-axis has a unique closest point on the surface, so we cannot choose a band surrounding the surface in which the Euclidean closest point is unique.

Note that the bound μ on the bandwidth (2.11) due to the curvature of the surface is a local condition. There may be an additional nonlocal bound on the width of the band, if $\mathcal{S}$ is the boundary of a non-convex region Ω. In this case, two surface points may be far in geodesic distance along $\mathcal{S}$, but be close in the embedding space. In this case, a closed ball $\bar{B}(z, \delta)$ intersecting the surface $\partial\Omega$ at y_1, with $\delta(y_1) < \mu$, may still intersect $\partial\Omega$ again at a geodesically distant point y_2. A surface which is the boundary of a convex region Ω will satisfy $\mu \leq \delta$, where $\delta = \min_{y\in\mathcal{S}} \delta(y)$, however in general we must use bandwidth $\alpha = \min(\mu, \delta)$.

2.2.3 Examples

We give some examples of Euclidean and non-Euclidean closest point functions for various simple surfaces. Note that $x \in B(\mathcal{S})$, $y \in \mathcal{S}$ refer to points in the band and on the surface respectively, with components (x_1, x_2) and (y_1, y_2) when the surface is embedded into $\mathbb{R}^2$.

1. Consider a circle of radius R in $\mathbb{R}^2$, centred at the origin, represented with the Euclidean closest point function cp = ecp. The surface is given by

$$\mathcal{S} = \{y \in \mathbb{R}^2 : |y| = R\}. \tag{2.13}$$

The closest point function is $\mathrm{cp}(x) = R\frac{x}{|x|}$, with magnitude $|\mathrm{cp}(x)| = R$. Note that this is defined everywhere except at $x = 0$, so the embedding band should be an annulus around the surface $B(\mathcal{S}) = \{x \in \mathbb{R}^2 : a_1 < |x| < a_2\}$ for some $a_1, a_2 \in \mathbb{R}$ with $0 < a_1 < 1 < a_2$. Unit normal vectors at the surface are $\nu(y) = \frac{y}{|y|} = \frac{y}{R}$, so the extension of the normal vector field ν into the band is

$$\nu(x) = \nu(\mathrm{cp}(x)) = \frac{\mathrm{cp}(x)}{|\mathrm{cp}(x)|} = \frac{\mathrm{cp}(x)}{R} = \frac{x}{|x|}. \tag{2.14}$$

The projection operator onto the surface at y in this case is

$$P_y = I - \nu(y)\nu(y)^T = \frac{1}{R^2}\begin{pmatrix} R^2 - y_1^2 & -y_1y_2 \\ -y_1y_2 & R^2 - y_2^2 \end{pmatrix} = \frac{1}{R^2}\begin{pmatrix} y_2^2 & -y_1y_2 \\ -y_1y_2 & y_1^2 \end{pmatrix}, \tag{2.15}$$

with $y = (y_1, y_2)$ and $y_1^2 + y_2^2 = R$. The Jacobian matrix of the closest point function is given by

$$D\mathrm{cp}(x) = \frac{R}{|x|^3}\begin{pmatrix} x_2^2 & -x_1x_2 \\ -x_1x_2 & x_1^2 \end{pmatrix}. \tag{2.16}$$

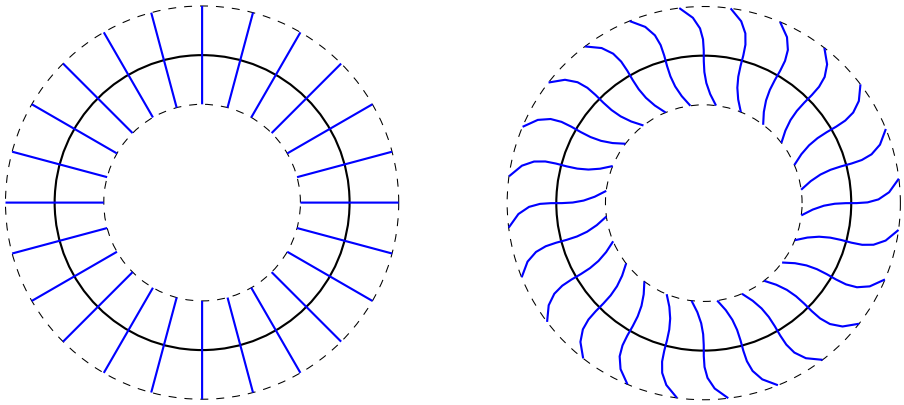
Then evaluating this at a point y on the surface, we have

$$D\mathrm{cp}(y) = \frac{1}{R^2}\begin{pmatrix} y_2^2 & -y_1y_2 \\ -y_1y_2 & y_1^2 \end{pmatrix} = P_y. \tag{2.17}$$

Thus the Jacobian $D\mathrm{cp}$ at points y on the surface is equal to a projection onto the surface: $D\mathrm{cp}(y) = P_y$, as expected by Definition 2.2.1.

2. We extend the first example to a $(n-1)$-sphere of radius R in $\mathbb{R}^n$, centred at the origin. The surface is $\mathcal{S} = \{y \in \mathbb{R}^n : |y| = R\}$, and the Euclidean closest point function is again $\mathrm{cp}(x) = R\frac{x}{|x|}$, in the region $B(\mathcal{S}) = \{x \in \mathbb{R}^n : a_1 < |x| < a_2\}$. The Jacobian can then be expressed as

$$[D\mathrm{cp}(x)]_{ij} = \frac{R}{|x|^3}\begin{cases} |x|^2 - x_i^2, & i = j \\ -x_ix_j, & i \neq j \end{cases} \tag{2.18}$$



(a) (b)

Figure 2.4: Lines of constant cp (shown in blue) for a unit circle, using (a) the Euclidean closest point function and (b) a non-Euclidean function (2.20).

The projection operator is

$$[P_y]_{ij} = \frac{1}{R^2}\begin{cases} R^2 - y_i^2, & i = j \\ -y_i y_j, & i \neq j \end{cases} \tag{2.19}$$

so that again, evaluating the Jacobian $D\text{cp}$ at y on the surface with $|y|^2 = 1$ results in a projection onto the surface: $D\text{cp}(y) = P_y$.

3. We now construct a novel example of a non-Euclidean closest point function for a circle of radius R in $\mathbb{R}^2$, and show that this also satisfies the key property. In polar coordinates $x_1 = r\cos\theta$, $x_2 = r\sin\theta$, we can express a closest point function as

$$\text{cp}(x_1, x_2) = (\cos\theta_0, \sin\theta_0), \tag{2.20a}$$

$$\theta_0 = \theta - (r - R)^3, \tag{2.20b}$$

in an annulus with $0 < a_1 < r < a_2$. In this case,

$$D\text{cp}(x) = R\begin{pmatrix} \sin\theta_0\left(\frac{\sin\theta}{r} + 3(r-R)^2\cos\theta\right) & \sin\theta_0\left(-\frac{\cos\theta}{r} + 3(r-R)^2\sin\theta\right) \\ \cos\theta_0\left(-\frac{\sin\theta}{r} - 3(r-R)^2\cos\theta\right) & \cos\theta_0\left(\frac{\cos\theta}{r} - 3(r-R)^2\sin\theta\right) \end{pmatrix}$$

At the surface, $r = R$, and $\theta_0 = \theta$, so that

$$D\text{cp}(y) = \frac{1}{R^2}\begin{pmatrix} y_2^2 & -y_1 y_2 \\ -y_1 y_2 & y_1^2 \end{pmatrix} = P_y. \tag{2.21}$$

Note that this cp-function satisfies the property that lines of constant cp intersect the surface orthogonally (see Figure 2.4) as in Lemma 2.2.2.

2.2.4 Implicit closest point representation

Since the closest point function cp is a retraction, the restriction of cp to the surface is the identity map, $\text{cp}|_{\mathcal{S}} = id_{\mathcal{S}}$, or equivalently $\text{cp}(y) = y$, for all $y \in \mathcal{S}$.

Using the inverse of this property, a closest point function can be used to represent a surface. This is an implicit representation, where the surface is defined as the set of points x in the neighbourhood which satisfy $x = \text{cp}(x)$. Given a map $\text{cp} : B(\mathcal{S}) \to U$ from an open set $B(\mathcal{S}) \subset \mathbb{R}^n$ to some subspace $U \subset B(\mathcal{S})$, we can define

$$\mathcal{S} := \{x \in B(\mathcal{S}) \,|\, x = \text{cp}(x)\}. \tag{2.22}$$

If cp is a retraction, this leads to $\mathcal{S} = U$. Note that this will not necessarily result in a pure manifold (the resulting surface may have non-constant dimension).

The closest point representation can also be used to visualize the surface, by searching for the fixed points of this function [AMT+12]. In contrast to the representation using level sets, the surface is not required to be closed. A fast GPU ray-caster described in [AMT+12] is used to render some surfaces in later chapters.

2.2.5 Construction of closest point representations

In the remainder of the thesis, we usually assume that a closest point representation for the surface is known. However, in applications we often have a surface given by a parameterization (for example, in an interface problem), or a triangulated surface (in computer graphics). For these surfaces, a closest point representation will first have to be constructed. In fact, the value of the cp-function only needs to be computed at a discrete set of points on the Cartesian grid. Once this function is computed, it can be used to represent the surface, and need not be re-computed.

Much work has been done in the areas of computer graphics and computational geometry on computing shortest paths and distance maps, based on solving an Eikonal equation in the embedding space. A popular method for this is the fast marching method [Set96], which propagates distance information as a front moving outwards from the surface. Once the distance map $d(x_j)$ has been computed on the Cartesian grid, the Euclidean closest point function can be constructed as

$$\text{ecp}(x_j) := x_j - d(x_j) D_h d(x_j), \tag{2.23}$$

using a finite difference approximation $D_h d(x_j)$ to the gradient $\nabla d(x)$.

The regions in which the cp or distance functions are smooth can also be used to characterize the surface. In shape analysis, the *medial axis* is the set of points where the distance map becomes non-unique. In image processing, this is also referred to as the *topological skeleton* of the surface, and can be used to reconstruct the shape.

In general, the theory of viscosity solutions is required to represent solutions of the Eikonal equation. Hovever, since we only compute in a tubular neighbourhood of the surface, in which the distance map is smooth, we can consider smooth closest point functions.

2.3 Extension operators

The penalty method of this thesis is based on the introduction of extension operators, which map between function spaces on the surface and in the band. A contribution of this thesis is to consider the composition with the function cp explicitly as a linear operator. This simplifies notation, and in addition allows us to specify properties such as boundedness of the operator in certain norms, idempotence, and smoothness of the resulting functions, which are useful for the analysis of the embedding equation. These new properties are stated and proven in the following sections. In particular, Proposition 2.4.3 is a key statement underpinning the construction of the embedding equation.

2.3.1 Surface extension operator

Given a closest point function, we define the *extension operator* E_S which acts on surface functions, and returns a function on the embedding band $B(\mathcal{S})$.

Definition 2.3.1 (Surface Extension Operator)**.** *Let $\mathcal{S}$ be a C^l surface, and let $u \in C^l(\mathcal{S})$ be a scalar-valued function defined on $\mathcal{S}$. If* cp *is a closest point function as in Definition* 2.2.1*, then the closest point extension $v = E_S u$ is a function $v : B(\mathcal{S}) \to \mathbb{R}$ defined as*

$$v(x) = E_S u(x) := u(\mathrm{cp}(x)), \qquad x \in B(\mathcal{S}).$$

Operation on a vector-valued function is defined componentwise.

Lemma 2.3.2. *The surface extension operator $E_S : C^l(\mathcal{S}) \to C^l(B(\mathcal{S}))$ is a bounded linear operator in the max norm.*

Proof. For surface functions $u_1, u_2 : \mathcal{S} \to \mathbb{R}$ and $\alpha, \beta \in \mathbb{R}$, we have $E_S(\alpha u_1 + \beta u_2)(x) = [\alpha u_1 + \beta u_2](\mathrm{cp}(x)) = \alpha u_1(\mathrm{cp}(x)) + \beta u_2(\mathrm{cp}(x)) = \alpha E_S u_1(x) + \beta E_S u_2(x)$, so E_S is linear. To show boundedness, we have

$$||E_S u||_{\infty, B(\mathcal{S})} = \max_{x \in B(\mathcal{S})} E_S u(x) = \max_{x \in B(\mathcal{S})} u(\mathrm{cp}(x)) \leq \max_{y \in \mathcal{S}} u(y) = ||u||_{\infty, \mathcal{S}},$$

so that

$$||E_S||_\infty := \sup \left\{ ||E_S u||_{\infty, B(\mathcal{S})} : ||u||_{\infty, \mathcal{S}} \leq 1 \right\} \leq 1.$$

□

Lemma 2.3.3. *Let $\mathcal{S}$ be a compact C^2 hypersurface, and let $B_\alpha(\mathcal{S})$ be a band of width α, with $\alpha < (\max_{i=1,..,k} ||\kappa_i||_{\infty,\mathcal{S}})^{-1}$. Then there exist $c_1, c_2 \in \mathbb{R}$, such that the surface extension operator $E_S : C^l(\mathcal{S}) \to C^l(B(\mathcal{S}))$ satisfies*

$$c_1 \sqrt{\alpha} ||u||_{2,\mathcal{S}} \leq ||E_S u||_{2, B_\alpha(\mathcal{S})} \leq c_2 \sqrt{\alpha} ||u||_{2,\mathcal{S}}.$$

Proof. Using the local coordinate representation of Lemma 2.2.4,

$$x = \Phi(\mathrm{cp}(x), \mathrm{sd}(x)) := \mathrm{cp}(x) + \mathrm{sd}(x)\nu(\mathrm{cp}(x)), \tag{2.24}$$

and a change of variables, we have

$$\begin{aligned} ||E_S u||^2_{2, B_\alpha(\mathcal{S})} &= \int_{B_\alpha(\mathcal{S})} |E_S u|^2 dx \\ &= \int_{-\alpha}^{\alpha} \int_{\mathcal{S}} |u \circ \mathrm{cp}(s)|^2 \det[D\Phi] ds \, d\mathrm{sd}. \end{aligned}$$

From [GT01]

$$\det[D\Phi] = \prod_{i=1}^{n-1} (1 - \mathrm{sd}(x)\kappa_i(x)), \tag{2.25}$$

so there exist $c_1', c_2' \in \mathbb{R}$ such that $0 < c_1' \leq \det[D\Phi] \leq c_2' < 1$, provided that $\alpha < (\max_{i=1,..,k} ||\kappa_i||_{\infty,\mathcal{S}})^{-1}$, where $\alpha = \max_{x \in B_\alpha(\mathcal{S})} \mathrm{sd}(x)$. Therefore

$$\int_{-\alpha}^{\alpha} \int_{\mathcal{S}} |u \circ \mathrm{cp}|^2 \det[D\Phi] ds \, d\mathrm{sd} \leq 2c_2' \alpha \int_{\mathcal{S}} |u \circ \mathrm{cp}(s)|^2 ds = 2c_2' \alpha \int_{\mathcal{S}} |u(s)|^2 ds \tag{2.26}$$

so that

$$||E_S u||_{2, B_\alpha(\mathcal{S})} \leq c_2 \sqrt{\alpha} ||u||_{2,\mathcal{S}}, \tag{2.27}$$

and E_S is bounded in the L^2 norm. Similarly, we can use the lower bound on $\det[D\Phi]$ to show further that

$$c_1 \sqrt{\alpha} ||u||_{2,\mathcal{S}} \leq ||E_S u||_{2, B_\alpha(\mathcal{S})} \leq c_2 \sqrt{\alpha} ||u||_{2,\mathcal{S}} \tag{2.28}$$

□

Remark. Although not considered further here, the extension operator could be generalized to act on nonsmooth functions, such as functions in Sobolev spaces. For a smooth compact k-dimensional surface $\mathcal{S}$ in $\mathbb{R}^n$, there exists a bounded linear map $E_S : H^{\tau-(n-k)/2}(\mathcal{S}) \to H^\tau(\mathbb{R}^n)$, with $\tau > (n-k)/2$, such that $E_S u|_{\mathcal{S}} = u$ for all $u \in H^{\tau-(n-k)/2}(\mathcal{S})$ [FW12]. Note that the regularity of the resulting extended function depends on the codimension $n-k$ of the surface. We now show some properties of functions composed with the map cp.

Lemma 2.3.4. *For a smooth surface $\mathcal{S}$ and smooth closest point function* cp*, we have $u \in C^l(\mathcal{S}) \Rightarrow v \in C^l(B(\mathcal{S}))$, i.e., the extended function $v = E_S u$ is at least as smooth as the original surface function u.*

Proof. This follows since v is defined as the the composition of the smooth functions u and cp. □

A similar result holds, under conditions on $\mathcal{S}$, if the surface is not infinitely smooth:

Lemma 2.3.5. *For a C^r surface $\mathcal{S}$, and C^{r-1} closest point function* cp*, we have $u \in C^l(\mathcal{S}) \Rightarrow v \in C^l(B(\mathcal{S}))$ with $r \geq l+1$, i.e., the extended function $v = E_S u$ is at least as smooth as the original surface function u.*

Proof. Since the extended function v is the composition of a C^l function u with a C^{r-1} function cp, the smoothness of v will be $\min(l, r-1)$. Choosing $r \geq l+1$ leads to $v \in C^l(B(\mathcal{S}))$. □

Note that this case includes the Euclidean closest point function ecp. We can therefore always construct a closest point function that will allow the extended function to be as smooth as the original surface function, provided that the surface has at least one additional order of smoothness.

The range of E_S is $\text{range}(E_S) := \{v \in C^l(B(\mathcal{S})) : v = E_S u \text{ for some } u \in C^l(\mathcal{S})\}$. We will later restrict solutions of differential equations to lie in this linear subspace. The null space of E_S is the zero function on $\mathcal{S}$. This follows since $\mathcal{S}$ is contained in $B(\mathcal{S})$, so if $v = E_S u$ is identically zero, then it is also zero on the surface, and therefore u is also zero on $\mathcal{S}$.

2.3.2 Restriction operator

We now define an operator R which restricts band functions to the surface.

Definition 2.3.6 (Restriction Operator)**.** *If $v : B(\mathcal{S}) \to \mathbb{R}$ is a C^l function defined in the embedding space, then $u = Rv = v|_{\mathcal{S}}$ is a function $u : \mathcal{S} \to \mathbb{R}$ which is equal to v for points on the surface,*

$$u(y) = Rv(y) := v(y), \qquad y \in \mathcal{S}. \tag{2.29}$$

Provided that the surface is at least C^l, the restricted function $u(y)$ is at least as smooth as $v(x)$, i.e. $v \in C^l(B(\mathcal{S})) \Rightarrow u \in C^l(\mathcal{S})$.

Remark. We could again generalize this to a trace operator applying to functions in Sobolev spaces. The trace operator maps $H^\tau(\mathbb{R}^n)$ onto $H^{\tau-(n-k)/2}(\mathcal{S})$ [FW12]. Importantly, we lose half an order of regularity, multiplied by the codimension, whenever we take the restriction.

2.3.3 Band extension operator

The surface extension operator above (Definition 2.3.1) can then be generalized to act on functions defined on all of $B(S)$ by operating on the restriction of the function to the surface $\mathcal{S}$, i.e. $E = E_S R$.

Definition 2.3.7 (Band Extension Operator)**.** *If $v : B(\mathcal{S}) \to \mathbb{R}$ is a function defined in the embedding space, then the closest point extension $w = Ev$ is a function $w : B(\mathcal{S}) \to \mathbb{R}$ defined as*

$$w(x) = Ev(x) := E_S R v(x) = E_S(v\,|_S) = v(\mathrm{cp}(x)), \qquad x \in B(\mathcal{S}). \tag{2.30}$$

Lemma 2.3.8. *The band extension operator $E : C^l(B(\mathcal{S})) \to C^l(B(\mathcal{S}))$ is continuous in the max norm.*

Proof. Linearity again follows from the composition with cp. For boundedness, we have

$$||Ev||_{\infty,B(\mathcal{S})} = \max_{x\in B(\mathcal{S})} Ev(x) = \max_{x\in B(\mathcal{S})} v(\mathrm{cp}(x)) \leq \max_{y\in\mathcal{S}} v(y) \leq \max_{x\in B(\mathcal{S})} v(x) = ||v||_{\infty,B(\mathcal{S})},$$

so that

$$||E||_{\infty,B(\mathcal{S})} := \sup\left\{||Ev||_{\infty,B(\mathcal{S})} : ||v||_{\infty,B(\mathcal{S})} \leq 1\right\} \leq 1.$$

□

Lemma 2.3.9. *For a smooth surface $\mathcal{S}$, and smooth closest point function* cp*, we have $v \in C^l(B(\mathcal{S})) \Rightarrow w \in C^l(B(\mathcal{S}))$ i.e., the new band function $w = Ev$ is at least as smooth as the original band function v.*

Proof. The extended band function w is the composition of the band function v with the smooth function cp, resulting in a function which is as smooth as v. □

As in the surface extension case for E_S, a similar result holds for surfaces which are not smooth.

Lemma 2.3.10. *For a C^r surface $\mathcal{S}$, and C^{r-1} closest point function* cp*, we have $v \in C^l(B(\mathcal{S})) \Rightarrow w \in C^l(B(\mathcal{S}))$ with $r \geq l + 1$, i.e., the extended function $w = Ev$ is at least as smooth as the original band function v.*

Proof. The extended band function w is the composition of a C^l function v with a C^{r-1} function cp, so the smoothness of w will be $\min(l, r-1)$. Choosing $r \geq l + 1$ leads to $w \in C^l(B(\mathcal{S}))$. □

Again, this covers the important case of the Euclidean distance closest point function. This proof will be used in later chapters to ensure that, provided the surface is smooth enough, we can always construct a cp function allowing the extended function to be as smooth as the original band function.

2.4 Properties of extension operators

We examine some simple properties of the surface and band extension operators E_S and E, which will be used frequently in the remainder of the thesis.

Proposition 2.4.1. *The restriction of a surface extension is the identity on the surface, i.e., for a smooth surface $\mathcal{S}$ and surface function $u \in C^l(\mathcal{S})$,*

$$RE_S u(y) = u(y), \quad y \in \mathcal{S}. \tag{2.31}$$

Proof. For a point $x \in B(\mathcal{S})$, we have $E_S u(x) = u(\mathrm{cp}(x))$. Restricting this to points $y \in \mathcal{S}$ on the surface, $RE_S u(y) = u(\mathrm{cp}(y)) = u(y)$. □

Proposition 2.4.2. *The extension operator E is idempotent.*

Proof. Let $x \in B(\mathcal{S})$. Since $\text{cp}(x) \in \mathcal{S}$, and the retraction cp is equal to the identity operator on the surface, we have $\text{cp}(\text{cp}(x)) = \text{cp}(x)$. Hence, from the definition of the operator E,

$$E^2 v(x) = v(\text{cp}(\text{cp}(x))) = v(\text{cp}(x)) = Ev(x). \tag{2.32}$$

□

Since E is both linear and idempotent, it is a projection. Acting on the function space $C^l(B(\mathcal{S}))$, it is the projection onto the subspace $\{v \in C^l(B(\mathcal{S})) : v = Ew \text{ for some } w \in C^l(B(\mathcal{S}))\}$.

Remark. From this, it follows that if a function v can be written as the band extension of another function ($v = Ew$) then v must be its own extension ($v = Ev$). This property will be used frequently in the following chapters.

Proposition 2.4.3. *If v is the extension of a surface function, then it is also the extension of itself,*

$$v = E_S u \Rightarrow v = Ev. \tag{2.33}$$

Proof. Using the fact that $RE_S = id_{\mathcal{S}}$, and the associativity of the operators, we have

$$Ev = E_S Rv = E_S RE_S u = E_S u = v. \tag{2.34}$$

□

Remark. We therefore refer to a function v in the band $B(\mathcal{S})$ as a *closest point extension* if $Ev = v$.

Proposition 2.4.4. *The surface extension operator E_S is one-to-one. If $u, w : \mathcal{S} \to \mathbb{R}$ are smooth surface functions, then*

$$E_S u = E_S w \Rightarrow u = w. \tag{2.35}$$

Proof. Let $E_S u(x) = E_S w(x)$ for all $x \in B(\mathcal{S})$. Then for a surface point $y \in \mathcal{S} \subset B(\mathcal{S})$, we also have $E_S u(y) = E_S w(y)$. Since $\text{cp}(y) = y$ for points on the surface,

$$u(y) = u(\text{cp}(y)) = E_S u(y) = E_S w(y) = w(\text{cp}(y)) = w(y), \quad \forall y \in \mathcal{S}. \tag{2.36}$$

2.5 Closest point principles

The closest point method relies on principles relating differential operators on the surface and in the embedding band. Although not originally stated in this way, the closest point principles of [RM08, MM12] can be reformulated using the extension and restriction operators. Note that the Closest Point Theorem (2.5.3) is stated in [MM12] for a smooth surface $\mathcal{S}$; here we generalize to surfaces which are only C^2, using the properties derived in Section 2.3.

Theorem 2.5.1 (Gradient Principle)**.** *Let $\mathcal{S}$ be a C^2 surface, and let $u \in C^1(\mathcal{S})$ be a scalar-valued surface function on $\mathcal{S}$. If E_S is a closest point extension operator according to Definition 2.3.1, and R is a restriction to the surface as in Definition 2.3.6, then*

$$\nabla_{\mathcal{S}} u = R\nabla[E_S u].$$

Equivalently, we can write $E_S \nabla_{\mathcal{S}} u = E\nabla[E_S u]$.

Theorem 2.5.2 (Divergence Principle)**.** *Let $\mathcal{S}$ be a C^2 surface, and let $g \in C^1(\mathcal{S}, \mathbb{R}^n)$ be a vector field on $\mathcal{S}$. Then*

$$\mathrm{div}_{\mathcal{S}} g = R\,\mathrm{div}\,[E_S g].$$

As above, this can be written equivalently as $E_S \mathrm{div}_{\mathcal{S}} g = E\,\mathrm{div}\,[E_S g]$.

The above principles follow as direct consequences of the Closest Point Theorem.

Theorem 2.5.3 (Closest Point Theorem [MM12])**.** *Let $\mathcal{S}$ be a C^2 surface, and let $f \in C^1(\mathcal{S}, \mathbb{R}^m)$ be a vector-valued surface function on $\mathcal{S}$. If E_S is a closest point extension operator according to Definition 2.3.1, and R is a restriction to the surface as in Definition 2.3.6, then*

$$D_{\mathcal{S}} f = RD[E_S f].$$

Proof. Recall that

$$D_{\mathcal{S}} f(y) := D\bar{f}(y) P_y,$$

where $\bar{f} : B(\mathcal{S}) \to \mathbb{R}^m$ is a regular extension of f. Now

$$D[E_S f](x) = D[E\bar{f}](x) = D\bar{f}\big|_{\mathrm{cp}(x)}\, D\mathrm{cp}(x). \tag{2.37}$$

Restricting to the surface,

$$D[E_S f](y) = D\bar{f}\big|_{\mathrm{cp}(y)} D\mathrm{cp}(y) = D\bar{f}(y)P_y = D_{\mathcal{S}} f(y) \tag{2.38}$$

using the fact that $\mathrm{cp}(y) = y$ for $y \in \mathcal{S}$, and that $D\mathrm{cp}(y) = P_y$ at the surface. □

This theorem can also be expressed in the band as $E_S D_{\mathcal{S}} f = E D[E_S f]$.

The closest point principles from [RM08, MM12] apply for any codimension of $\mathcal{S}$, and any choice of closest point function cp. However, they hold only when evaluated at a point on the surface (due to the fact that $D\mathrm{cp}(y) = P_y$ only locally at $y \in \mathcal{S}$).

We note here that these principles can be generalized to expressions for the surface differential operators which apply throughout the band. These expressions involve the curvature of the surface, and only apply in the case of surfaces of codimension 1 (hypersurfaces).

For example, let cp be the Euclidean closest point function. We have from (2.7) and (2.9) that

$$\begin{aligned} D\mathrm{cp}(x) &= I - \nabla \mathrm{sd}(x) \nabla \mathrm{sd}(x)^T - \mathrm{sd}(x) D^2 \mathrm{sd}(x) \\ &= P_x - \mathrm{sd}(x)\mathcal{H}(x), \end{aligned} \tag{2.39}$$

where $\mathcal{H}(x) = D^2 \mathrm{sd}(x)$ is the Weingarten map (defined in Section 1.4.3). Then we obtain for the surface gradient (following [DE13])

$$\begin{aligned} \nabla[E_S u](x) &= \nabla[E E_S u](x) = D\mathrm{cp}(x) E \nabla[E_S u](x) \\ &= (P_x - \mathrm{sd}(x)\mathcal{H}(x)) E_S \nabla_{\mathcal{S}} u(x) \\ &= (I - \mathrm{sd}(x)\mathcal{H}(x)) E_S \nabla_{\mathcal{S}} u(x), \end{aligned} \tag{2.40}$$

using Theorem 2.5.1 and the fact that $E_S \nabla_{\mathcal{S}} u = P_x E_S \nabla_{\mathcal{S}} u$, since the surface gradient of a function lies in the tangent space of the surface. This formula (2.40) is now defined throughout the embedding band, and is therefore more general than Theorem 2.5.1. At the surface, $\mathrm{sd}(y) = 0$, so applying a restriction operator, we again obtain

$$R\nabla[E_S u] = R E_S \nabla_{\mathcal{S}} u = \nabla_{\mathcal{S}} u, \tag{2.41}$$

as in Theorem 2.5.1.

2.5.1 More general differential operators

The gradient and divergence principles above can be combined to apply to a wider class of operators [RM08, MM12]. In general, we assume that A_S is any surface-spatial differential operator such that the above principles can be applied to give a corresponding operator A.

Definition 2.5.4 (Operators with a closest point principle)**.** *If A is a Cartesian differential operator, and A_S is the corresponding differential operator acting over the surface, then we say this operator obeys a* closest point principle*, if*

$$RA(t, x, E_S u) = A_S(t, y, u), \qquad y \in \mathcal{S}, x \in B(\mathcal{S}). \tag{2.42}$$

That is, the operator A (acting on functions on $B(\mathcal{S})$) is an analogue of the operator A_S (acting on functions on $\mathcal{S}$) where A has a standard differential operator wherever A_S has a surface differential operator. When restricted to the surface, the operators coincide, provided that the Cartesian operator is applied to the extended function $E_S u$.

An important example of this class is the Laplace–Beltrami operator in the case of the Euclidean closest point function.

Theorem 2.5.5 (Laplacian Principle)**.** *Let $\mathcal{S}$ be a C^3 surface, and let $u \in C^2(\mathcal{S})$. If E_S is the particular extension operator corresponding to Euclidean distance to the surface (Definition 2.2.3), then*

$$\Delta_{\mathcal{S}} u = R\Delta[E_S u].$$

The proof for general codimension is given in [MM12]. Here, we present an alternative proof for the case where $\mathcal{S}$ is a hypersurface.

Proof. Following [XZ03], we express the Laplace–Beltrami operator in terms of the normal gradient and mean curvature. Recall that for hypersurface $\mathcal{S}$, we have (see Section 1.3.3),

$$\nabla_{\mathcal{S}} \bar{u}(x) = (I - \nu(x)\nu(x)^T)\nabla \bar{u}(x), \tag{2.43}$$

where the vector $\nu(x)$ is the unit normal vector to the level set at the point $x \in B(\mathcal{S})$. Note that when defined in the embedding space, the surface gradient is the component

of the gradient which is tangent to the level set at x. Then the Laplace–Beltrami operator is

$$\begin{aligned}\Delta_{\mathcal{S}}\bar{u}(x) &= \operatorname{tr}\left((I-\nu\nu^T)D(\nabla\bar{u}-(\nu\cdot\nabla\bar{u})\nu)\right), \\ &= \underbrace{\operatorname{tr}\left(D\left(\nabla\bar{u}-\frac{\partial\bar{u}}{\partial\nu}\nu\right)\right)}_{\text{(I)}} - \underbrace{\operatorname{tr}\left(\nu^T D(\nabla\bar{u}-(\nu\cdot\nabla\bar{u})\nu)\nu\right)}_{\text{(II)}}, \end{aligned} \tag{2.44}$$

where we have used that the gradient in the direction normal to the level set is given by $\frac{\partial\bar{u}}{\partial\nu} = \nu\cdot\nabla\bar{u}$. Then

$$\begin{aligned}(I) &= \operatorname{div}\left(\nabla\bar{u}-\frac{\partial\bar{u}}{\partial\nu}\nu\right) \\ &= \Delta\bar{u}-\frac{\partial\bar{u}}{\partial\nu}(\operatorname{div}\nu)-\nu\cdot\nabla\left(\frac{\partial\bar{u}}{\partial\nu}\right), \\ &= \Delta\bar{u}-\kappa\frac{\partial\bar{u}}{\partial\nu}-\frac{\partial^2\bar{u}}{\partial\nu^2}, \end{aligned} \tag{2.45}$$

where the mean curvature is $\kappa = \operatorname{div}\nu$. The remaining term is

$$\begin{aligned}(II) &= \operatorname{tr}\left(\nu^T(D^2\bar{u})\nu-\nu^T D((\nu\cdot\nabla\bar{u})\nu)\nu\right), \\ &= \operatorname{tr}\left(\nu^T(D^2\bar{u})\nu+\nu^T((\nu\cdot\nabla\bar{u})D\nu)\nu-\nu^T(D^2\bar{u})\nu\right), \\ &= \operatorname{tr}((\nu\cdot\nabla\bar{u})\nu^T(D\nu)\nu), \\ &= 0, \end{aligned} \tag{2.46}$$

since ν is a unit normal, so $\nu^T\nu = 1$, and therefore $(D\nu)\nu = \nabla(\nu^T\nu) = 0$.

Then we have for the surface Laplacian,

$$\begin{aligned}\Delta_{\mathcal{S}}\bar{u}(x) &= (I)-(II) \\ &= \Delta\bar{u}(x)-\frac{\partial^2\bar{u}}{\partial\nu^2}(x)-\kappa\frac{\partial\bar{u}}{\partial\nu}(x), \quad x\in B(\mathcal{S}). \end{aligned} \tag{2.47}$$

Since the extended function $\bar{u} = E_S u$ is constant in the normal direction, we have that

$$\frac{\partial\bar{u}}{\partial\nu}(y) = 0 \quad \text{and} \quad \frac{\partial^2\bar{u}}{\partial\nu^2}(y) = 0, \quad y\in\mathcal{S}. \tag{2.48}$$

After inserting (2.48) into (2.47), and evaluating the equation at a point $y \in \mathcal{S}$, the equivalence of the surface and Cartesian Laplacian operators at the surface is established. □

Note that the statement of Theorem 2.5.5 requires that the operator E_S corresponds to the Euclidean closest point function. For a general extension operator E_S,

we satisfy $\frac{\partial \bar{u}}{\partial \nu}(y) = 0$, since the extended function $\bar{u} = E_S u$ is required by definition to be constant in the normal direction at least locally at the surface. However, the proof also requires that $\frac{\partial^2 \bar{u}}{\partial \nu^2}(y) = 0$ (2.48), and hence we need a property of higher derivatives of the extended function. Since the Euclidean closest point function is constant in the normal direction throughout the band, this is satisfied for any order of derivative. This property would also be satisfied, for example, by the non-Euclidean function of (2.20).

The Laplacian principle is extended in [MM12] to more general diffusion operators A_S, where the surface differential operator involves an $n \times n$ diffusion tensor a, with

$$A_S = \operatorname{div}_{\mathcal{S}}(a \nabla_{\mathcal{S}} u). \tag{2.49}$$

A key feature of the Laplacian operator is that $\nabla_{\mathcal{S}} u(y) \in \mathcal{T}_y \mathcal{S}$, from the definition of the surface gradient as a projection onto the tangent space at y. If the diffusion tensor a is such that $a \nabla_{\mathcal{S}} u \in T_y \mathcal{S}$, then the differential operator A_S also satisfies a closest point principle, as in the case of the Laplacian. Diffusion operators involving this property are studied further in Chapter 5.

More general equations

The principles above could be applied to more general equations involving combinations of differential operators, as in [MR08]. For example, we could consider an operator F such that

$$\begin{aligned} &RF(t, x, E_S u(x), \nabla[E_S u(x)], \operatorname{div}[E_S g(x)], ...) \\ &\qquad = F(t, y, u(y), \nabla_{\mathcal{S}} u(y), \operatorname{div}_{\mathcal{S}} g(y), ...), \qquad y \in \mathcal{S}, x \in B(\mathcal{S}). \end{aligned} \tag{2.50}$$

For simplicity of notation, in this work we restrict the discussion to the case of a single differential operator A_S, however the theory extends to equations involving combinations of differential operators. This includes, for example, the case of the biharmonic operator considered in Section 4.7.

2.5.2 Examples

The closest point principles are easily demonstrated with the simple case of a circle of radius R in $\mathbb{R}^2$. A function $u : \mathcal{S} \to \mathbb{R}$ can be parameterized as $u = u(\theta)$. Using a Euclidean closest point map, the extended function $v = E_S u$ is described by $v(r, \theta) = u(\theta)$. The surface gradient can be written as $\nabla_{\mathcal{S}} u = \frac{1}{R} u_\theta$. Taking the Cartesian gradient of v,

$$\nabla v(r, \theta) = v_r + \frac{1}{r} v_\theta = \frac{1}{r} v_\theta, \tag{2.51}$$

so that restricting to the surface with $r = R$ gives

$$\nabla v(R,\theta) = \frac{1}{R} u_\theta(\theta) = \nabla_{\mathcal{S}} u(\theta). \tag{2.52}$$

Similarly, the Laplace–Beltrami operator is given by $\Delta_{\mathcal{S}} u = \frac{1}{R^2} u_{\theta\theta}$. The Cartesian Laplacian of v is

$$\Delta v(r,\theta) = \frac{1}{r} v_r + v_{rr} + \frac{1}{r^2} v_{\theta\theta} = \frac{1}{r^2} v_{\theta\theta}, \tag{2.53}$$

so that

$$\Delta v(R,\theta) = \frac{1}{R^2} u_{\theta\theta}(\theta) = \Delta_{\mathcal{S}} u(\theta). \tag{2.54}$$

2.6 The closest point method

The gradient and divergence principles above are used to convert surface PDEs to equations with standard Cartesian differential operators. The explicit formulation of the closest point method introduced by Ruuth and Merriman [RM08] is described here with an example.

Suppose that the function $u : \mathcal{S} \to \mathbb{R}$ satisfies the surface diffusion equation

$$u_t(y) = \Delta_{\mathcal{S}} u(y) \quad y \in \mathcal{S}. \tag{2.55}$$

On the surface, the principles allow us to replace the Laplace–Beltrami operator with a Laplacian:

$$u_t(y) = \Delta \bar{u}(y), \quad y \in \mathcal{S}, \tag{2.56}$$

provided that the particular extended function $\bar{u} := E_S u$ is used.

We now assume that this equation holds throughout the entire band, and construct the embedding equation

$$v_t(x) = \Delta v(x), \quad x \in B(\mathcal{S}), \tag{2.57}$$

for a function $v : B(\mathcal{S}) \to \mathbb{R}$. If the initial condition is chosen to be $v^0 = E_S u^0$, the equation (2.57) will initially be equivalent on the surface to (2.56). However, after $t = 0$ this will no longer be the case, since v will not necessarily satisfy $v = Ev$. When using an explicit time-stepping scheme, the equation is therefore re-initialized with a closest point extension $v = Ev$ after each time step.

To construct the numerical scheme, the computational band is discretized with a uniform Cartesian grid. The matrix L represents a standard finite difference discretization of the linear operator Δ, and the operators E and R are discrete versions of the extension and restriction operators E and R. The construction of these discrete operators on the grid involves interpolation (for further details see Chapter 4).

Algorithm 1: Explicit closest point method [RM08, Mac08]

Data: U^0, initial condition on surface
extension matrix E, differentiation matrix L, restriction matrix R
Result: U, discrete solution on surface at time T_f
begin
  `/* Initial condition in band */`
  $V \leftarrow \mathsf{E}U^0$
  for $t = 1$ **to** $T_f/\Delta t$ **do**
    `/* Perform an explicit time step in the band */`
    $V \leftarrow V + \Delta t \mathsf{L}V$
    `/* Re-extend */`
    $V \leftarrow \mathsf{E}V$
  end
  `/* Restrict solution to discrete points on the surface */`
  $U \leftarrow \mathsf{R}V$
end

2.6.1 The implicit closest point method

An implicit version of the closest point method was introduced in [MR09], in which the equation (2.57) is discretized with an implicit time-stepping scheme. Applying first a discretization in space would yield the semi-discrete form

$$V_t = \tilde{\mathsf{M}}V = \mathsf{LE}V. \tag{2.58}$$

For stability, the diagonal terms of the matrix are not interpolated at the closest points. Instead, in [MR09], a new matrix is formed,

$$\mathsf{M} = \text{stab}(\mathsf{L}, \mathsf{E}) := \text{diag}\,\mathsf{L} + (\mathsf{L} - \text{diag}\,\mathsf{L})\mathsf{E}, \tag{2.59}$$

where $\text{diag}\,\mathsf{L}$ is the diagonal matrix containing the diagonal entries of L. An implicit scheme is then used to solve $V_t = \mathsf{M}V$.

This will be shown to be a special case of the general algorithm introduced in Chapter 3.

2.6.2 Remark on boundaries

We have required above that the surface $\mathcal{S}$ is compact, and hence is a submanifold without boundary. However, the computational band $B(\mathcal{S})$ has a non-empty boundary $\partial B(\mathcal{S})$, so boundary conditions need to be defined for the numerical scheme. In the case of the explicit closest point method, the extension step $v = Ev$ lifts values on the surface to the boundary $\partial B(\mathcal{S})$ after each time step. Provided that the band

Algorithm 2: Implicit closest point method [MR09]

Data: U^0, initial condition on surface
extension matrix E, differentiation matrix L, restriction matrix R
Result: U, discrete solution on surface at time T_f
begin
  `/* Initial condition in band */`
  $V \leftarrow \mathsf{E}U^0$
  `/* Create diagonally-stabilized implicit matrix */`
  $\mathsf{M} \leftarrow \operatorname{diag}\mathsf{L} + (\mathsf{L} - \operatorname{diag}\mathsf{L})\mathsf{E}$
  for $t = 1$ **to** $T_f/\Delta t$ **do**
    `/* Solve one implicit time step in the band */`
    $V \leftarrow (\mathsf{I} - \Delta t\mathsf{M})\backslash V$
  end
  `/* Restrict solution to discrete points on the surface */`
  $U \leftarrow \mathsf{R}V$
end

is sufficiently large (the differential operator stencil at the surface must not intersect the boundary), the choice of boundary condition does not influence the result, since any boundary values will be overwritten by this step. In the implicit case, a boundary condition will be implied by the constraint that $v = Ev$ everywhere, including at the boundary. This is discussed further in Chapters 3 and 4.

The closest point method has also been applied in practice to the case where $\mathcal{S}$ is a submanifold with non-empty boundary. It is shown in [RM08] that applying the standard Euclidean cp representation leads to first-order homogeneous Neumann conditions. In this case, grid points in the normal direction to the boundary all map to the same surface boundary point, meaning that the function cp is constant in the normal direction at the boundary.

The accuracy can be increased with an adaptation of the closest point function near boundaries, using a new function $\bar{\text{cp}}$, which reflects the cp function across the boundary point. It is shown in [MBR11] that this can be used to impose homogeneous Dirichlet or Neumann conditions, with at least second-order accuracy.

2.6.3 Discretization of extensions

The closest point method is based on the fact that the extension operator E is idempotent. However, the discrete matrix E is not idempotent, due to the introduction of an interpolation step. Chapter 4 describes the discretization further, and investigates the matrix properties of the discrete operators.

Chapter 3

A Constrained Embedding Equation

In this chapter, we derive a modified formulation of the closest point embedding equation for evolution PDEs, and show that there is a one-to-one correspondence between solutions of the embedding and surface equations. This formulation is simple to discretize and solve numerically using a standard method of lines approach. This generalizes the stabilized implicit method of [MR09], and has the advantage that it can be adapted to a very general class of problems. An appropriate explicit or implicit time-stepping scheme can be used, depending on the particular problem considered. The new method retains the advantages of the original closest point method with respect to simplicity of the numerical scheme and generality of dimension and codimension of the surface.

The chapter begins with an overview of the new formulation using an example. We then define a system of embedding equations, and show that this is consistent with the original surface PDE. This system is reduced to a single equation involving a penalty term, and a method of lines discretization is presented. Due to the introduction of extension operators in the embedding equation, the equation solved in the band is not a PDE, however we are able to apply standard discretizations for the differential operators, while approximating the extension operators using interpolation. A full discussion of the discretization appears in a later chapter.

3.1 Defining the embedding equation

Rather than alternating between time-steps and re-extensions as in the explicit closest point method of [RM08] (see Section 2.6), we investigate an alternative approach, in

which a single equation can be solved throughout the entire embedding space, for all time, without separate extension steps. This is achieved by creating a modified embedding equation with a constraint.

The solution of a given surface evolution PDE is a function u which is defined only for points on the surface. We consider instead the function $v = u \circ \text{cp}$, which is defined for all points in a band surrounding the surface. Based on the gradient and divergence principles described in Chapter 2, we formulate a new equation for v. The constraint or side condition that v is a closest point extension is enforced by adding a penalty term to the equation. We show that solving this new equation throughout the embedding band to a given time t, and then restricting to the surface, results in a solution of the original surface PDE at time t.

If the resulting Cartesian differential operators and extension operators are discretized in space as in [MR09], we obtain a system of ordinary differential equations in the computational band. Thus we have a new method of lines formulation of the closest point method, which can be implemented using either explicit or implicit time-stepping.

3.1.1 Example - diffusion equation

We first illustrate the method with the example of Section 1.1.1, before giving a more detailed derivation. Let $\mathcal{S}$ be a smooth compact surface embedded in $\mathbb{R}^n$, and u a scalar function on $\mathcal{S}$. Consider again the surface diffusion equation

$$u_t = \Delta_{\mathcal{S}} u,$$

subject to an initial condition u_0.

Let $\text{cp} : B(\mathcal{S}) \to \mathcal{S}$ be the Euclidean closest point function as in Definition 2.2.3, which maps points in the band to points on the surface. Recall that the surface *extension* operator $E_{\mathcal{S}}$ is then defined as $E_{\mathcal{S}} u(x) = u \circ \text{cp}(x)$.

The surface differential operator (Laplace–Beltrami operator) may be replaced by a standard Laplacian using the principles in Section 2.5

$$u_t = R\Delta[E_{\mathcal{S}} u] \text{ on } \mathcal{S}.$$

This equation is valid only for points $y \in \mathcal{S}$, since the left-hand side u_t is defined only on the surface. In order to obtain an equation defined throughout the entire band $B(\mathcal{S})$, we perform an extension on both sides of the equation

$$E_S u_t = E_S R\Delta[E_S u] = E\Delta[E_S u] \text{ on } B(\mathcal{S}).$$

We now define a function $v = E_S u$ in the embedding space. Since the operator E_S is independent of t, the previous equation can be rewritten as

$$v_t = E\Delta v \text{ on } B(\mathcal{S}),$$

subject to the condition $v = E_S u$. But if v is the extension of u then v must also be its own extension (see also Proposition 2.4.3) and we obtain a system of two equations in v:

$$v_t = E\Delta v, \tag{3.1a}$$

$$v = Ev. \tag{3.1b}$$

We will show that the solutions v of this system, when restricted to the surface $\mathcal{S}$, agree with the solutions u of the original equation; i.e., $u = v|_{\mathcal{S}}$.

This system could then be approximated using the two-step method of [RM08], where a single time step of the first equation is carried out, and then the side condition imposed by extending the data $v(\cdot, t_k)$ off the surface.

We propose an alternative method, in which a single equation is solved. The side condition is added to the equation, with a constant multiplication factor γ,

$$v_t = E\Delta v - \gamma(v - Ev). \tag{3.2}$$

This equation forms the basis for the method of lines. We show that, although the penalty would appear to only weakly enforce the constraint, in fact the solutions of this single equation agree with the solutions of the system (3.1), for any non-zero choice of the penalty parameter γ. In practice the choice of γ affects the resulting numerical methods, as investigated in Section 3.3 and later chapters.

3.2 Equivalence of surface and embedding equations

Given a PDE defined on a surface, we will construct an equation defined in the embedding band, and show that there is a one-to-one correspondence between solutions of this and the original surface PDE. In contrast to previous formulations, the embedding equation is satisfied for all time throughout the computational band, not only on the surface. The embedding equation is defined through the application of an extension operator to the original equation.

We now show that two problems, one defined only on the surface $\mathcal{S}$, and one defined in the band $B(\mathcal{S})$, have the same solutions when restricted to the surface.

Problem 3.2.1 (Surface Evolution PDE)**.** *Given a smooth closed surface $\mathcal{S}$ in $\mathbb{R}^n$, let $u : \mathcal{S} \times (0,T) \to \mathbb{R}$, be a smooth solution of the PDE,*

$$u_t = A_S(t, y, u) \qquad u(y, 0) = u_0(y), \qquad y \in \mathcal{S},\ t \in (0,T)$$

where $A_S(t, y, u)$ is a linear or nonlinear surface differential operator of the class defined in (2.42).

Problem 3.2.2 (Embedding Equation)**.** *Given a neighbourhood $B(\mathcal{S}) \subset \mathbb{R}^n$ of $\mathcal{S}$, let $v : B(\mathcal{S}) \times (0,T) \to \mathbb{R}$ satisfy the system of equations*

$$v_t = EA(t, x, v) \tag{3.3a}$$

$$v = Ev, \qquad x \in B(\mathcal{S}),\ t \in (0,T) \tag{3.3b}$$

with initial condition $v(x, 0) = v_0(x)$. Here, $A(t, x, v)$ is a spatial differential operator on $B(\mathcal{S})$ defined from A_S as in (2.42).

Remark Note that no additional boundary conditions have been explicitly specified for Problem 3.2.2. By (3.3b), the solution everywhere off the surface (including at the boundary $\partial B(\mathcal{S})$) is determined by values on the surface, so we do not have the freedom to choose a boundary condition. Imposing Dirichlet boundary conditions, for example, could make the problem ill-posed. Indeed, since the embedding equation (3.3a) is no longer a PDE due to the introduction of extension operators, boundary conditions many not be required in order to guarantee existence and uniqueness of solutions.

However, the constraint that v is a closest point extension can be seen as playing the role of a boundary condition. The equation $v = Ev$ specifies that the gradient of the solution in the direction normal to the surface is zero, so we have

$$\nabla v \cdot E\nu = 0, \quad \text{on } \partial B(\mathcal{S}), \tag{3.4}$$

where $\nu(y)$ is the normal to the surface, and $E\nu(x) = \nu(\mathrm{cp}(x))$ is the extended normal field. This is an oblique Neumann boundary condition on $\partial B(\mathcal{S})$.

Theorem 3.2.3. *Let $\mathcal{S}$ be a smooth compact surface embedded in $\mathbb{R}^n$, and let $B(\mathcal{S}) \subset \mathbb{R}^n$ be a neighbourhood of the surface, such that there exists a closest point function* $\mathrm{cp} : B(\mathcal{S}) \to \mathcal{S}$.

If $u : \mathcal{S} \times (0,T) \to \mathbb{R}$ is a sufficiently smooth solution of the surface PDE (Problem 3.2.1), then the function $v : B(\mathcal{S}) \times (0,T) \to \mathbb{R}$ defined by $v = E_S u$ satisfies the embedding equation (Problem 3.2.2). Conversely, if v is a solution of Problem 3.2.2, then the restriction of v to $\mathcal{S}$ is a solution of Problem 3.2.1.

Proof. Let $v(x,t) = E_S u(x,t)$, where u satisfies Problem 3.2.1. Then

$$v_t = \partial_t(E_S u) = E_S(u_t) = E_S(A_S(t,y,u)),$$

from the surface PDE, and the fact that E_S is time-independent. Now, using (2.42) and the definition of E, we have that

$$E_S(A_S(t,y,u)) = E_S(RA(t,x,E_S u)) = EA(t,x,E_S u) = EA(t,x,v), \qquad x \in B(\mathcal{S}),$$

so the first equation (3.3a) is satisfied. The second (3.3b) follows from

$$Ev = EE_S u = E_S u = v.$$

Uniqueness follows from the fact that v agrees with u on $\mathcal{S}$, and that the off-surface values are uniquely defined by $v(x,t) = Ev(x,t)$. The smoothness of v is determined by the smoothness of the surface and smoothness of u (see Section 2.3).

For the converse, suppose that v is a solution of Problem 3.2.2, and let u be the restriction of v to the surface, $u = v|_{\mathcal{S}}$. Then

$$\partial_t u = \partial_t v|_{\mathcal{S}} = EA(t,x,v)|_{\mathcal{S}}.$$

Since the extension operator leaves values on the surface unchanged, and using the closest point principles on A,

$$\partial_t u = EA(t,x,v)|_{\mathcal{S}} = RA(t,x,v) = A_{\mathcal{S}}(t,y,v|_{\mathcal{S}}) = A_{\mathcal{S}}(t,y,u),$$

so u is a solution of Problem 3.2.1 as required. □

Remark on smoothness. We have assumed above that the surface $\mathcal{S}$ is smooth, or $\mathcal{C}^\infty$, but this is not strictly necessary. If the differential operator is r-th order, then the surface needs to be at least $\mathcal{C}^r$ for the function u to be defined. However, since the cp-function is one order less smooth than the surface, the function $v = E_S u$ will also be one order less smooth. We therefore require that the surface $\mathcal{S}$ has at least one extra order of smoothness than the equation would appear to require.

3.2.1 A single constrained equation

We will show that the system of embedding equations (3.3) defined in the band $B(\mathcal{S})$ has the same set of solutions as a single equation on $B(\mathcal{S})$.

Problem 3.2.4. *Given $\gamma \in \mathbb{R}$, let $v : B(\mathcal{S}) \times (0,T) \to \mathbb{R}$ satisfy*

$$v_t = EA(t,x,v) - \gamma\left(v - Ev\right), \qquad x \in B(\mathcal{S}),\ t \in (0,T) \tag{3.5}$$

with initial condition $v(x,0) = v_0(x)$.

Theorem 3.2.5. *Suppose that $v : B(\mathcal{S}) \times (0,T) \to \mathbb{R}$ is a smooth solution of the system of equations (Problem 3.2.2). Then v also satisfies the penalized equation (Problem 3.2.4) for all $\gamma \in \mathbb{R}$. Conversely, if v is a smooth solution of Problem 3.2.4 with initial condition $v_0 = Ev_0$, then v will satisfy the system of embedding equations (Problem 3.2.2).*

Proof. The first part follows directly, since $v = Ev$ (3.3b) implies that the extra term multiplied by γ is zero, and the single equation (Problem 3.2.4) becomes equivalent to (3.3a).

For the converse, we operate on both sides of (3.5) with an extension operator E, and use the fact that this operator is idempotent (Proposition 2.4.2):

$$Ev_t = EA(t,x,v) - \gamma\left(Ev - Ev\right) = EA(t,x,v).$$

Subtracting this from (3.5) gives

$$(v - Ev)_t = -\gamma(v - Ev).$$

We now define a function $z = v - Ev$, to obtain an (infinite-dimensional) ODE for z:

$$z_t = -\gamma z,$$

with initial condition $z_0 = v_0 - Ev_0 = 0$. This has unique solution $z \equiv 0$.

It follows that $v = Ev$, and so the second equation of the system (3.3b) holds. Again, the extra term in (3.5) is zero, and so the single equation is equivalent to the system of equations. □

3.2.2 The Poisson problem

A similar approach can be used for time-independent problems. Here, we show the Poisson equation as an example, but this can be generalized to equations of the form $A_S(x,t,u) = f$ for the same class of operators as above.

Theorem 3.2.6. *Consider the system of embedding equations obtained as above from the Poisson equation $\Delta_{\mathcal{S}} u = f$ on a surface $\mathcal{S}$,*

$$E\Delta v = E_S f \tag{3.6a}$$

$$v = Ev, \qquad x \in B(\mathcal{S}). \tag{3.6b}$$

The solutions of this system are the same as the solutions of the single equation

$$E\Delta v - \gamma(v - Ev) = E_S f, \qquad x \in B(\mathcal{S}), \tag{3.7}$$

for any $\gamma \in \mathbb{R}\backslash\{0\}$.

Proof. If v is a solution of (3.6a) and (3.6b), then the additional term in (3.7) is zero, so the single equation is satisfied. Conversely, if v satisfies (3.7), then we may extend the equation to obtain

$$E^2\Delta v - \gamma(Ev - E^2 v) = EE_S f,$$

and using the idempotence of E, this is

$$E\Delta v - \gamma(Ev - Ev) = E_S f.$$

It follows that

$$E\Delta v = E_S f,$$

and substituting back in (3.7), for any non-zero γ, we have that $v = Ev$. □

This approach could easily be implemented for a more general differential operator as defined in (2.42). Poisson problems are investigated in another work [CM14].

3.3 Penalty term

We study the effect of the additional term in the equation (3.5). The term $-\gamma(v - Ev)$ imposes the side condition $v = Ev$ and can be viewed as a penalty term in the equation. If v is not constant in the direction normal to the surface, then the term $v - Ev$ can be large, and dominate the term containing the differential operator.

We show that the penalized embedding problem (Problem 3.2.4) is Lyapunov stable [HNW93] if γ is positive.

Theorem 3.3.1 (Stability of the penalized problem)**.** *Let $\gamma > 0$. For problem* (3.5)*, any perturbation in the normal direction (for example, in the initial conditions) will decay to a closest point extension like $z_t = -\gamma z$.*

Proof. As in the proof of Theorem 3.2.5, we start from the embedding equation

$$v_t = EA(t, x, v) - \gamma(v - Ev) \tag{3.8}$$

and operate on both sides of the equation with an extension E. Using the idempotence of the operator gives $Ev_t = EA(t, x, v)$. We subtract this from (3.8), and define the function $z = v - Ev$. We can then study this as an ODE system

$$z_t = -\gamma z, \quad \text{with } z_0 = v_0 - Ev_0, \tag{3.9}$$

which has unique solution $z(x, t) = z_0(x)e^{-\gamma t}$. If the initial condition is perturbed slightly, so that it is no longer zero (i.e., v_0 is not exactly a closest point extension), then the function z will still decay to zero, provided that γ is positive. □

3.3.1 Effect of the penalty parameter

It should be emphasized that the parameter γ is not a Lagrange multiplier; it is not necessary to solve for a value of γ as part of the solution procedure. Rather, γ is a numerical parameter that controls how strongly the constraint is imposed.

Any deviation in the normal direction will be penalized by a large right-hand side. Positive values of γ will return the system to the stable equilibrium, while negative values of γ may lead to instability.

The parameter γ may affect the consistency and stability of the method, as investigated further in Chapter 4. We show numerical tests which suggest that a wide range of values result in convergent schemes.

3.4 A method of lines

After converting the system of two equations — the extended equation (3.3a) and the constraint (3.3b) — into a single equation (3.5), we can now formulate a method of lines scheme, using standard finite difference or finite element methods in space.

For example, the computational band $B(\mathcal{S})$ can be discretized as in [RM08] with a uniform Cartesian grid in $\mathbb{R}^n$, and finite difference schemes on this grid can then be used to approximate the spatial differentiation operators.

As an illustration, let the matrix L be the standard 5-point discrete Laplacian in 2D, and let the vector $V \in \mathbb{R}^J$ denote the set of values of the function v at the grid points, with J points in the embedding band.

Multiplication by the $J \times J$ matrix E [MR09] implements the discrete extension of a surface function, approximating the extension E using interpolation on the grid

points surrounding the closest point. We note that this matrix operator is no longer idempotent.

The necessary size of the band $B(\mathcal{S})$ to contain these differentiation and interpolation stencils is discussed in Chapter 4 and is a small multiple of the mesh size Δx (also see [RM08, MR09]).

With the discrete operators inserted into the equation, we obtain a system of ordinary differential equations for the vector V,

$$\partial_t V = \mathsf{E}\mathsf{L}V - \gamma(\mathsf{I} - \mathsf{E})V, \tag{3.10}$$

or more generally,

$$\partial_t V = \mathsf{E}\mathsf{A}V - \gamma(\mathsf{I} - \mathsf{E})V, \tag{3.11}$$

where A is the discretization of the operator A in (2.42). In the linear case, A is a matrix, however in general this can be a nonlinear operator $\mathsf{A}(V)$.

This system of ODEs can then be solved using either implicit or explicit time-stepping methods (or a combination). Note, however, that since the embedding equation is no longer a PDE, the standard notions of consistency, convergence and stability of numerical schemes are not defined. These properties of the method will also depend on the interpolation, spatial discretization and time-stepping schemes used. We discuss some of these issues, in particular how these relate to the choice of the parameter γ, in Chapter 4.

3.4.1 Comparison to other semi-discrete formulations

In the formulation of [RM08], time steps of the discretized PDE are alternated with an extension step:

1. complete one explicit time step of $\partial_t V = \mathsf{L}V$;
2. perform a re-extension $V = \mathsf{E}V$.

This approach is not a method of lines; it forces the solution to be constant in the direction normal to the surface after each time step.

In contrast, in our method of lines approach, this requirement is imposed with a penalty term in the equation itself, so that no explicit re-extension step is required. The semi-discrete form of the penalty equation can then be solved with an implicit time-stepping scheme, for example in the case of stiff problems.

In order to derive an implicit formulation based on the method of [RM08], the conversion of the alternating PDE and re-extension steps into a single semi-discrete equation is studied in [MR09]. The scheme $\partial_t V = \mathsf{LE}V$ is observed to be unstable in practice, so a stabilized version $\partial_t V = \mathsf{M}V$ is proposed for the particular case of the diffusion equation. The matrix M is given by

$$\mathsf{M} = \operatorname{diag}\mathsf{L} + (\mathsf{L} - \operatorname{diag}\mathsf{L})\mathsf{E}.$$

with the diagonal matrix $\operatorname{diag}\mathsf{L}$ containing the diagonal entries of the discrete Laplacian L. Using a standard second-order centred difference stencil on a uniform Cartesian grid, M can be written as

$$\mathsf{M} = \mathsf{LE} - \frac{2n}{(\Delta x)^2}(\mathsf{I} - \mathsf{E}), \tag{3.12}$$

where n is the dimension of the embedding space, and Δx is the discrete grid spacing. At least for the diffusion equation, this is very similar to our (3.10) which also solves $\partial_t V = \mathsf{M}V$ but with

$$\mathsf{M} = \mathsf{EL} - \gamma(\mathsf{I} - \mathsf{E}).$$

Note that the order of the matrices E and L is reversed, and the factor $\frac{2n}{(\Delta x)^2}$ is generalized with the introduction of a new parameter γ. In [MR09], the discrete operator M is derived based on a special treatment of the diagonal of the discretized Laplace operator. The new formulation (3.11) is derived from a study of extended solutions of the surface PDE, and applies to more general surface equations. In Chapter 4 we study the application to semi-linear and higher-order operators.

3.4.2 Summary of the method of lines approach

The resulting algorithm can be summarized as:

1. Extend the surface PDE into the band by applying the extension operator E, and then use Principles 2.5.1 and 2.5.2 to replace surface differential operators with their Cartesian analogues.

2. Add the penalty term $-\gamma(v - Ev)$ to the embedding equation.

3. Use standard discretizations in space for the differential and extension operators, and an appropriate time-stepping scheme to solve the resulting system.

Note that the resulting embedding equation is not a PDE, since the term $A(t, x, v)$ involving a differential operator is evaluated at the surface (due to the extension operator). Nevertheless, we treat the equation using standard PDE discretization schemes, after approximating the extension operators using interpolation.

Algorithm 3: Closest point penalty method [vGMM14]

Data: U^0, initial condition on surface
extension matrix E, differentiation matrix L, restriction matrix R
Result: U, discrete solution on surface at time T_f
begin
  /* Initial condition in band */
  $V \leftarrow \mathsf{E}U^0$
  /* Create matrix */
  $\mathsf{M} \leftarrow \mathsf{EL} - \gamma(\mathsf{I} - \mathsf{E})$
  for $t = 1$ **to** $T_f/\Delta t$ **do**
    /* Solve one explicit or implicit time step in the band */
    /* e.g. using explicit Euler */
    $V \leftarrow V + \Delta t \mathsf{M} V$
  end
  /* Restrict solution to discrete points on the surface */
  $U \leftarrow \mathsf{R}V$
end

Chapter 4

Discretization and the Penalty Term

The discretization of the embedding equation (using a semi-discrete formulation, or method of lines) is constructed in this chapter. We investigate the effect of discretization on the extension operator E, and examine the properties of the penalty term and parameter γ introduced in Chapter 3. Notions of consistency of the scheme with the surface and band equations are defined, and a study of consistency of the method is given for the surface diffusion equation. Numerical tests show stability and convergence over a range of values of the parameter γ. Finally, the method is applied to more general higher-order and nonlinear operators, using the examples of the biharmonic equation and semi-linear systems of reaction-diffusion equations.

4.1 Discretization

The simplicity of the closest point method for surface differential operators comes from the ability to use standard discretization schemes in the surrounding embedding space. Here we study for example a simple finite difference scheme on a standard Cartesian grid.

The penalty formulation of the previous chapter is now discretized on the Cartesian grid, using a method of lines approach. The differential and extension operators are first discretized in space to give a semi-discrete scheme, and an explicit or implicit time-stepping scheme is implemented.

An implicit version of the closest point method was described in [MR09], in which the use of matrices to perform the discrete differentiation and extension operations was introduced. The method described in this chapter also makes use of these ma-

trices, however, motivated by the penalty formulation, the order of operations is interchanged (we discuss this in Section 3.4.1).

4.1.1 Finite difference scheme

The n-dimensional embedding band $B(\mathcal{S}) \subseteq \mathbb{R}^n$ is covered by a fixed mesh in Cartesian coordinates, of grid spacing $\Delta x_1, \ldots, \Delta x_n$. We assume for simplicity that the grid spacing in each dimension is equal, i.e., $\Delta x := \Delta x_1 = \ldots = \Delta x_n$ (see Figure 4.1). The grid is not necessarily aligned with either the surface or the boundary of the band. We will also discretize in time; for this analysis, a uniform time step Δt is chosen, although adaptive time-stepping could be implemented in a standard way.

The Cartesian grid points are labelled x_j, with $j \in J_{B(\mathcal{S})}$, where $J_{B(\mathcal{S})}$ denotes the indices of grid points x_j inside $B(\mathcal{S})$. Let V_j^m denote the value of the discrete solution resulting from the finite difference scheme at grid point x_j and time $t_m = m\Delta t$. We write $V^m := \left(V_j^m,\, j \in J_{B(\mathcal{S})}\right)$ as the solution at the mth time step. With $|J_{B(\mathcal{S})}| = J$ points lying inside the computational band, the vector $V^m \in \mathbb{R}^J$ has length J. The exact solution is denoted $v(x,t)$, and we write $v^m(x) := v(x,t_m)$ for the continuous solution at time step m.

A sampling operator R^h is also defined, where $R^h v^m \in \mathbb{R}^J$ is a vector with jth entry $(R^h v^m)_j := v(x_j,t_m)$, the continuous solution evaluated at the grid point x_j and time t_m. The operator R^h is not to be confused with the continuous operator R in Section 2.3.2 which restricts a function in $B(\mathcal{S})$ to the surface $\mathcal{S}$.

We use standard finite difference schemes for the differential operators. For example, the Laplacian in $\mathbb{R}^2$ can be discretized with a standard 5-point second-order finite difference stencil (now with two-dimensional indices for the grid points),

$$(\mathsf{L}V)_{i,j} = \frac{1}{\Delta x^2}(V_{i,j+1} + V_{i,j-1} + V_{i+1,j} + V_{i-1,j} - 4V_{i,j}). \tag{4.1}$$

We use q to denote the order of the truncation error of the discrete differential operator, with $q = 2$ in this case.

4.1.2 Discrete norms

The error of the scheme can be measured either on the surface or in the embedding band. We use both the discrete maximum norm and the discrete L_2 norm (see also Section 1.4 for the continuous case).

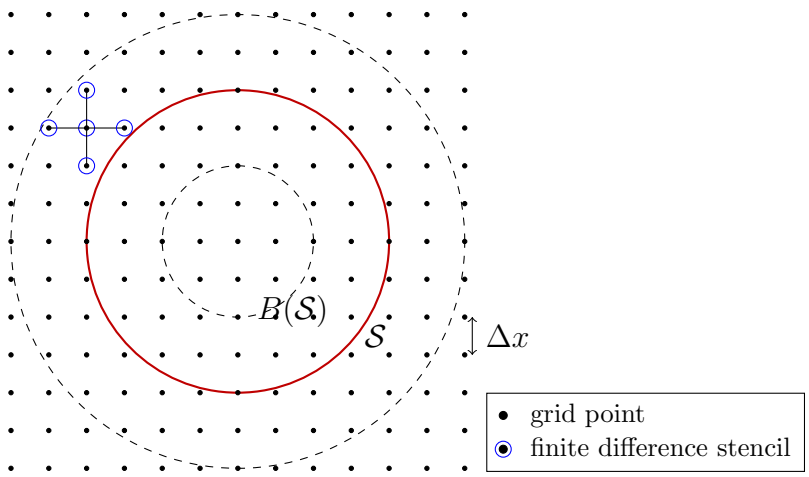


Figure 4.1: Cartesian grid in the embedding space, for a circle in $\mathbb{R}^2$.

In the embedding band, the discrete maximum and L_2 norms are defined as

$$\|V^m\|_{\infty,B(\mathcal{S})} := \max\{|V_j^m|,\, j \in J_{B(\mathcal{S})}\}, \tag{4.2}$$

$$\|V^m\|_{2,B(\mathcal{S})} := \left((\Delta x)^n \sum_{j \in J_{B(\mathcal{S})}} |V_j^m|^2 \right)^{\frac{1}{2}}. \tag{4.3}$$

We also measure the norms of a function projected back onto a discrete sampling of the surface $\mathcal{S}_h$, which is a set of points $\{y_i \in \mathcal{S}\}$ with indices $i \in J_{\mathcal{S}_h}$. The points y_i do not necessarily lie on the Cartesian grid. We assume that the sampling of the surface is uniform, and finer than the Cartesian grid discretization.

Let U_i^m denote the value obtained from interpolation of V^m through the grid points x_j, followed by evaluation at a point $y_i \in \mathcal{S}_h$. We write U^m to denote the vector $(U_i^m)_{i \in J_{\mathcal{S}_h}}$. In the case of triangulated surfaces, the y_i may lie on the faces or vertices of the triangulation.

The maximum and L_2 norms on the approximated surface are then calculated as

$$\|U^m\|_{\infty,\mathcal{S}_h} := \max\{|U_i^m|,\, i \in J_{\mathcal{S}_h}\}, \tag{4.4}$$

$$\|U^m\|_{2,\mathcal{S}_h} := \left(h^k \sum_{i \in J_{\mathcal{S}_h}} |U_i^m|^2 \right)^{\frac{1}{2}}, \tag{4.5}$$

where k is the dimension of the surface.

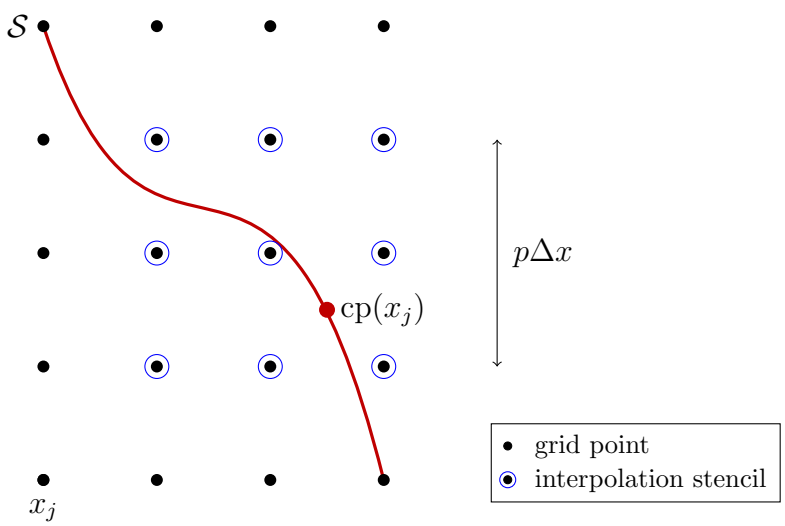


Figure 4.2: Interpolation stencil of width $p\Delta x$ about the point $\mathrm{cp}(x_j)$.

4.2 Discrete extension operators and interpolation

The continuous extension operator E is based on a closest point function cp that is defined everywhere throughout the band. Applying the extension to a function $v : B(\mathcal{S}) \to \mathbb{R}$ requires evaluating v at the point $\mathrm{cp}(x) \in \mathcal{S}$. However, in the discrete setting, we only have samples of the function v at grid points x_j. Each x_j on the grid has a closest point $\mathrm{cp}(x_j)$, but $\mathrm{cp}(x_j)$ will not in general lie at a grid point. We therefore use interpolation to find an approximation to the function value $v(\mathrm{cp}(x_j))$, based on a subset of the values $v(x_i)$ surrounding the surface (see Figure 4.2).

As in [RM08], polynomial interpolation is used to approximate the extension operator. In each dimension, local one-dimensional polynomial interpolants of degree p are calculated through neighbouring points. The interpolation is continued in a dimension-by-dimension fashion (as in Figure 4.3), and the result is evaluated at the point $\mathrm{cp}(x_j)$. The value of the function v at $v(\mathrm{cp}(x_j))$ is expressed as a weighted sum of the neighbouring values $v(x_i)$, where the weights w_i are calculated using Barycentric Lagrange interpolation [BT04],

$$v(\mathrm{cp}(x_j)) \approx \sum_{x_i \in \mathcal{N}(\mathrm{cp}(x_j))} w_i v(x_i). \tag{4.6}$$

Since the interpolation step involves a linear combination of grid point values, this extension can be expressed as a matrix E_p, introduced in the implicit formulation of the closest point method [MR09],

$$[\mathsf{E}_p V]_j = \sum_{x_i \in \mathcal{N}(\mathrm{cp}(x_j))} w_i V_i. \tag{4.7}$$

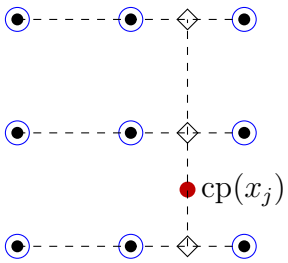


Figure 4.3: Higher-order interpolation is carried out in a dimension-by-dimension fashion, with local one-dimensional interpolants first evaluated at the points $\diamond$. A further interpolant is computed through these points, and evaluated at $\mathrm{cp}(x_j)$.

Recall from Section 4.1.2 that we compute the discrete norm on the surface through a vector U^m, which has values at a set of sampled surface points $\{y_i \in \mathcal{S}\}$, with $i \in J_{\mathcal{S}_h}$. The calculation of U^m involves a discrete extension operator E_{plot}, which interpolates between values in the band and evaluates at the surface points y_i, so that $U^m = \mathsf{E}_{\text{plot}} V^m$. The matrix E_{plot} is computed in a similar manner to the extension matrix E.

4.3 Bandwidth

The computations are carried out on a Cartesian grid in a band surrounding the surface. For our theory to apply, the region of computation should be contained within $B(\mathcal{S})$, the neighbourhood of the surface in which the solution v is defined. We can increase computational efficiency by computing only on a narrow band scaling with the grid size Δx, as the extension operators guarantee that only values of the function near to the surface affect the solution. In [RM08], a bound for the necessary computational bandwidth was calculated for the two-step formulation with the second-order centred difference Laplacian and polynomial interpolation of order p, in n dimensions.

4.3.1 Bounds on the bandwidth

We assume that the differential operator has a stencil which is a symmetric hypercross, of width $c\Delta x$ (see Figure 4.4). For example, the second-order centred difference Laplacian in 2D (5-point stencil) has width $c = 2$, while a fourth-order Laplacian with a 13-point stencil has width $c = 4$. Using polynomial interpolation of degree p in each dimension, the interpolation stencil is an n-dimensional hypercube, with width $p\Delta x$ in each dimension. The computational region must contain all points in the differential

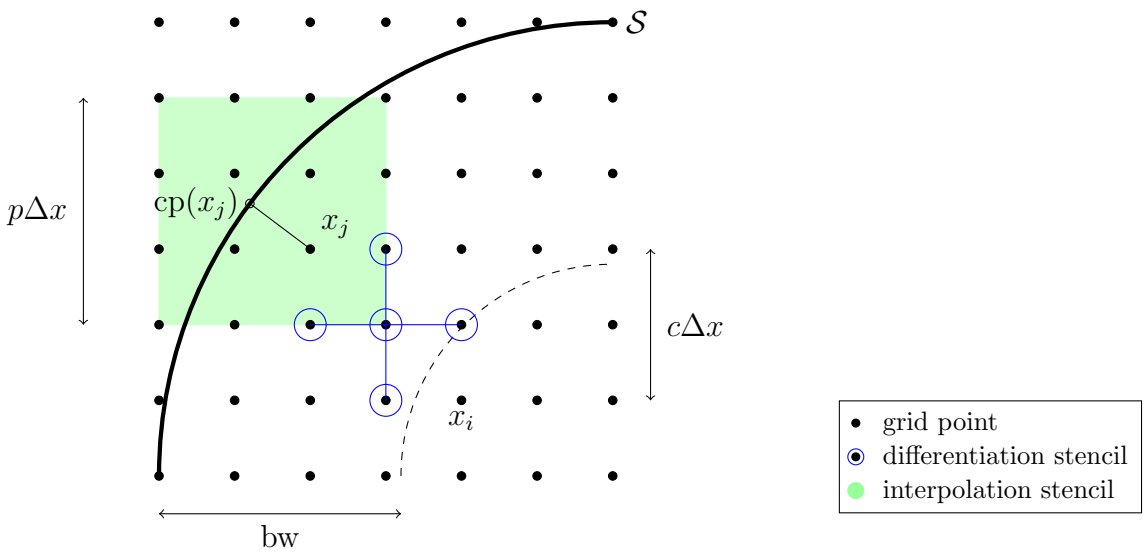


Figure 4.4: Bandwidth bw required for a scheme with interpolation order p and a differential operator stencil of width c in each dimension. Here, $p = 3$ and $c = 2$.

operator hypercross centred at each point in the interpolation hypercube. The width of the band surrounding the surface containing all required points is therefore

$$\mathrm{bw} := \sqrt{(n-1)\left(\frac{p+1}{2}\right)^2 + \left(\frac{c}{2} + \frac{p+1}{2}\right)^2}\,\Delta x. \tag{4.8}$$

In terminology common in image processing (e.g. [CS05]), the set of points in the band is the *dilation* of the width-p interpolation stencil, under the width-c structuring element $S_c = \mathsf{L}$.

Note that the matrix EL depends only on points within the dilation of the interpolation stencil, and hence the value at any grid point x_i outside of this does not affect the computation (see Figure 4.4). If we construct the matrix EL for a larger region containing points x_i outside of this region, the corresponding columns of the matrix will be zero.

Note that we require Δx to be small enough so that the computational band above is contained completely within $B(\mathcal{S})$, the neighbourhood of the surface in which the map cp is sufficiently smooth. The width of this band $B(\mathcal{S})$ depends on the curvature of the surface. For a smooth curve in $\mathbb{R}^2$, the Euclidean closest point function ecp is well-defined in an open band of width $1/\kappa(s)$, where $\kappa(s)$ is the curvature of the surface at a point s. We must therefore choose the grid size Δx, interpolation degree

p, and difference stencil size c such that

$$\text{bw} < \left(\max_{y \in \mathcal{S}} \kappa(y) \right)^{-1}. \tag{4.9}$$

The computational cost of the scheme will depend on the size of the computational band used. The matrices E and L are sparse, so the matrix vector multiplications require $O(J)$ operations, where there are J points in the band. In the embedding space $\mathbb{R}^n$, the $J \times J$ matrix E will have $(p+1)^n$ non-zero entries in each row, while L will have $cn+1$ non-zero entries per row. Since the bandwidth bw scales with the grid spacing Δx, the number of points in the band (and hence the number of operations) scales with Δx^{-k}, where k is the dimension of the surface rather than the embedding space. The computational cost of the embedding method is therefore of the same order as that of a surface intrinsic method.

4.3.2 Boundary

As the neighbourhood or embedding band $B(\mathcal{S})$ has a non-empty boundary $\partial B(\mathcal{S})$, the discretized system may require some boundary conditions. Unlike in the original explicit closest point method, the solution is not re-extended after each time step. The constraint $v = Ev$ is consistent with a Neumann boundary condition,

$$\nabla v \cdot E\nu(x) = 0, \quad x \in \partial B(\mathcal{S}). \tag{4.10}$$

If the band $B_\alpha(\mathcal{S})$ has constant width α, then $\partial B(\mathcal{S})$ is a level set of the signed distance function $\text{sd}(x)$. Then the normal $\nu(y)$ to the surface at $y = \text{cp}(x)$ is orthogonal to the tangent space of the boundary $\mathcal{T}_x \partial B(\mathcal{S})$, so the boundary condition (4.10) is a Neumann condition (see Figure 4.5). If the width of the band $B(\mathcal{S})$ is allowed to vary, the tangent to $\partial B(\mathcal{S})$ at $x \in \partial B(\mathcal{S})$ is no longer necessarily perpendicular to the normal $\nu(y)$, so this is an oblique boundary condition.

In the discrete case, we apply zero Dirichlet boundary conditions in the differential operator matrix L, which appears in the discrete equation only in the combination EL. The application of the extension matrix E then selects only those grid points which are within the interpolation stencil of the surface. Provided that the bandwidth is sufficiently large (as in (4.8)), we have satisfied $V^n = \mathsf{E}V^n$ at the boundary. We thus have a discrete form of the oblique boundary condition above.

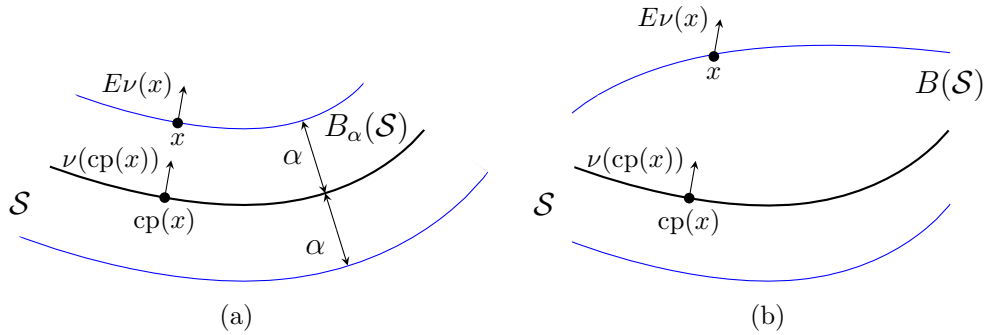


Figure 4.5: For a band $B_\alpha(\mathcal{S})$ of constant width α (a), the boundary condition $\nabla v \cdot E\nu(x) = 0$ is a Neumann condition. If the width of the band is allowed to vary (b), this becomes an oblique boundary condition.

4.4 Consistency

We now consider the consistency of the fully discrete problem, obtained from applying a time-stepping scheme to the semi-discrete form (3.11) of the constrained embedding equation (3.5). We give a definition of consistency for this scheme, and show that we can choose discretizations of the extension operator resulting in a numerical solution which approaches the solution of (3.5) as the discretization is refined. For the analysis of the scheme, γ is assumed to be a constant independent of the grid spacing Δx and the time step Δt, although in practice we choose to vary γ with the grid spacing.

We first write down a general form of the numerical scheme. Recall the constrained embedding equation,

$$v_t = EA(t, x, v) - \gamma\left(v - Ev\right), \qquad x \in B(\mathcal{S}),\ t \in (0, T) \tag{4.11}$$

with initial condition $v(x, 0) = v^0(x)$ satisfying $v^0(x) = Ev^0(x)$, and where A is a linear operator which obeys a closest point principle. Let L be a discretization of this operator, and E_p be the p-th order polynomial interpolation matrix of the continuous operator E. Then using explicit Euler time-stepping, the numerical scheme can be written as

$$\begin{aligned} V^{n+1} &= V^n + \Delta t\left(\mathsf{E}_p\mathsf{L}V^n - \gamma(\mathsf{I} - \mathsf{E}_p)V^n\right), \\ &= \left(I + \Delta t(\mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p))\right)V^n, \end{aligned} \tag{4.12}$$

with initial condition $V^0 = R^h v^0$. A similar form of the update scheme can be found for more general higher-order time-stepping schemes such as Runge–Kutta methods.

We will consider consistency of the numerical scheme (4.12) with the band equation (4.11). In Chapter 3, we showed that the smooth solutions of this band equation

can be mapped to solutions of the surface equation. We will now show that if the numerical scheme is consistent with the band equation (i.e., if the truncation error tends to zero when the scheme is applied to an exact solution v, while Δx, Δt tend to zero along a given refinement path), then the same scheme applied to $E_S u$ is consistent with solutions u of the original surface equation.

4.4.1 Truncation error

Two forms of trucation error are defined, one relating the discrete scheme to the embedding equation, and the other to the original surface equation.

Writing the numerical scheme (4.12) as $V^{n+1} = G^h V^n$, we define the truncation error in the embedding space, e.g. [MM05],

$$\begin{aligned} T^n &:= \frac{1}{\Delta t}\left(R^h v^{n+1} - G^h R^h v^n\right) \\ &= \frac{1}{\Delta t}\left(R^h v^{n+1} - \left(\mathsf{I} + \Delta t(\mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p))\right) R^h v^n\right), \end{aligned} \tag{4.13}$$

recalling that v^n is the exact solution of (4.11) evaluated at time t^n, and defining $G^h := G(\Delta x, \Delta t, p)$ to be the *amplification matrix* of the scheme. Recall that R^h restricts a continuous function to the grid (Section 4.1.1).

Definition 4.4.1 (Consistency with embedding equation)**.** *We say that the scheme $V^{n+1} = G^h V^n$ is consistent with the embedding equation if the truncation error* (4.13) *satisfies $\max_{j \in J_{B(\mathcal{S})}} |T^n| \to 0$, as $\Delta x, \Delta t \to 0$ (possibly along some pre-defined refinement path relating Δx and Δt).*

We now define the truncation error on the surface to be

$$T^n_{\mathcal{S}} := \frac{1}{\Delta t}\left(R^h E_S u^{n+1} - G^h R^h E_S u^n\right), \tag{4.14}$$

as in [MMC14], where u^n is the exact solution of the original surface equation (Problem 3.2.1) at time t^n, and $R^h E_S u^n$ is the restriction to grid points of the continuous extension of this function. Note that the function u^n is not dependent on the penalty parameter γ.

Definition 4.4.2 (Consistency with surface PDE)**.** *We say that the scheme $V^{n+1} = G^h V^n$ is consistent with the surface PDE if the truncation error on the surface* (4.14) *satisfies $\max_{j \in J_{B(\mathcal{S})}} |T^n_{\mathcal{S}}| \to 0$, as $\Delta x, \Delta t \to 0$ (possibly along some pre-defined refinement path relating Δx and Δt).*

Theorem 4.4.3. *If a scheme $V^{n+1} = G^h V^h$ is consistent with the embedding equation (3.5), so that $\max_{j\in J_{B(\mathcal{S})}}|T^n| \to 0$ as $\Delta x, \Delta t \to 0$, then we also have consistency with the original surface problem (Problem 3.2.1), i.e., $\max_{j\in J_{B(\mathcal{S})}}|T^n_{\mathcal{S}}| \to 0$ along the same refinement path.*

Proof. As in the previous chapter, if $u(t,x)$ is a smooth solution of Problem 3.2.1, then $v(t,x) = E_S u(t,x)$ satisfies (3.5). Then $v^n = E_S u^n$, so $T^n = T^n_{\mathcal{S}}$. □

Note that the two notions of consistency differ in their treatment of the penalty parameter γ. Since the solution u of the original surface PDE has no dependence on γ, we are free to choose γ as a varying numerical parameter as $\Delta t, \Delta x$ tend to zero, while still retaining consistency with the surface PDE, although consistency with the embedding equation may no longer be defined in this case.

In the following sections, we consider consistency with the embedding equation, as in Definition 4.4.1, and treat γ as a fixed parameter. Later, we will consider consistency with the surface PDE (Definition 4.4.2) with non-constant γ.

4.4.2 Truncation error of the penalty term

Consider the zero equation on the surface,

$$u_t = 0, \quad y \in \mathcal{S}, t \in (0,T) \tag{4.15}$$

with some initial condition $u(y,0) = u^0(y)$, so that the exact solution is $u = u^0$. The corresponding embedding equation contains only the penalty term and is

$$v_t = -\gamma(v - Ev), \quad x \in B(\mathcal{S}), t \in (0,T), \tag{4.16}$$

with $v^0 = E_S u^0 = Ev^0$. Again, the exact solution is $v = v^0$, and this is a stable solution in time for each fixed x, provided that $\gamma > 0$. The numerical scheme is then

$$V^{n+1} = \mathsf{I} - \gamma\Delta t(\mathsf{I} - \mathsf{E}_p)V^n \tag{4.17}$$

using explicit Euler time-stepping. The truncation error resulting from the penalty term is

$$\begin{aligned} T^n &= \frac{1}{\Delta t}\left(R^h v^{n+1} - (\mathsf{I} - \gamma\Delta t(\mathsf{I} - \mathsf{E}_p))\, R^h v^n\right) && (4.18)\\ &= \gamma(\mathsf{I} - \mathsf{E}_p)V^0. && (4.19) \end{aligned}$$

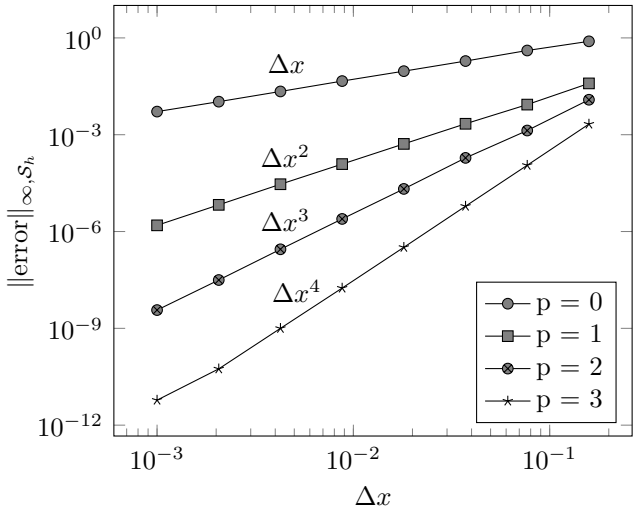


Figure 4.6: Maximum error on the surface resulting from the truncation error of the penalty term, for the zero problem on the unit circle at time $t = 0.5$. The scheme uses first-order explicit time-stepping with $\Delta t = \frac{1}{4}\Delta x$ and $\gamma = 4$.

If we start with an initial condition satisfying $V^0 = \mathsf{E}_p V^0$, then the method has truncation error $T^n = 0$ for this equation. However, this is not equivalent to requiring that $v^0 = Ev^0$, since the interpolation matrix E_p is not idempotent. Instead, we have

$$\mathsf{E}_p V^0 = \mathsf{E}_p R^h v^0 = R^h E v^0 + O(\Delta x^{p+1}) = V^0 + O(\Delta x^{p+1}), \tag{4.20}$$

from the properties of polynomial interpolation, so that

$$T^n = \gamma O(\Delta x^{p+1}). \tag{4.21}$$

If γ is a constant independent of $\Delta x, \Delta t$, then in principle we may use polynomial interpolation of any degree $p \geq 0$ for a consistent scheme.

Note that the truncation error depends only on the spatial step size, and not on the time step, although the original surface problem involved only a time derivative.

Numerical results

Figure 4.6 shows the surface error at $t = 0.5$ for explicit Euler time-stepping applied to the zero problem $u_t = 0$ on the unit circle. Initial conditions and exact solution are both $u(\theta) = \cos\theta + \cos(3\theta)$, and the time step is chosen to be $\Delta t = \frac{1}{4}\Delta x$, with parameter $\gamma = 4$. We see the expected convergence rate of $O(\Delta x^{p+1})$ for polynomial interpolation of order p.

4.4.3 Diffusion equation

We now consider the consistency of the method applied to the case of the Laplace–Beltrami operator. The penalty equation in $B(\mathcal{S})$ is

$$v_t = E\Delta v - \gamma(v - Ev), \qquad x \in B(\mathcal{S}),\ t \in (0,T), \tag{4.22}$$

with initial condition $v(x,0) = v^0(x)$ satisfying $v^0(x) = Ev^0(x)$.

With first-order explicit time-stepping, the numerical scheme can be written

$$V^{n+1} = V^n + \Delta t\left(\mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p)\right)V^n. \tag{4.23}$$

Evaluating the penalty term as in the zero case gives

$$\mathsf{E}_p R^h v^n = R^h E v^n + O(\Delta x^{p+1}), \tag{4.24}$$

so that the penalty term results in

$$(\mathsf{I} - \mathsf{E}_p)R^h v^n = R^h(v^n - Ev^n) + O(\Delta x^{p+1}). \tag{4.25}$$

With a second-order discretization of the Laplacian operator, provided that v^n is sufficiently smooth we have

$$\mathsf{L}R^h v^n = R^h \Delta v^n + O(\Delta x^2), \tag{4.26}$$

and including the discrete extension operator,

$$\mathsf{E}_p\mathsf{L}R^h v^n = R^h E\Delta v^n + O(\Delta x^2) + O(\Delta x^{p+1}), \tag{4.27}$$

provided that Δv^n is sufficiently smooth for the error analysis of the interpolation scheme. From the first-order time-stepping,

$$\frac{1}{\Delta t}(R^h v^{n+1} - R^h v^n) = R^h v_t + O(\Delta t). \tag{4.28}$$

Thus, combining terms and using the continuous equation (4.22),

$$\begin{aligned} T^n &= \frac{1}{\Delta t}\left(R^h v^{n+1} - R^h v^n + \Delta t\left(\mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p)\right)R^h v^n\right) \\ &= R^h\left(v_t - E\Delta v^n + \gamma(v^n - Ev^n)\right) + \\ &\quad + O(\Delta t) + O(\Delta x^2) + O(\Delta x^{p+1}) + \gamma O(\Delta x^{p+1}) \\ &= O(\Delta t) + O(\Delta x^2) + O(\Delta x^{p+1}) + \gamma O(\Delta x^{p+1}). \end{aligned} \tag{4.29}$$

We therefore have consistency of the scheme: if the penalty parameter γ is independent of Δx and Δt, then the scheme (4.23) with first-order explicit time-stepping is consistent with the continuous equation (4.22) if $p \geq 0$. If we choose $\Delta t = O(\Delta x^2)$ and $p \geq 1$, then the scheme will be second-order. The analysis of higher-order explicit time-stepping methods (such as Runge–Kutta methods) is similar.

4.4.4 Penalty parameter

We now consider a discretization scheme of a more general PDE. Requiring the method to be consistent and stable for a given value of the penalty parameter γ may place certain restrictions on the spatial grid size Δx, the time step Δt, and the interpolation order p of the extension operator. Since we are free to choose any positive value of γ in the continuous case, we would like to choose γ in order to minimize the restrictions on the numerical scheme.

Although we cannot define consistency with the continuous penalty equation (4.22) if the value of γ varies, we can use the notion of consistency with the original surface PDE (which contains no parameter γ).

Assume that the standard PDE with no extension operators admits a consistent discretization, where the truncation error scales like $O(\Delta t^\alpha) + O(\Delta x^q)$. The surface truncation error of the corresponding embedding discretization will then include a term proportional to Δx^{p+1} from the discrete extension operator.

Recall that the truncation error of the scheme also includes a term of order $\gamma O(\Delta x^{p+1})$ from the penalty term. If γ is chosen to scale with $\Delta x^{-\beta}$ for some constant $\beta \in \mathbb{R}$, then this term is a contribution of $O(\Delta x^{p+1-\beta})$. Combining these results gives an overall order of accuracy of the method of

$$T_{\mathcal{S}}^n = O(\Delta t^\alpha) + O(\Delta x^q) + O(\Delta x^{p+1}) + O(\Delta x^{p+1-\beta}).$$

For a scheme which is consistent with the surface PDE, it is necessary that $p \geq \beta$. If $\gamma = O(\Delta x^{-2})$ (which may be required for stability, as discussed in Section 4.6.2), then at least degree 2 interpolation in p is needed.

Note that if for some equations, a stable scheme using a constant value of γ can be constructed, then using $p = 1$ should also give second-order convergence. This would be computationally more efficient, since then only bilinear/trilinear interpolation matrices could be used. The consistency of the closest point method applied to various other problems, including advection and fourth-order diffusion, is investigated in [MMC14].

4.5 Numerical experiments

We demonstrate the convergence of the method with examples in 2D and 3D, showing the convergence rates expected based on the discussion in Section 4.4.4.

4.5.1 Diffusion equation on the unit circle and unit sphere

The diffusion equation example of Section 3.1.1 is studied on the unit circle embedded in $\mathbb{R}^2$, and the unit sphere embedded in $\mathbb{R}^3$. Starting from the surface equation $u_t = \Delta_{\mathcal{S}} u$, the resulting embedding equation with the penalty term is

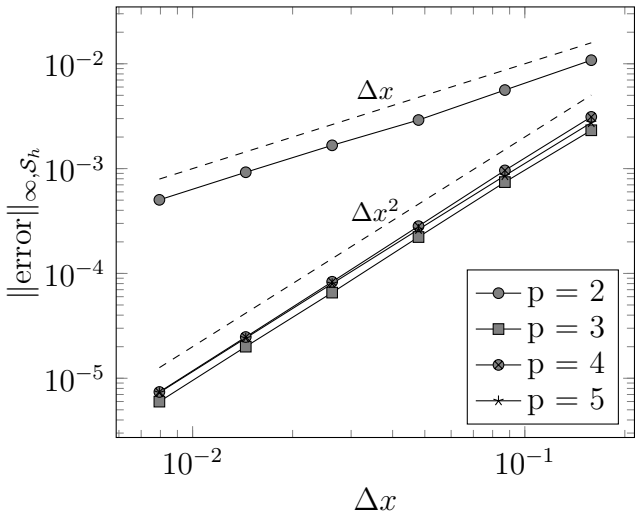
$$v_t = E\Delta v - \gamma(v - Ev). \tag{4.30}$$

We take the standard parameterization $\sigma : [0, 2\pi) \to \mathcal{S}, \sigma(\theta) = (\cos(\theta), \sin(\theta))^T$, for the unit circle, and write $\bar{u}(t, \theta) = u(t, \sigma(\theta))$. The initial condition on the circle is taken to be $\bar{u}(0, \theta) = \cos\theta + \cos 3\theta$, giving exact solution $\bar{u}(t, \theta) = e^{-t}\cos\theta + e^{-9t}\cos 3\theta$. Similarly, the parameterization of the sphere is given by $\sigma : (-\pi, \pi] \times [-\frac{\pi}{2}, \frac{\pi}{2}] \to \mathcal{S}, \sigma(\theta, \phi) = (\cos\theta\cos\phi, \sin\theta\cos\phi, \sin\phi)$. The initial condition is $\bar{u}(0, \theta, \phi) = \cos(\phi + 1/2)$, so that $\bar{u}(t, \theta, \phi) = e^{-2t}\cos(\phi + 1/2)$.

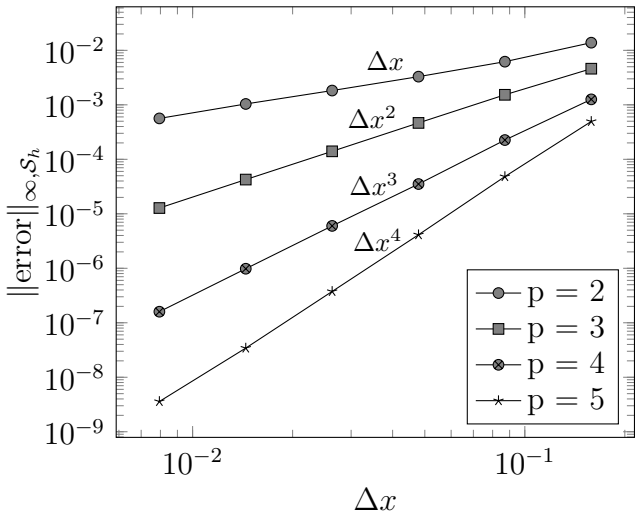
Standard second-order and fourth-order central differences ($q = 2$ and $q = 4$ respectively) are used to discretize the Laplacian, and the order p of the polynomial interpolation is varied. Figure 4.7 shows convergence studies with explicit time-stepping on the unit circle, using a forward Euler and fourth-order Runge–Kutta scheme respectively. The parameter γ is taken to be $\frac{2n}{\Delta x^2}$, where n is the dimension of the embedding space, and the solution is run in time until $t = 0.5$, using $\Delta t = \frac{1}{4}\Delta x^2$. Figure 4.8 shows the case of a unit sphere in $\mathbb{R}^3$, with an implicit BDF2 scheme and again using $\Delta t = \frac{1}{4}\Delta x^2$. The figures demonstrate convergence rates of order $O(\Delta x^{p-1}) + O(\Delta x^q)$. We compute the error by restricting the solution of the embedding equation to a discrete sampling $\mathcal{S}_h$ of the surface $\mathcal{S}$, and computing the max-norm error over $\mathcal{S}_h$. This restriction is approximated via interpolation (using the E_{plot} matrix of Section 4.2) to sufficient accuracy so as not to affect the observed convergence.

4.5.2 Surface and band errors

The examples above were tested by measuring the maximum error on the sampled surface $\mathcal{S}_h$. Figure 4.9 shows the results of the diffusion equation on the unit circle as above, where we now compute the L_2 and max-norm errors on the surface and in the band (see Section 4.1.2). With $p = 3$ and $q = 2$, we achieve second-order convergence in the maximum norms on the surface and in the band, and in the surface L_2 norm. The convergence of the L_2 norm in the band is half an order higher, which we expect since the number of grid points in the band J scales approximately with Δx^{-1}, so integrating over the band gives a factor of $\sqrt{J\Delta x^2} = O(\Delta x^{1/2})$.



(a)



(b)

Figure 4.7: Numerical convergence studies for the diffusion equation (4.30) on the unit circle, using forward Euler (a) and 4th order Runge–Kutta (b) time-stepping, with $\gamma = \frac{2n}{\Delta x^2}$, and $\Delta t = \frac{1}{4}\Delta x^2$. The Laplacian is discretized with order $q = 2$ (a) and $q = 4$ (b).

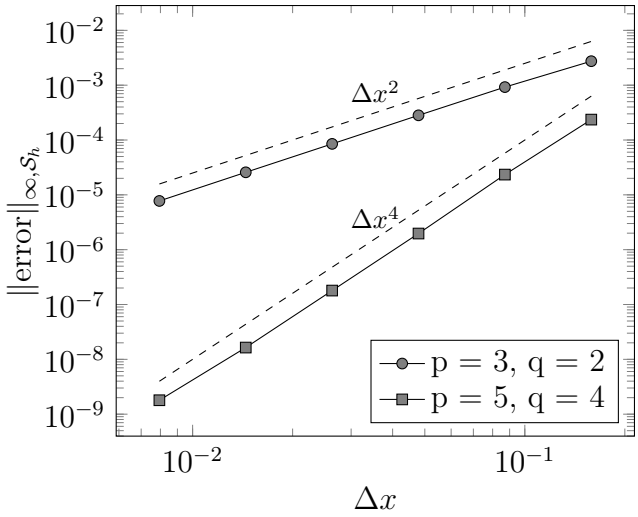


Figure 4.8: Numerical convergence study for the diffusion equation (4.30) on the unit sphere, using implicit BDF2 time-stepping with $\gamma = \frac{2n}{\Delta x^2}$ and $\Delta t = \frac{1}{4}\Delta x^2$.

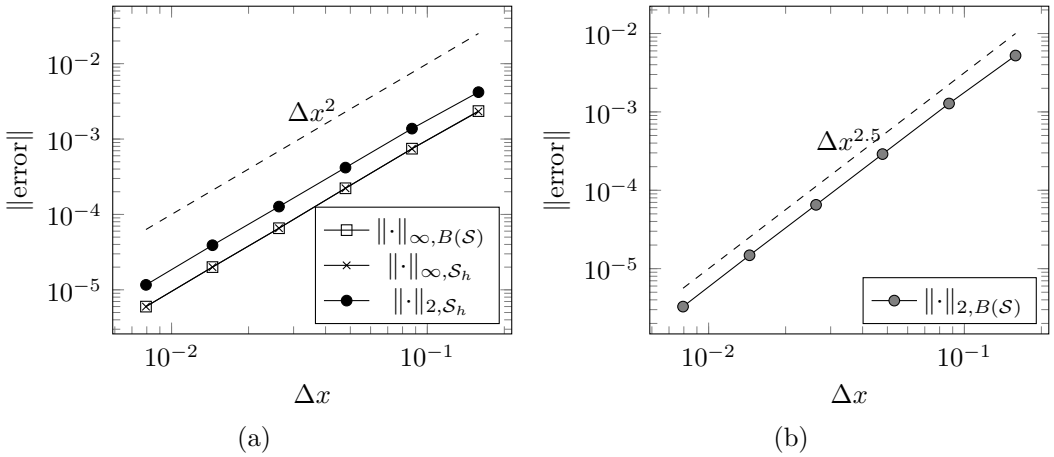


Figure 4.9: Maximum and L_2 error on the surface $\mathcal{S}$ and in the band $B(\mathcal{S})$, for diffusion equation on the unit circle at time $t = 0.5$, using $p = 3$, $q = 2$, and first-order explicit time-stepping with $\Delta t = \frac{1}{4}\Delta x^2$ and $\gamma = \frac{4}{\Delta x^2}$.

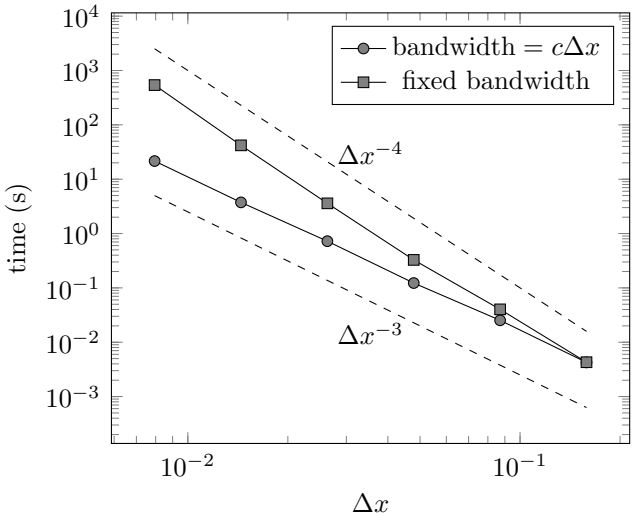


Figure 4.10: Time taken to compute the numerical solution up to $t = 0.5$ for the diffusion equation on the unit circle, using either a narrow band scaling with Δx, or a fixed large band. Explicit Euler time-stepping with $\Delta t = \frac{1}{4}\Delta x^2$ is used.

4.5.3 Bandwidth

We demonstrate that the bandwidth bw given by (4.8) is sufficient for the method, and gives the same results as a computation in a larger band which does not scale with Δx. Figure 4.10 shows the time taken to compute the solution to time $t = 0.5$ using explicit Euler time-stepping for the diffusion equation on the unit circle, using either the bandwidth bw calculated above (4.8) which scales with Δx, or a fixed bandwidth. The solutions U^m computed in the two cases are identical when evaluated on the surface.

We have used $\Delta t = \frac{1}{4}\Delta x^2$, so computation to a fixed time will take $O(\Delta x^{-2})$ time steps. The number of points J in the band is approximately $O(\Delta x^{-1})$ if the bandwidth scales with Δx, and $O(\Delta x^{-2})$ in the fixed bandwidth case. Multiplication by the matrix E or L is an $O(J)$ operation, so we expect the computation time to be $O(\Delta x^{-3})$ and $O(\Delta x^{-4})$ respectively, as observed in practice.

4.6 Stability

The stability of the numerical scheme depends on the properties of the matrices E and L, as well as on standard time step restrictions and the parameter γ. We first consider the semi-discrete system: discrete in time while still continuous in space, before turning to the fully discrete system.

4.6.1 ODE stability

The absolute stability of the semi-discrete system (e.g. [LeV07]) is used to find a relationship between the value of γ and the time step Δt. The test equation $u_t = 0$ on the surface will result in the embedding equation (4.16). Recall that this can be written in the form of an infinite-dimensional ODE system as in Section 3.3,

$$z_t = -\gamma z, \quad \text{where } z = v - Ev. \tag{4.31}$$

We discretize this equation in time, leaving space continuous. As (4.31) is the Dahlquist test equation [HNW93], the region of absolute stability is determined by the time-stepping method used. For example, for the forward Euler method, the region of absolute stability is $|1 - \gamma\Delta t| \leq 1$. With positive γ, this implies a time step restriction $\Delta t \leq \frac{2}{\gamma}$. Likewise, for the explicit four-stage Runge–Kutta scheme we have a stability restriction of $\Delta t \leq \frac{2.79}{\gamma}$. For A-stable methods such as the implicit Euler method, all positive values of γ and Δt give stable solutions, at least for this semi-discrete problem.

Thus returning to the method of lines formulation (3.11), we expect a stability restriction (when using an explicit scheme) based on each of the two terms in the equation and we will take Δt based on the minimum of the two restrictions. One reasonable strategy in choosing γ is to avoid increasing the stiffness of the system (compared to that of the Cartesian discretization of the equivalent non-surface PDE problem). We explore this further in the next section.

4.6.2 Time step restrictions

Consider the surface diffusion equation. If we discretize (4.22) using forward Euler in time and the standard second-order scheme for the Laplacian, as in (4.23) above, we might expect a time step restriction of

$$\Delta t \leq \min\left(\frac{\Delta x^2}{2n}, \frac{2}{\gamma}\right). \tag{4.32}$$

For example, if γ satisfies $\gamma \leq \frac{4n}{\Delta x^2}$, then we expect the usual choice $\Delta t \leq \frac{\Delta x^2}{2n}$ to result in a stable scheme. We are free to choose γ larger than this, provided that the time step is correspondingly restricted.

Figure 4.11 illustrates that the above theory correctly predicts the practical stability properties. On the unit circle, we consider the equation $u_t = \Delta_{\mathcal{S}} u - u$ with semi-discrete form $V_t = \mathsf{E}\mathsf{L}V - V - \gamma(\mathsf{I} - \mathsf{E})V$. We discretize with forward Euler

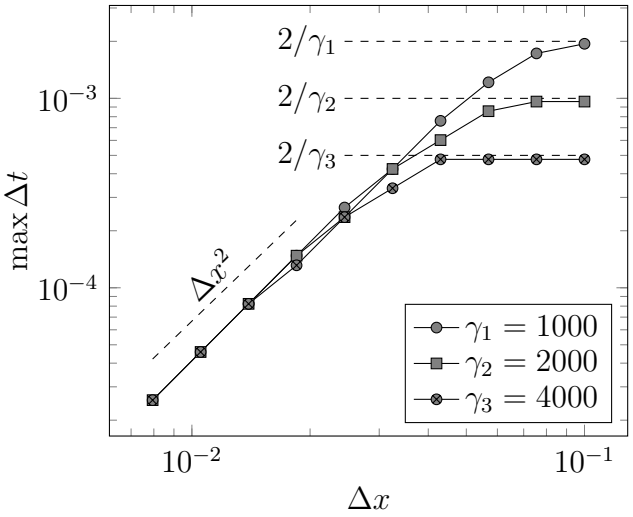


Figure 4.11: Maximum value of Δt giving stable solutions of $u_t = \Delta_{\mathcal{S}} u - u$ on the unit circle, using forward Euler time-stepping.

and estimate the largest possible stable time-step; the results are very close to (4.32). Using a parameter value satisfying $\gamma \leq \frac{4n}{\Delta x^2}$ allows the time step predicted by the standard non-surface Cartesian finite difference scheme. In practice if a larger value of γ is desirable, then the time step Δt could be reduced for stability.

4.6.3 Stability of the fully discrete system

Here we give some observations on the stability of the fully discrete system. Further details of the stability of both the penalty and the re-extension formulations are given in [MMC14], although completely general stability results are still lacking.

The stability of the scheme will also depend on the choice of γ, for the equation and discretization considered. In the case that γ is zero, the side condition is not enforced at each time step. In practice, in this case small errors in the normal direction tend to grow over time, eventually leading to instability. Figure 4.12 shows how the maximum error in the solution depends on γ for the particular case of the diffusion equation on the unit circle at time $t = 0.5$, using forward and backward Euler time-stepping with $\Delta t = \frac{1}{4}\Delta x^2$ and $\Delta t = \frac{1}{4}\Delta x$ respectively. The vertical line in the first figure is at $\gamma \Delta x^2 = 8$, i.e., at $\Delta t = \frac{2}{\gamma}$, where the solution becomes unstable due to the time step restrictions described in Section 4.6. For implicit time-stepping, this instability does not occur. As γ becomes small relative to $\frac{1}{\Delta x^2}$, (or for fixed γ, as Δx becomes small relative to $\frac{1}{\sqrt{\gamma}}$), the solution may also become unstable. Intuitively, this is because

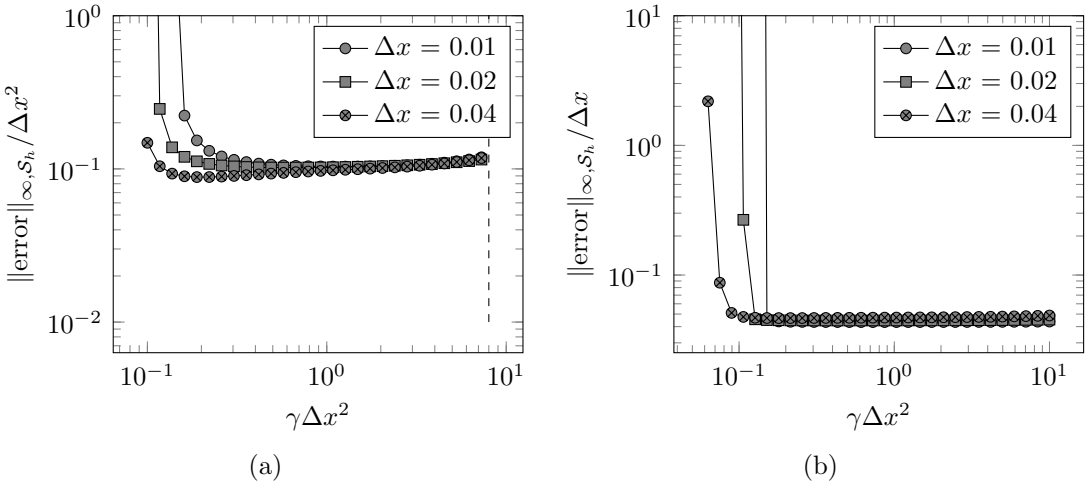


Figure 4.12: Maximum error at time $t = 0.5$ as a function of $\gamma\Delta x^2$, for the heat equation on the unit circle, using (a) explicit and (b) implicit Euler time-stepping.

the penalty is not strong enough to impose the constraint. For the diffusion equation, a suggested value is $\gamma \approx \frac{4}{\Delta x^2}$.

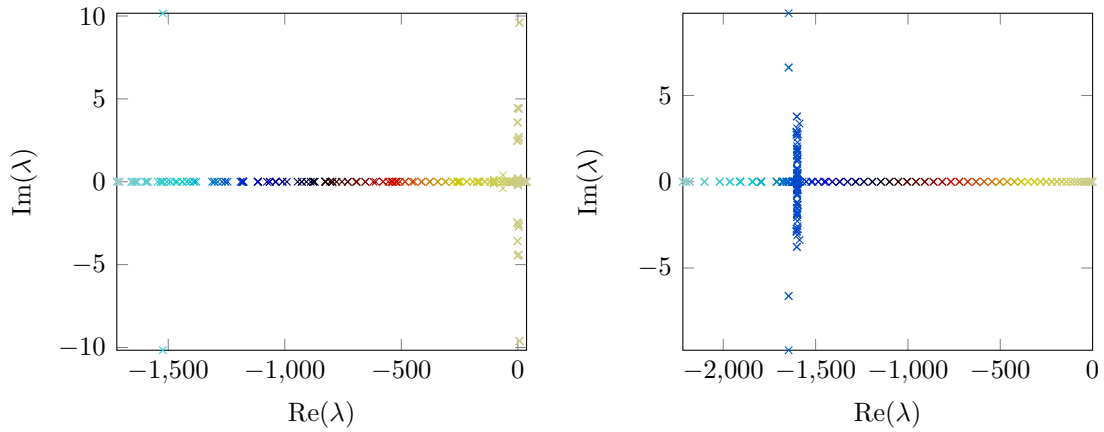
In [CM14], a case is considered where the two extension operators in the scheme (3.11) have different degrees p of interpolation. For the Poisson equation on closed curves in $\mathbb{R}^2$, the scheme with two extension operators with polynomial interpolations of order 1 and 3, and $\gamma = \frac{4}{\Delta x^2}$ is proved to be second-order and stable, i.e., using the matrix $\mathsf{M} = \mathsf{E}_1\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_3)$.

Convergence

In the case of numerical schemes for standard Cartesian PDEs, convergence can be deduced from the consistency and stability of the scheme. However, this analysis may not apply to the embedding equations discussed in this work, in which the differential operator terms appear only with extension operators. We have demonstrated convergence numerically for the extended analogues of some standard PDEs, but a full analysis of this type of equation remains to be carried out.

4.6.4 Properties of extension matrices

We investigate the eigenvalues of the extended Laplace operator $\mathsf{E}_p\mathsf{L}$, as well as the matrix with penalty term $\mathsf{M} = \mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p)$. For stability of the numerical scheme, the eigenvalues of the matrix must lie in the stability region of the time-stepping scheme chosen.



(a) Unit circle, $\mathsf{E}_p\mathsf{L}$ (b) Unit circle, M

Figure 4.13: Eigenvalues of the discrete matrix operator $\mathsf{E}_p\mathsf{L}$ (a) and the penalized operator $\mathsf{M} = \mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p)$ (b) for a unit circle, with $p = 3$, $\Delta x = 0.05$ and $\gamma = 1600$.

Figure 4.13a shows the eigenvalues of the matrix $\mathsf{E}_p\mathsf{L}$, for a unit circle, using interpolation $p = 3$, and grid size $\Delta x = 0.05$. Note that some eigenvalues lie in the right half plane, consistent with our intuition that the solution may grow in an unstable way in the normal direction if the penalty is omitted. In Figure 4.13b, the eigenvalues of the matrix $\mathsf{M} = \mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p)$ are shown. All eigenvalues now lie in the left half plane or at the origin.

Similar results are found for surfaces in $\mathbb{R}^3$. Figure 4.14 shows the eigenvalues of $\mathsf{E}_p\mathsf{L}$ and M for the unit sphere and the genus 3 surface of Section 4.7.1.

4.6.5 Relationship to the explicit method of Ruuth & Merriman

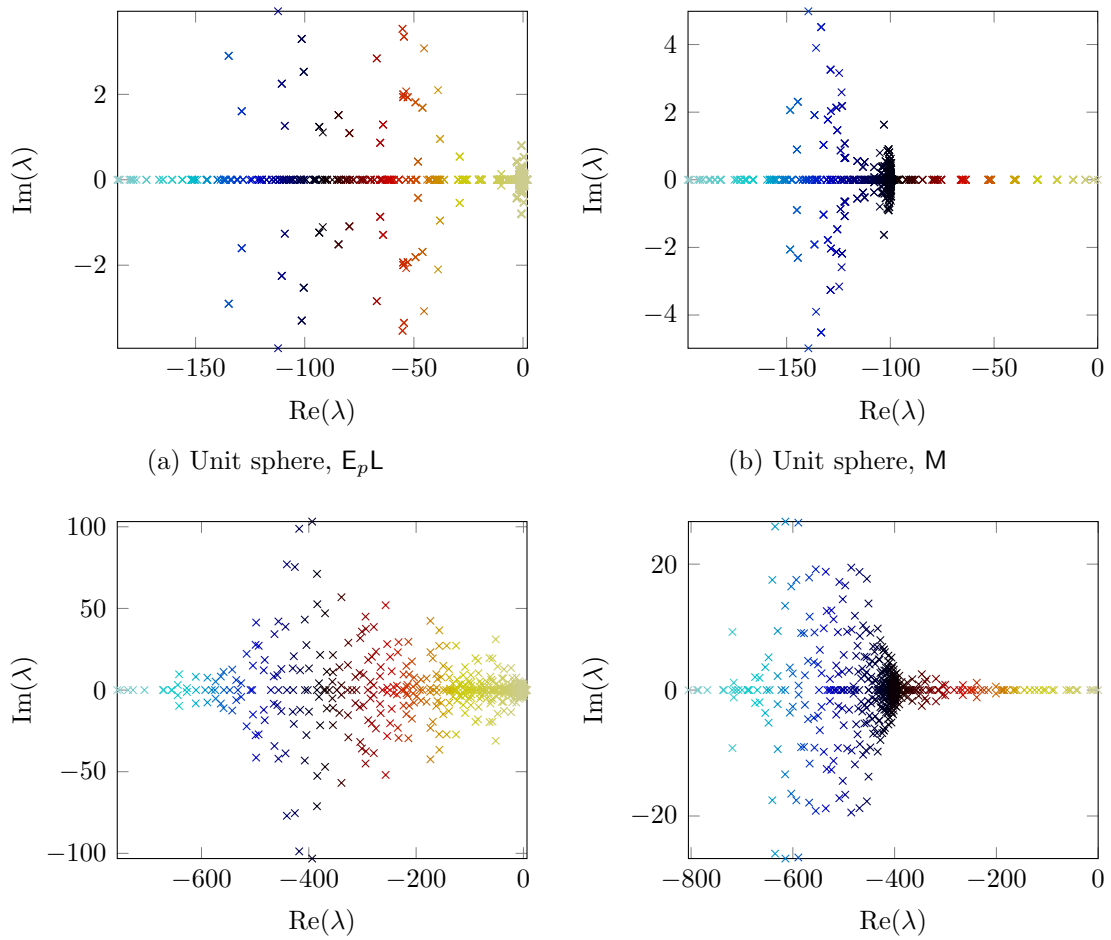
Note that if we choose $\gamma = \frac{1}{\Delta t}$, the resulting iteration for the surface heat equation with explicit Euler time-stepping will be the same as the two-step method of [RM08]. The scheme (4.23) becomes

$$V^{n+1} = \mathsf{E}_p(V^n + \Delta t \mathsf{L} V^n), \tag{4.33}$$

which corresponds to applying one step of first-order explicit time-stepping to the non-extended PDE, followed by performing an extension:

$$\tilde{V}^{n+1} = V^n + \Delta t \mathsf{L} V^n, \tag{4.34a}$$

$$V^{n+1} = \mathsf{E}_p \tilde{V}^{n+1}. \tag{4.34b}$$



(a) Unit sphere, $\mathsf{E}_p\mathsf{L}$

(b) Unit sphere, M

(c) Genus 3 surface, $\mathsf{E}_p\mathsf{L}$

(d) Genus 3 surface, M

Figure 4.14: Eigenvalues of the discrete matrix operator $\mathsf{E}_p\mathsf{L}$ (a) (c), and the penalized operator $\mathsf{M} = \mathsf{E}_p\mathsf{L} - \gamma(\mathsf{I} - \mathsf{E}_p)$ (b) (d), for the unit sphere with $\Delta x = 0.2$ and a genus 3 surface in $\mathbb{R}^3$ [AIM04] with $\Delta x = 0.1$. Interpolation of order $p = 3$ is used in each case, with penalty values $\gamma = 100$ and $\gamma = 400$ respectively.

This case is not a method of lines, and is therefore not covered by our analysis of the discrete scheme.

4.6.6 Alternative discrete forms

Other methods of discretizing the extension operator may lead to better stability properties of the scheme. We note that the discrete operators L and E_p, which are both based on interpolating the function locally and evaluating at grid points, are applied in sequence in the discrete form of the embedding equation.

We wish to evaluate the function $E\Delta v = [\Delta v] \circ \mathrm{cp}$ at a point x_j. Note that the standard discrete differential operator L above (4.1) can be obtained by creating a local interpolant to the function v through the grid points neighbouring x_j, and then differentiating and evaluating the interpolant at x_j. Instead of first evaluating the interpolant $\mathsf{L}v$ at x_j, and then applying a second interpolation to form the extension, we could evaluate the interpolant $\mathsf{L}v$ exactly at the known (non-grid) point $\mathrm{cp}(x_j)$. This avoids the repeated application of the matrix E_p on the differential operator, although the matrix remains in the penalty term $\gamma(\mathsf{I} - \mathsf{E}_p)$. Future work investigates this possibility.

Different orders of interpolation

In [CM14], it was noted that the order of interpolation for the discrete extension operator E does not have to be the same each time it appears in the scheme (3.10).

Recall that we write E_p to denote a matrix constructed with p-th order polynomial interpolation. It could be computationally more efficient to use a lower order interpolation for the differential operator term $\mathsf{E}_p\mathsf{L}$, and a higher value of p for the discrete penalty term $\gamma(\mathsf{I} - \mathsf{E}_p)$. Especially for higher-order equations, where the operator E appears once for each order of differentiation, using different values of p could be advantageous.

4.7 Higher-order and nonlinear operators

Previous formulations of the stabilized operator, such as those in [MR09] and [MBR11], were stated for the Laplace–Beltrami operator, and did not include a general principle for variable-coefficient or nonlinear equations. The penalty method can easily be formulated to include such operators.

In this section, we demonstrate the method with examples of systems of reaction-diffusion equations, and the surface biharmonic operator.

4.7.1 Reaction-diffusion equations

We first show numerical results on semi-linear reaction-diffusion equations, which involve a nonlinearity in the reaction term. The embedding equation is not affected by extension operators applied to this term. For example, when extended to the band, a term u^2 would become

$$E_S u^2 = (E_S u)^2 = v^2, \tag{4.35}$$

under the constraint that $v = E_S u$.

The Gray–Scott reaction-diffusion equations are used as a model of pattern formation [GS83, Pea93]. Formulated on a surface, the equations are given by

$$u_t = \nu_u \Delta_{\mathcal{S}} u - uw^2 + F(1-u), \tag{4.36a}$$
$$w_t = \nu_w \Delta_{\mathcal{S}} w + uw^2 - (F+k)w, \tag{4.36b}$$

where $u, w : \mathcal{S} \to \mathbb{R}$ are surface functions which usually represent the concentration of two interacting chemical species, the diffusion coefficients $\nu_u, \nu_w \in \mathbb{R}$ are taken to be scalar constants, and $F, k \in \mathbb{R}$ are scalar parameters. The system has a steady state at $u = 1$, $v = 0$. If the initial conditions are taken to be a small perturbation about this steady state, diverse patterns may appear for various choices of the parameters F and k.

This example involves nonlinear terms in u and w, which are simple to treat with the method of lines. After extension and discretization in space, the system of equations in the computational band becomes

$$U_t = \nu_u \mathsf{EL} U - UW^2 + F(1-U) - \gamma(U - \mathsf{E}U), \tag{4.37a}$$
$$W_t = \nu_w \mathsf{EL} W + UW^2 - (F+k)W - \gamma(W - \mathsf{E}W). \tag{4.37b}$$

In this formulation, both U and W are discrete band functions. Diffusion constants used are $\nu_u = (1/60)^2$, $\nu_w = \nu_u/2$, with parameters $k = 0.063$, $F = 0.054$ [Mun96].

For reference, the system (4.36) is first solved on the 2D planar domain $[0,1] \times [0,1]$ with standard Cartesian differential operators instead of surface operators, and periodic boundary conditions. The initial condition is a random perturbation about the zero diffusion steady state $(u, w) = (1, 0)$. The grid size is $\Delta x = 0.0083$, and the

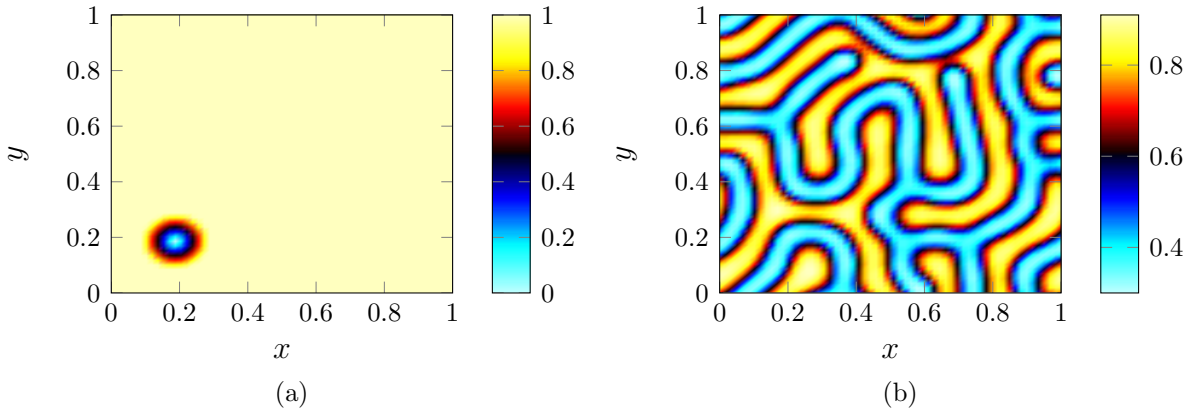


Figure 4.15: Gray–Scott reaction-diffusion (4.36) on a planar region with periodic boundary conditions. Initial conditions for the function u (a) and at time $t = 15000$ (b) are shown.

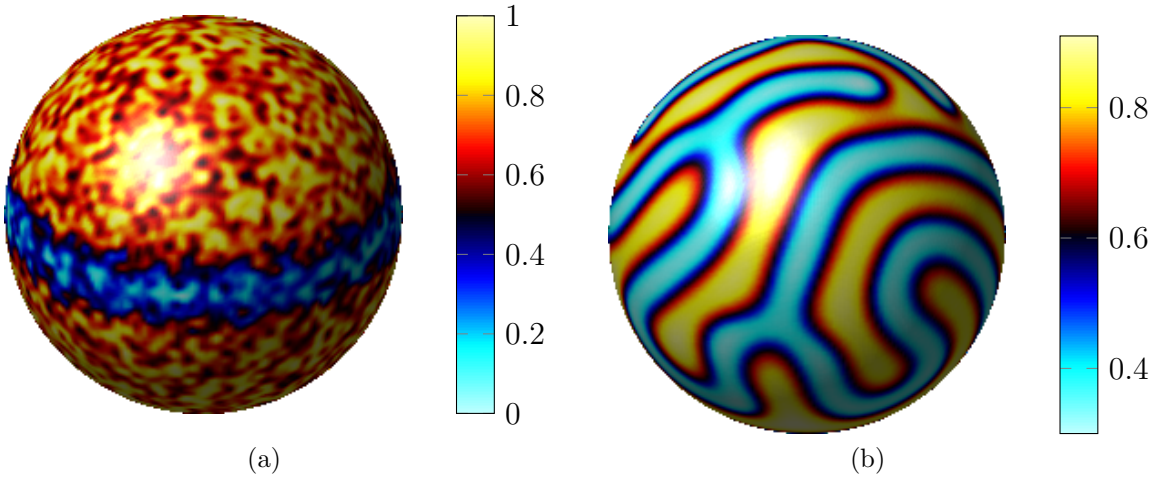


Figure 4.16: Gray–Scott reaction-diffusion (4.36) on a unit sphere: initial conditions for u (a) and value of u at $t = 15000$ (b).

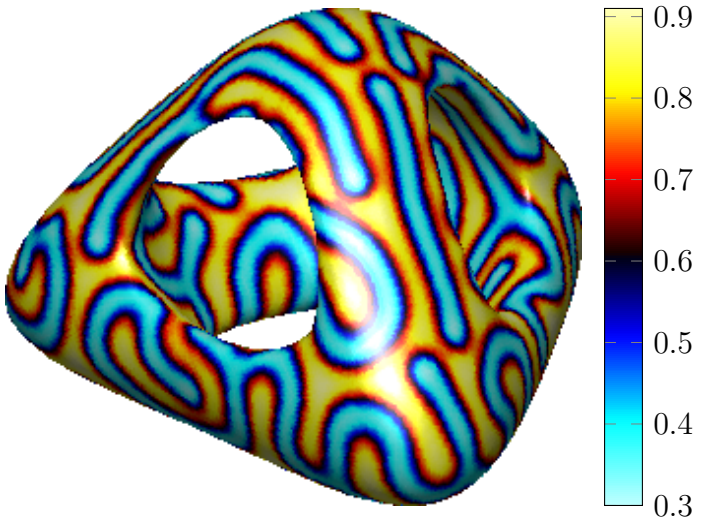


Figure 4.17: Gray–Scott reaction-diffusion (4.36) on a closed genus 3 surface: the function u at $t = 15000$.

time step is $\Delta t = 0.2/\nu_u \Delta x$ using explicit Euler timestepping. The results for u at $t = 15000$ are shown in Figure 4.15.

The equations are then formulated on a unit sphere, using an exact closest point function with a Cartesian grid of size $\Delta x = 0.05$. This system of equations is solved with a implicit-explicit IMEX scheme treating the Laplace–Beltrami operators implicitly, and the nonlinear terms explicitly [Ruu95]. The scheme is a second-order semi-implicit backward difference formula (2-SBDF). Initial conditions for the function u and the steady state at $t = 15000$ are shown in Figure 4.16. Finally, a triangulated genus 3 shape [AIM04] is used, from which a discrete closest point function is calculated using a direct search [MR08]. Using the same set of parameters, similar patterns are found to occur on the plane, sphere and triangulated shape (Figure 4.17).

These examples demonstrate the generality of the method with respect to surface geometry and topology. We observe qualitatively correct results based both on the form of the patterns over the surface, and on the width of the patterns in relation to the space interval Δx.

In Chapter 5, we show further numerical results on nonlinear curvature-dependent diffusion and reaction-diffusion equations, where the nonlinearity appears in the diffusion operator.

4.7.2 Biharmonic operator

As an application of the method to higher-order differential operators, we consider here the biharmonic operator $\Delta_{\mathcal{S}}^2$ (linear fourth order diffusion operator).

Note that a closest point principle (as in Definition 2.5.4) does not apply directly to the surface biharmonic operator. However, with a repeated application of Theorem 2.5.5, the time-dependent surface biharmonic equation $u_t = -\Delta_{\mathcal{S}}^2 u$ can be converted to the form $u_t = -\Delta E \Delta E \bar{u}$ on the surface. Now operating with an extension on both sides of the equation, and substituting $v = E_S u$, we have

$$v_t = -E\Delta E\Delta v, \qquad \text{subject to } v = Ev.$$

As in Section 3.4, the resulting embedding equation is

$$v_t = -E\Delta E\Delta v - \gamma(v - Ev). \tag{4.38}$$

Discretizing as before in (3.10) gives the semi-discrete form

$$V_t = -\mathsf{ELELL}V - \gamma(\mathsf{I} - \mathsf{E})V.$$

We see that in general, the penalty term $\gamma(v - Ev)$ remains, and an additional extension operator E is included. Note that this differs from the procedure in [MR09], which uses the square of the matrix for the Laplace–Beltrami operator in the biharmonic case.

Applying a first-order explicit time-stepping scheme results in the numerical scheme

$$V^{n+1} = V^n - \Delta t \mathsf{E}_p \mathsf{L} \mathsf{E}_p \mathsf{L} V^n - \Delta t \gamma (V^n - \mathsf{E}_p V^n). \tag{4.39}$$

Note that fourth-order equations require two boundary conditions at $\partial B(\mathcal{S})$. The level set framework for fourth-order equations on surfaces is studied in [GBS06], in which both $\nabla u \cdot \nu$ and $\nabla \Delta u \cdot \nu$ are fixed on the boundary. As in the Laplace–Beltrami case, our formulation is consistent with oblique Neumann boundary conditions, since the restriction $v = Ev$ enforces that v and all higher derivatives of v are constant along manifolds of constant cp.

One might expect that if we penalize both the function v as well as the Laplacian Δv, then we could drop the second extension operator inside the Laplacian, and obtain

$$v_t = -E\Delta\Delta v - \gamma_1(v - Ev) - \gamma_2(\Delta v - E\Delta v). \tag{4.40}$$

This is the approach taken in [MR09] for biharmonic problems. However, we cannot simultaneously satisfy both $v = Ev$ and $\Delta v = E\Delta v$, so the equation will not be consistent with the original surface PDE.

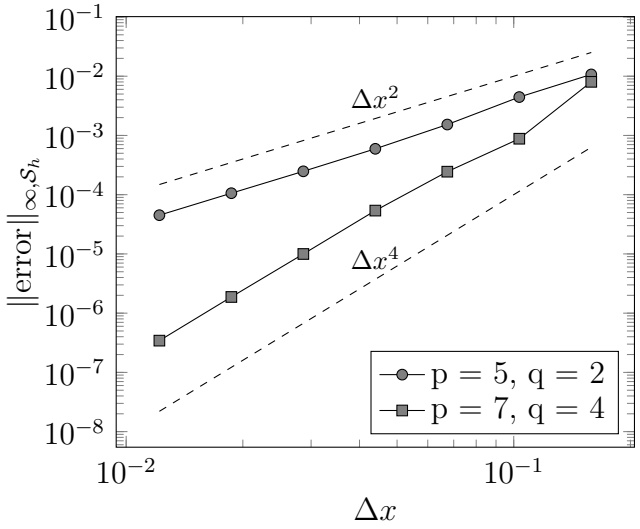


Figure 4.18: Convergence study for the time-dependent surface biharmonic equation on the unit circle (4.39), using BDF2 implicit time stepping, with $\gamma = \frac{4}{\Delta x^2}$, and $\Delta t = \frac{1}{4}\Delta x^2$.

Numerical experiments

The truncation error for the numerical scheme (4.39) in the embedding band can be split into the error in the penalty term, and that in the discretization of the fourth order diffusion term. As in Section 4.4, the penalty term gives a truncation error $\gamma O((\Delta x)^{p+1})$, where p is the order of interpolation in E_p. For the biharmonic operator, however, we have additional restrictions on the interpolation degree p based on the order q of the discrete Laplacian L, since we now apply an outer discrete Laplacian operator to the interpolated result $\mathsf{E}_p\mathsf{L}V^n$.

This method is again demonstrated on the unit circle in 2D. The initial condition $u(\theta, 0) = \cos\theta + \cos 3\theta$ results in the exact solution $u(\theta, t) = e^{-t}\cos\theta + e^{-81t}\cos 3\theta$ at time t.

Explicit time step restrictions become prohibitive for the higher-order operators, so only implicit schemes are considered. Figure 4.18 shows the error in a BDF2 implicit time-stepping scheme. With $\gamma = \frac{4}{\Delta x^2}$ and $\Delta t = \frac{1}{4}\Delta x^2$, we observe convergence with rate $O(\Delta x^{p-3}) + O(\Delta x^q)$.

Chapter 5

Diffusion Equations on Surfaces

Surface diffusion and reaction-diffusion equations are common in mathematical models of biological and physical systems, such as pattern formation on animal coats [Mur03], and models of cell motility [ESV12].

In this chapter, we recall the notion of surface intrinsic diffusion as given in [MM12], and apply the closest point method to various cases of linear, nonlinear, and inhomogeneous reaction-diffusion equations on surfaces. In particular, our method can handle problems depending explicitly on the geometry of the surface, such as PDEs where the diffusion coefficients depend on the curvature.

5.1 Surface diffusion operators

In previous chapters, the Laplace–Beltrami operator was used to define a surface diffusion equation with a constant diffusion coefficient. We now include a non-constant diffusion tensor, and study more general diffusion equations on surfaces, as in [MM12]. The surface diffusion equation considered is

$$\partial_t u = \operatorname{div}_{\mathcal{S}}(A(y)\nabla_{\mathcal{S}} u), \tag{5.1}$$

where $u(y,t)$ is a smooth surface function $u : \mathcal{S} \times (0,T) \to \mathbb{R}$, and A is the diffusion tensor. If $A(y) = I$, the identity map, then (5.1) reduces to the standard Laplace–Beltrami case. If $A(y)$ does not depend on the function u, we call this linear diffusion. In later sections, this will be generalized to include dependence on u and $\mathcal{S}$, the surface function and the surface itself. This can lead to different cases: linear inhomogeneous diffusion (the scalar diffusion coefficient is a non-constant function of y in the domain), nonlinear diffusion (the scalar diffusion coefficient depends on the value of u), and fully anisotropic (the linear or nonlinear diffusion coefficient is a tensor).

In many applications to physical and biological models, the diffusion equation is derived from a continuity equation on the surface. This states that if some quantity stays intrinsic to the surface, then the change in density over a surface region is equal to the flux into or out of that region. If u is the density on the surface of some quantity (for example, the density of a chemical species over the surface), and $\vec{j}$ is the flux over the surface, then

$$\partial_t u + \operatorname{div}_{\mathcal{S}} \vec{j} = 0, \tag{5.2}$$

where we have assumed no sources and sinks on the surface. If the flux is proportional to the gradient of the density, we can write $\vec{j} = -A(y)\nabla_{\mathcal{S}} u$, and obtain the surface diffusion equation (5.1).

If the flux $\vec{j} = -A(y)\nabla_{\mathcal{S}} u$ is a tangential vector field, then we call this *surface intrinsic diffusion* [MM12]. This corresponds to the statement that the quantity is constrained to move only on the surface. This can happen in two different ways: the matrix $A(y)$ maps all vectors in $\mathbb{R}^n$ to vectors tangential to the surface (such as if $A(y)$ is a projection onto the tangent space), or if $A(y)$ maps only tangent vectors to the tangent space (such as a stretching matrix, which does not rotate.) A special case of the latter is scalar diffusivity, where $A(y) = a(y)I$. If neither case is true then $\vec{j}$ is not tangent; this case can still be treated with the closest point method, although we do not pursue it here.

Applying the closest point principles directly to the gradient and divergence operators in (5.1), we obtain the extended operator

$$\begin{aligned} \operatorname{div}_{\mathcal{S}}(A\nabla_{\mathcal{S}} u) &= R \operatorname{div}\left(E[A\nabla[E_S u]]\right) \\ &= R \operatorname{div}\left(A(\operatorname{cp}(x), E_S u)E\nabla[E_S u]\right). \end{aligned} \tag{5.3}$$

5.1.1 Fewer extension operators

In either case above where $\vec{j}$ is tangential, we can reduce the number of extension operators required in the embedding equation, thus simplifying the numerical schemes. The following result from [MM12] shows this, in the special case where the closest point function is based on Euclidean distance.

Theorem 5.1.1 (Surface intrinsic diffusion [MM12])**.** *Let $\mathcal{S}$ be a smooth compact hypersurface, let $u \in C^2(\mathcal{S}, \mathbb{R})$ and let $A : \mathcal{S} \times \mathbb{R} \to \mathbb{R}^{n\times n}$ be C^1. Let E_S correspond to the Euclidean closest point extension operator* ecp *(Definition 2.2.3). If $A(y, u)\nabla_{\mathcal{S}} u$ lies in the tangent space $\mathcal{T}_y\mathcal{S}$ for all $y \in \mathcal{S}$, then*

$$\operatorname{div}_{\mathcal{S}}\left(A(y, u)\nabla_{\mathcal{S}} u\right) = R \operatorname{div}\left(A(\operatorname{cp}(x), E_S u)\nabla[E_S u]\right).$$

Proof. This follows directly from [MM12, Theorem 4.3 and Corollary 4.4]. □

Contrast this with the result (5.3) above; note that here the quantity $\nabla[E_S u]$ is not re-extended before computing the divergence.

As in the case of the divergence and gradient (Section 2.5), we can write Theorem 5.1.1 in the embedding space as

$$E_S \operatorname{div}_{\mathcal{S}} (A(y,u)\nabla_{\mathcal{S}} u) = E \operatorname{div} (A(\operatorname{cp}(x), E_S u)\nabla[E_S u]). \tag{5.4}$$

In this chapter, we chiefly consider the case where the diffusion coefficient is a scalar multiple of the identity map $A(y) = a(y)I$. Since the vector $a(y)\nabla_{\mathcal{S}} u$ is in the direction of $\nabla_{\mathcal{S}} u$, and hence lies in the tangent space, this is diffusion intrinsic to the surface and we may apply Theorem 5.1.1.

5.2 Curvature-dependent diffusion

In certain applications to biological or physical models, the geometry of the surface itself may be explicitly included in the PDE model. Examples include interface motion, or diffusion of chemicals over surfaces, where the diffusion coefficients may depend on the curvature of the surface.

Note that, although the equations in this section are related to the case of mean curvature flow — in which the surface evolves in the normal direction with a velocity proportional to the mean curvature — we do not consider here moving surfaces. Instead, we consider the case where a function on the surface satisfies a PDE depending explicitly on the geometry of the surface, for example through the mean curvature. For a review of geometric PDEs, including mean curvature flow, see [DDE05].

5.2.1 Closest point representation of mean curvature

The implicit representation of the surface, in which points on the surface are defined as the fixed points of the function cp, retains information about the geometry of the surface, including the mean curvature. We can use the closest point function to calculate the curvature directly. Recall the definition of mean curvature in terms of the divergence of the normal vector $\nu(x)$ to a level set,

$$\kappa(x) = \operatorname{tr}(D^2 \operatorname{sd}(x)) = \operatorname{div} \nu(x). \tag{5.5}$$

We will make use of an identity relating the mean curvature $\kappa(y)$ at a point $y \in \mathcal{S}$ to the closest point function cp.

Lemma 5.2.1. *Let $\mathcal{S}$ be a C^2 surface, and let* cp : $B(\mathcal{S}) \to \mathcal{S}$ *be a closest point function. Then the mean curvature κ of $\mathcal{S}$ can be expressed as*

$$\kappa(y) = |[\Delta \mathrm{cp}](y)|, \qquad y \in \mathcal{S}. \tag{5.6}$$

Here, the Laplacian acts row-wise on the vector function cp.

Proof. We first prove that the mean curvature vector $\kappa(x)\nu(x)$ can be expressed as the Laplace–Beltrami operator applied row-wise to the vector identity function on the surface,

$$\kappa(x)\nu(x) = -\Delta_{\mathcal{S}} \mathrm{Id}_{\mathcal{S}}(x). \tag{5.7}$$

To show this, consider the i-th component $[x]_i$ of $x \in B(\mathcal{S})$, so that

$$\begin{aligned} -\Delta_{\mathcal{S}}[x]_i &= -\mathrm{div}_{\mathcal{S}}\left((I - \nu\nu^T)\nabla[x]_i\right) \\ &= -\mathrm{div}_{\mathcal{S}}\left(\hat{e}_i - \nu_i\nu\right) \\ &= \mathrm{tr}\left((I - \nu\nu^T)D(\nu_i\nu)\right) \\ &= (\mathrm{div}\,\nu)\nu_i, \end{aligned} \tag{5.8}$$

and therefore

$$\kappa(x)\nu(x) = (\mathrm{div}\,\nu(x))\nu(x) = -\Delta_{\mathcal{S}}\mathrm{Id}_{\mathcal{S}}(x). \tag{5.9}$$

Applying the closest point principles at the surface to the Laplace–Beltrami operator, we have

$$\kappa(y)\nu(y) = -\Delta(E\,\mathrm{Id}_{\mathcal{S}})(y) = -[\Delta \mathrm{cp}](y), \tag{5.10}$$

since the closest point function is the extension of the identity on the surface. The magnitude κ of the mean curvature is found by taking the norm of this vector. □

For example, for a circle of radius R in $\mathbb{R}^2$, the cp function is $\mathrm{cp}(x) = R\frac{x}{|x|}$, giving

$$-[\Delta \mathrm{cp}](x) = \frac{R}{r^3}x = -\frac{R}{r^2}\nu(x). \tag{5.11}$$

At the surface, $r = R$, so $\kappa(y) = \frac{R}{r^2} = \frac{1}{R}$.

Discretization

The mean curvature result (Lemma 5.2.1) is now discretized as in Chapter 4, with a uniform Cartesian grid in the embedding band. Let L be a discrete Laplacian matrix with truncation error of order q, and let E_p be the matrix evaluating an interpolant through x_j at the points $\mathrm{cp}(x_j)$, using polynomial interpolation of degree p. Each

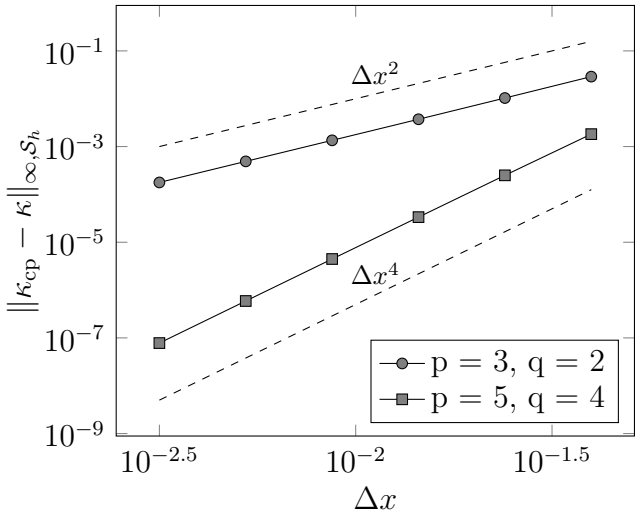


Figure 5.1: Error in the discrete approximation κ_{cp} (5.12) of the mean curvature of an ellipse, using polynomial interpolation of order p, and a qth order finite difference stencil for the Laplacian.

grid point in the embedding space ($\mathbb{R}^n$) is represented as a vector of n components. For each of these n components, we calculate a value

$$[\tilde{\kappa}]_i = \mathsf{LE}_p[\text{Id}]_i, \tag{5.12a}$$

where $[\text{Id}]_i \in \mathbb{R}^J$ is a vector containing the ith component of each Cartesian grid point. We then find the magnitude κ_{cp} of this vector

$$\kappa_{cp} = \left(\sum_{i=1}^{n} [\tilde{\kappa}]_i^2 \right)^{\frac{1}{2}}. \tag{5.12b}$$

The exact curvature can be calculated directly from the parameterization, and compared to the curvature computed from the discrete cp function. We demonstrate an example of a surface of non-constant curvature in $\mathbb{R}^2$: an ellipse parameterized by $(x_1, x_2)(s) = (a\cos s, b\sin s)$, with $a = 1.5$, and $b = 0.75$. The closest point on the surface $\text{cp}(x_j)$ to each Cartesian grid point x_j is computed with Newton's method. Figure 5.1 shows the error in the resulting curvature on the surface. The error depends on the order p of the polynomial interpolation and q of the discretization of the Laplacian, as $O(\Delta x^{p-1}) + O(\Delta x^q)$, which is optimal in the discretization.

5.2.2 Curvature-dependent diffusion on curves

Inhomogeneous surface diffusion is demonstrated with an example of a diffusion equation where the diffusivity depends on the curvature of the surface. Consider the

equation

$$u_t(y) = \mathrm{div}_{\mathcal{S}}(a(y)\nabla_{\mathcal{S}} u(y)), \tag{5.13}$$

where we choose the inhomogeneous diffusivity $a(y)$ related to the mean curvature $\kappa(y)$ of the surface by

$$a(y) = \frac{1}{1 + |\kappa(y)|}.$$

Note that although the scalar diffusion coefficient varies over the surface, this is not a case of anisotropic diffusion (which involves a tensor diffusion coefficient).

We will first consider a one-dimensional surface (curve) embedded in $\mathbb{R}^2$. Following the discretization of Section 3.4, we compute the discrete mean curvature based on (5.12). Note that we also extend the diffusivity and curvature fields throughout the embedding band using

$$\bar{a} := Ea = \frac{1}{1 + |E\kappa|}. \tag{5.14}$$

From this we compute the vector $\mathbf{a}$ consisting of the values of $\bar{a}$ at each grid point.

The PDE (5.13) is simple to solve numerically using the penalty approach detailed in Chapter 3. Since the diffusion coefficient $a(y)$ is a scalar, we have that the term $a(y)\nabla_{\mathcal{S}} u(y)$ lies in the tangent space of the surface. This is therefore a case of surface intrinsic diffusion, and Theorem 5.1.1 applies. The embedded surface with penalty term is

$$v_t = E\,\mathrm{div}\ (\bar{a}\nabla v) - \gamma(v - Ev). \tag{5.15}$$

A standard scheme based on forward and backward differences is used to discretize the variable-coefficient diffusion term as follows, yielding the semi-discrete form

$$V_t = \mathsf{E}\Big[\mathsf{D}_b^x\big(\mathsf{A}_f^x\mathbf{a}\,\mathsf{D}_f^x V\big) + \mathsf{D}_b^y\big(\mathsf{A}_f^y\mathbf{a}\,\mathsf{D}_f^y V\big)\Big] - \gamma\big(\mathsf{I} - \mathsf{E}\big)V, \tag{5.16}$$

where D_b and D_f are the backward and forward finite difference matrices in the direction indicated by the superscripts. Similarly, the matrices A_f refer to forward two-point averages of the pointwise diffusivity values. That is, the half-point diffusivities are approximated by the averages:

$$a_{i+\frac{1}{2},j} = \frac{a_{i+1,j} + a_{ij}}{2}, \qquad a_{i,j+\frac{1}{2}} = \frac{a_{i,j+1} + a_{ij}}{2}.$$

The discrete problem is then solved with an explicit time-stepping method, such as forward Euler.

The method is implemented on the ellipse of Section 5.2.1. Figure 5.2a shows the computational band around the ellipse in $\mathbb{R}^2$, and the computed curvature in the

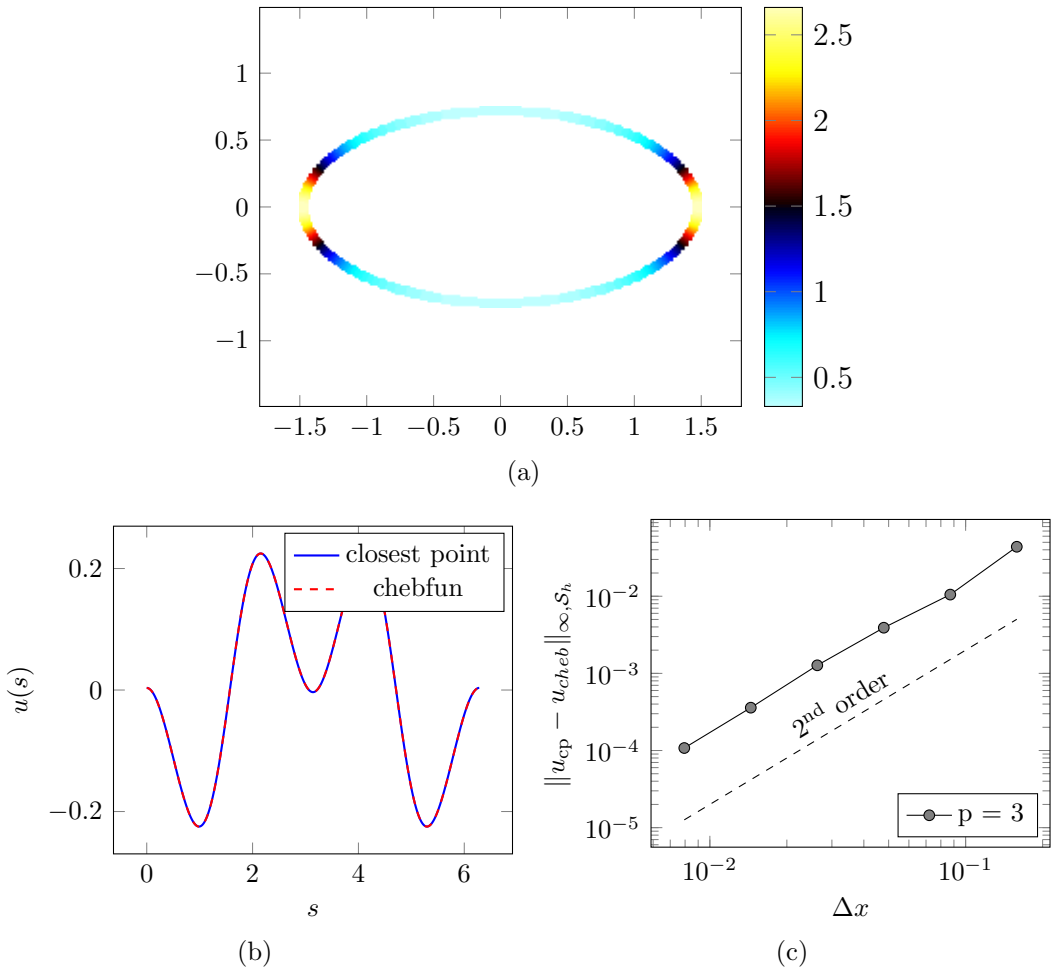


Figure 5.2: Curvature-dependent diffusion on an ellipse in $\mathbb{R}^2$. Computed curvature in the computational band (a). Results of curvature-dependent diffusion (5.13) at time $t = 0.5$, using our method and Chebfun [DHT14] (b). Surface maximum difference between the two solutions as a function of Δx (c).

band, with $\Delta x = 0.01$. The equation is discretized using a second-order scheme in space and explicit Euler in time, with $\Delta t = \frac{1}{4}\Delta x^2$, interpolation of order $p = 3$, and $\gamma = \frac{4}{\Delta x^2}$. Initial conditions are $u(s, 0) = \cos(3s)$, where s is used to parameterize the ellipse as in Section 5.2.1. The resulting solution parameterized by s at time $t = 0.5$ is shown in Figure 5.2b. Solutions are also computed using Chebfun [DHT14] (based on the parameterization) and used as a reference. The convergence of the difference between the solutions is shown in Figure 5.2c.

5.2.3 Reaction-diffusion equations with curvature-dependent parameters

We now demonstrate the method on more complex curvature-dependent problems, in particular systems of reaction-diffusion equations with curvature-dependent coefficients. Recall the Gray–Scott model of Section 4.7.1, where a diffusion-driven instability leads to pattern formation [Pea93]. Patterns form when the difference in diffusion coefficients of two chemical species leads to an instability; if the diffusivities vary over the surface, patterns may form only in certain areas.

In Section 4.7.1, we expect the ratio of diffusion coefficients $\nu_w = \frac{\nu_u}{2}$ to form a patterned steady state. We now consider a case where ν_w varies with curvature of the surface.

The approach of the numerical method in this section extends that of Section 4.7.1. The Gray–Scott scheme (4.37) is solved on a surface of non-constant curvature, with ν_w related to ν_u by

$$\nu_w = \nu_u / \big(3 - \tfrac{2}{c_1 - c_2}(\kappa - c_2)\big),$$

where c_1 and c_2 are the maximum and minimum curvatures of the surface. At areas of low curvature, the ratio will be close to 3, while areas of high curvature will have equal coefficients.

Based on an understanding of the Grey-Scott model in planar regions, we expect patterns to form preferentially in low-curvature areas, as demonstrated in Figure 5.3. Initial conditions are taken to to be the steady state $(u_0, v_0) = (1, 0)$, with random Gaussian noise added. Figure 5.3a shows the ratio of the diffusivities calculated from the mean curvature of an ellipsoid. Steady states for u demonstrating spot and stripe formation on this surface are shown in Figures 5.3b and 5.3c. Parameters used are $F = 0.026$, $k = 0.061$ for spots and $F = 0.054$, $k = 0.063$ for stripes, as expected on flat domains [Mun96]. A similar system is solved on a parameterized surface with the shape of a red blood cell (derived in [EF72] and used with reaction-diffusion models in [FW12]), this time with unequal coefficients at areas of high curvature. Figure 5.4 shows spot and stripe formation on the high-curvature regions of the surface.

Discussion

The results of the numerical experiments reveal patterns of shape and size typical in reaction-diffusion systems [Pea93], formed in the high- or low-curvature regions expected from the analysis of the diffusion coefficients. These problems are therefore

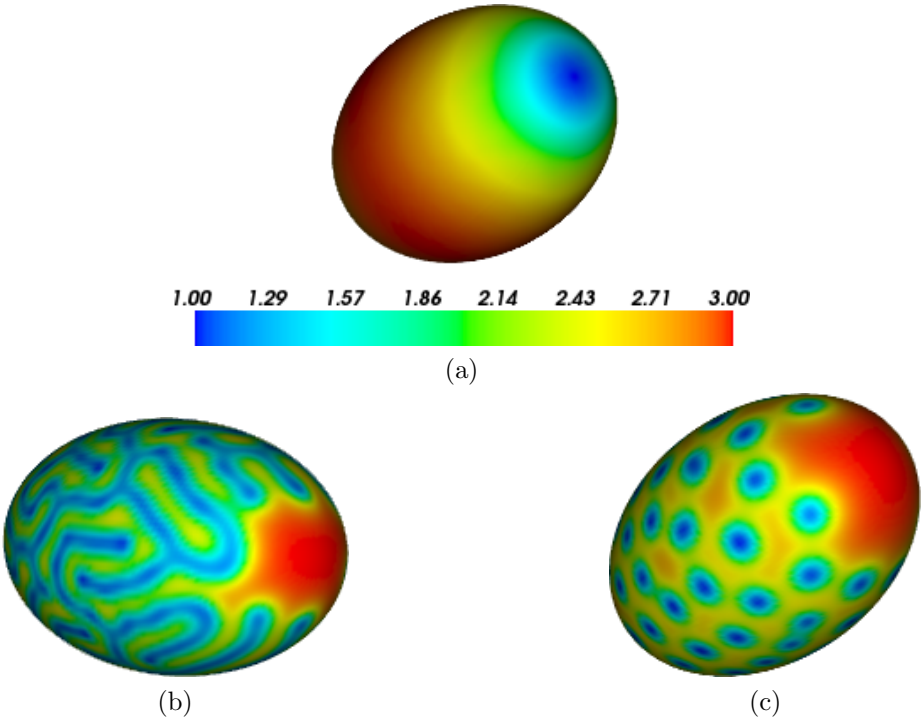


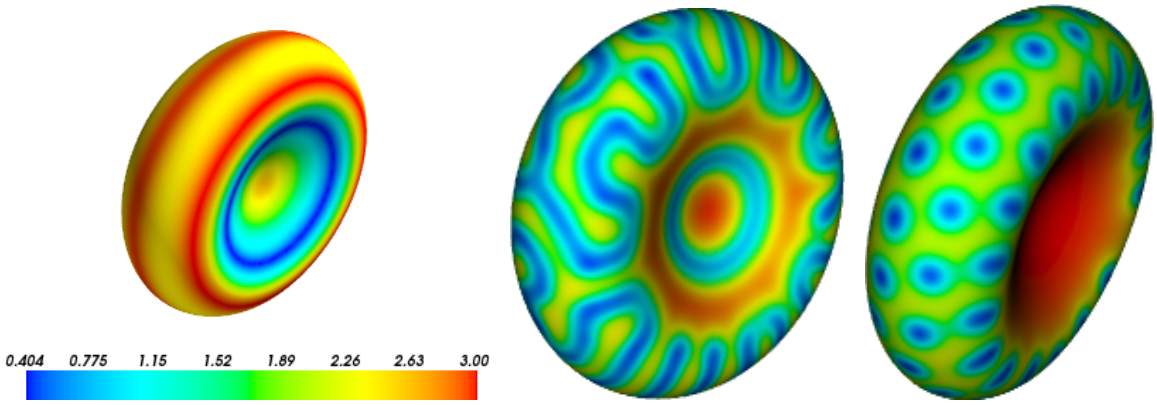
Figure 5.3: Curvature-dependent reaction-diffusion on an ellipsoid. Ratio of diffusion coefficients, inversely proportional to surface curvature (a). Stripe (b) and spot (c) formation in regions of low curvature using the Gray–Scott model (4.37).

good indicators of the effectiveness of the method in handling curvature-dependent problems on complex geometries.

5.3 Nonlinear diffusion

In the examples above, the coefficient $a(y)$ depends only on the surface itself, so the diffusion operator is linear. In the following chapter, we will turn to the nonlinear case, where $a(u)$ is a scalar function of the surface function u. Again, the treatment with the closest point method — in particular the number of extension operators in the embedding equation — depends on the fact that the scalar diffusion is always tangential to the surface. A reduced formulation with fewer extensions can be used, as in Section 5.1.

Note that some authors call these scalar diffusion operators anisotropic, e.g. [PM90]. Since the diffusion is always tangential, we will refer to this as *nonlinear* diffusion, and reserve the term *anisotropic* diffusion for the case when $A(u)$ is a tensor [Wei98].



(a) (b) (c)

Figure 5.4: Curvature-dependent reaction-diffusion on a red blood cell shape. Ratio of diffusion coefficients, proportional to surface curvature (a). Stripe (b) and spot (c) formation in regions of high curvature using the Gray–Scott model (4.37).

Chapter 6

Applications in Image Processing on Surfaces

Images defined on curved surfaces are common in fields such as computer vision, graphics, and medical imaging, in part due to the growing popularity of 3D image acquisition technologies. For example, point cloud scanners return a set of embedded points sampling a surface, with colour data at each point. In medical imaging, volumetric scanners capture 3D data, from which image values lying on certain surfaces can be obtained through segmentation. PDEs on surfaces can be used for feature extraction, such as brain surface mapping [GWC+04].

During the scanning process, occlusions could create surfaces with image data which is corrupt or missing. These data may also be subject to noise, as in the case of standard planar or volumetric images. PDE-based techniques can be used to post-process these images, for example, to fill in missing pieces of the surface itself [VCBS03, GS03], or to edit the image by adding or removing features such as text. As an application of the closest point method, we adapt some existing methods for 2D PDE-based image denoising and inpainting to surface images.

We begin with a review of surface images, which are defined here in the space of functions of bounded variation on surfaces. We then describe some common denoising and inpainting PDEs, and write down in each case an analogous equation for surface images. The surface equation can be extended into the surrounding space to create a closest point embedding equation, which is then discretized using standard schemes from numerical image processing.

The first example is an application of nonlinear diffusion, using both the explicit closest point method and the penalty formulation for surface diffusion operators. The method of nonlinear diffusion filtering known as Perona–Malik denoising is illustrated

with examples of noisy images on various triangulated and smooth surfaces. Some of these numerical examples based on the explicit closest point method first appeared in [BvGMM13], and are the work of the author.

Alternative approaches to denoising based on variational methods are then discussed, in particular the popular total variation (TV) denoising scheme [ROF92]. Results of TV denoising on surface colour images are presented.

We investigate parallels between the penalty method and the constrained optimization problems arising from variational formulations in image processing. Finally, variational methods are applied to image inpainting on surfaces, and we demonstrate some examples of total variation surface inpainting.

6.1 Images on surfaces and BV functions

Many image processing methods are based on PDE models, where the PDE describes the process such as diffusion which will enhance the image, or variational models, in which some form of energy is minimized to obtain an optimal image in some norm.

A greyscale image is typically represented as a matrix of values, with values between 0 and 1 representing the shade between black (0) and white (1) of a particular pixel. This matrix can be considered as a discretization of the continuous limit, treating an image as a function of a continuous variable. In image processing, the continuous domain is usually the rectangle $\Omega = (x_1, x_2) \in [a, b] \times [c, d] \subset \mathbb{R}^2$, and the image is represented as a function $u(x_1, x_2) : \Omega \to [0, 1]$. The function u is not required to be continuous, but rather lies in a function space chosen as the image model.

A natural function space for images is functions of bounded variation $BV(\Omega)$ [Giu84]. This space allows functions with sharp discontinuities or edges, but also ensures some regularity of the functions, so is well suited to the mathematical description of natural images [CS03].

Given a function $u \in L^1(\Omega)$, the total variation of u is written

$$|Du|(\Omega) = \sup \left\{ \int_\Omega u(x) \operatorname{div} g(x) dx : g \in C_0^1(\Omega), \|g\|_{L^\infty(\Omega)} \leq 1 \right\}. \tag{6.1}$$

If $u \in C^1(\Omega)$, then the total variation reduces to (using integration by parts)

$$|Du|(\Omega) = \int_\Omega |\nabla u(x)| dx. \tag{6.2}$$

A space BV of functions of bounded variation in Ω is then defined as

$$BV(\Omega) = \{u : u \in L^1(\Omega),\ |Du|(\Omega) < \infty\}. \tag{6.3}$$

These functions are roughly those discontinuous functions whose derivative exists almost everywhere.

We now turn to surface images, and give a definition from [BAL07] of functions of bounded variation on surfaces. We assume in this section that the surface $\mathcal{S}$ is smooth, and represent the surface images as functions $u : \mathcal{S} \to [0, 1]$.

Definition 6.1.1. *Let $\mathcal{S}$ be a smooth compact surface, and let $u \in L^1(\mathcal{S})$. The total variation of u is given by*

$$|D_{\mathcal{S}}u|(\mathcal{S}) = \sup\left\{\int_{\mathcal{S}} u(s)\mathrm{div}_{\mathcal{S}}g(s)ds : g \in \Gamma(\mathcal{TS}), |g| \leq 1\right\}, \tag{6.4}$$

where $\Gamma(\mathcal{TS})$ is the set of C^1 tangent vector fields on $\mathcal{S}$.

An analogous space $BV(\mathcal{S})$ is defined as the surface functions u such that the total variation of u is bounded,

$$BV(\mathcal{S}) = \{u : u \in L^1(\mathcal{S}),\ |D_{\mathcal{S}}u|(\mathcal{S}) < \infty\}. \tag{6.5}$$

These functions are introduced in [BAL07]. In [LC11], the co-area formula and boundary perimeter formula are extended to the case of BV functions on surfaces, so that we can use the space $BV(\mathcal{S})$ to derive variational image processing methods on surfaces, analogous to those in $\mathbb{R}^2$.

6.2 Nonlinear diffusion in image processing

Nonlinear diffusion operators have been popular in image processing since their introduction by Perona and Malik [PM90]. Denoising with linear Gaussian diffusion results in blurry images, and does not maintain contrast at edges. Nonlinear diffusion, however, can preserve edges if the diffusion at edges is small compared to that in more homogeneous regions. A further generalization is anisotropic diffusion, such as coherence-enhancing diffusion [Wei98], in which the direction of diffusion is nonconstant across the image.

Note that the term surface denoising could also refer to smoothing the values representing the position of the surface itself (for example, in [CDR04]). In all cases considered here, the surface itself is fixed, and the value of the texture or image function u is smoothed.

6.2.1 Denoising surface images with Perona–Malik diffusion

In this section, we consider surfaces of dimension $k = 2$, embedded in $\mathbb{R}^3$, although the approach is general. We apply Perona–Malik nonlinear diffusion to surface images as in [BvGMM13, Bid12], now using the penalty approach for the closest point embedding equation.

The Perona–Malik diffusion model on a planar image is [PM90]

$$u_t = \text{div}\,(g(|\nabla u|)\nabla u), \tag{6.6}$$

where the function g depends on the magnitude of the gradient of the image. Several choices of this function have been proposed, including the popular choice [PM90]

$$g(s) = \frac{1}{1 + \frac{s^2}{\lambda^2}}. \tag{6.7}$$

The parameter λ describes a threshold on the gradient, which determines the location of edges. In regions where $|\nabla u|$ is small, the value of $g(|\nabla u|)$ is close to 1. The PDE is diffusive in these regions, which have intensity which is constant or near constant. The PDE therefore acts like standard diffusion, and smoothes out noise. However, near edges, the magnitude of the gradient of u will be large compared to λ. In these regions, the diffusion coefficient $g(|\nabla u|)$ is small, so edges do not experience the same amount of blurring.

We now consider a Perona–Malik equation for a surface image. Given an initial image u_0 on $\mathcal{S}$, the equation for the function $u : \mathcal{S} \to \mathbb{R}$ becomes

$$u_t = \text{div}_{\mathcal{S}}(g(|\nabla_{\mathcal{S}} u|)\nabla_{\mathcal{S}} u). \tag{6.8}$$

Note that since the diffusion is tangential to the surface, no intensity is lost through diffusion off the surface.

This in-surface formulation of denoising via nonlinear diffusion is applied to surface images in [BvGMM13], using the explicit closest point method of [RM08] (Section 2.6). The examples from [BvGMM13] which are reproduced in this section are the work of the author. We now also investigate the use of the penalty formulation, which results in a simple numerical implementation.

The embedding equation in $B(\mathcal{S})$ is

$$v_t = E\text{div}\,(g(|E\nabla v|)\nabla v), \tag{6.9a}$$

with side condition

$$v = Ev. \tag{6.9b}$$

Again we have surface intrinsic diffusion with a scalar diffusion coefficient, so extension operators are only required outside the divergence operator.

Discretization

The discretized scheme is similar to the case of curvature-dependent diffusion of Section 5.2. Considering the standard Cartesian equation without extensions

$$v_t = \operatorname{div}\left(g(|\nabla v|)\nabla v\right), \tag{6.10}$$

or in three dimensional form,

$$v_t = \frac{\partial}{\partial x}\Big[g\Big(\sqrt{v_x^2 + v_y^2 + v_z^2}\Big)\, v_x\Big] + \frac{\partial}{\partial y}\Big[g(\ldots)\, v_y\Big] + \frac{\partial}{\partial z}\Big[g(\ldots)\, v_z\Big], \tag{6.11}$$

we write down a standard finite difference scheme using forward Euler time-stepping,

$$\begin{aligned}\tilde{V}_{ijk}^{n+1} = V_{ijk}^n + \Delta t\Bigg[&\frac{g^n_{i+\frac{1}{2},j,k}\, D_f^x V^n_{ijk} - g^n_{i-\frac{1}{2},j,k}\, D_b^x V^n_{ijk}}{\Delta x} + \\ &\frac{g^n_{i,j+\frac{1}{2},k}\, D_f^y V^n_{ijk} - g^n_{i,j-\frac{1}{2},k}\, D_b^y V^n_{ijk}}{\Delta y} + \\ &\frac{g^n_{i,j,k+\frac{1}{2}}\, D_f^z V^n_{ijk} - g^n_{i,j,k-\frac{1}{2}}\, D_b^z V^n_{ijk}}{\Delta z}\Bigg].\end{aligned} \tag{6.12a}$$

Here D_f^α and D_b^α indicate forward and backward finite differences, in the direction indicated by the superscript α, on a grid spacing of Δx.

The half-point diffusivities are approximated by averages

$$\begin{aligned} g^n_{i+\frac{1}{2},j,k} &= \tfrac{1}{2}(g^n_{i+1,j,k} + g^n_{ijk}), \\ g^n_{i-\frac{1}{2},j,k} &= \tfrac{1}{2}(g^n_{i-1,j,k} + g^n_{ijk}), \end{aligned} \tag{6.13}$$

where the nodal g^n_{ijk} are computed using central finite differences

$$g^n_{ijk} = g\Big(\sqrt{(D_c^x v^n_{ijk})^2 + (D_c^y v^n_{ijk})^2 + (D_c^z v^n_{ijk})^2}\Big). \tag{6.14}$$

Now including extensions and the penalty term, as in Section 5.2, we have

$$V_t = \mathsf{E}\Big[\mathsf{D}_b^x\big(\mathsf{A}_f^x \mathsf{Eg}\ \mathsf{D}_f^x V\big) + \mathsf{D}_b^y\big(\mathsf{A}_f^y \mathsf{Eg}\ \mathsf{D}_f^y V\big) + \mathsf{D}_b^z\big(\mathsf{A}_f^z \mathsf{Eg}\ \mathsf{D}_f^z V\big)\Big] - \gamma(\mathsf{I} - \mathsf{E})V, \tag{6.15}$$

where the A_f^α matrics denote forward two-point averages in the direction indicated. The scheme (6.9) can also be applied to the special case of Gaussian diffusion, in which case the diffusion coefficient g is taken to be a constant.

6.2.2 Examples

For all examples below, the Perona–Malik threshold parameter is chosen to be $\lambda = 5$. Noise is added in the embedding space, normally distributed with mean 0, standard deviation 1, and amplitude 0.3. The time step is chosen as $\Delta t = 0.15\Delta x^2$, and degree $p = 3$ interpolation is used. The penalty parameter is $\gamma = 6/\Delta x^2$.

Triangulated surface: Fish

Both Gaussian and nonlinear diffusion are applied on a triangulated surface representing a fish [AIM05]. A closest point function is calculated at discrete Cartesian grid points, as in [MR08], and the triangulation is then discarded. The fish is 0.95 units long, on a Cartesian grid of spacing $\Delta x = 0.002$. With a time step of $\Delta t = 0.15\Delta x^2$, the algorithm is run until $t = 0.0015$. Results are shown in Figure 6.1, visualized using a ray-tracer described in [AMT+12]. Note that the ray-tracer uses the implicit closest point representation of the surface to produce the visualizations directly, rather than a triangulation of the surface.

The Gaussian diffusion produces blurring as expected (Figure 6.1b), while the Perona-Malik model results in localized edges. Since the closest point method uses polynomial interpolation, we expect that we may see some overshoot at sharp edges, observed in Figure 6.1c.

World map

The method is then applied to a map of the world which has been texture-mapped onto a sphere. The noise is created in the embedding space, as though the image were acquired from a three-dimensional source. Figure 6.2 shows the noisy image, and the results of denoising using linear Gaussian diffusion and Perona–Malik edge-preserving diffusion. Using a unit sphere, the embedding grid spacing is $\Delta x = 0.01$. The stopping time is again $t = 0.0015$, with a time step of $\Delta t = 0.15\Delta x^2$.

Metric distortions on the sphere

For this simple surface, a global parameterization of the surface can be constructed. It would therefore be possible to map the surface image to a plane using the surface metric and perform standard Perona–Malik denoising on the planar image. However, since the noise is produced in the embedding space, the scale space would also need to be scaled by the metric, which complicates the PDE model considerably, in particular since some coefficients would be singular at the poles. A naïve version without

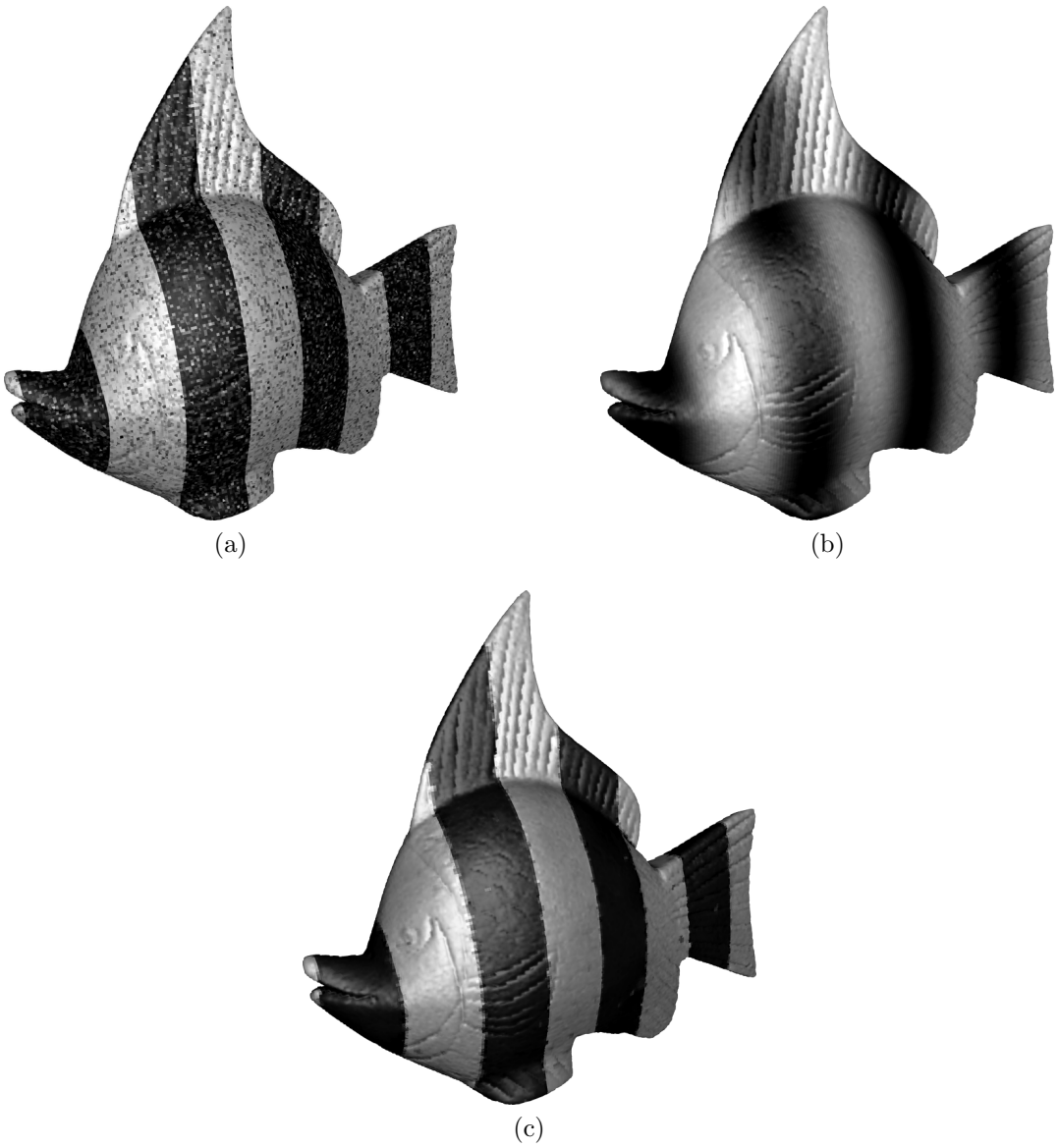
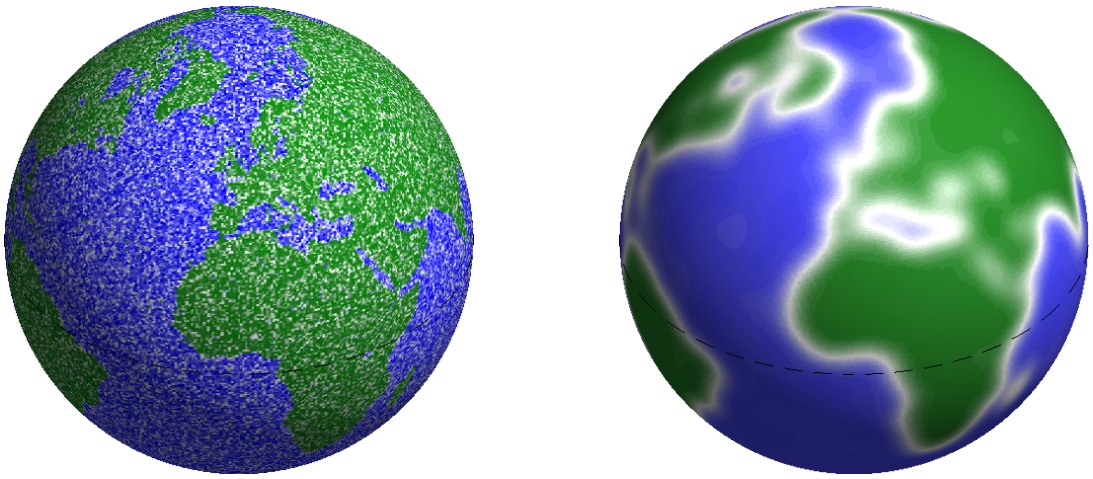
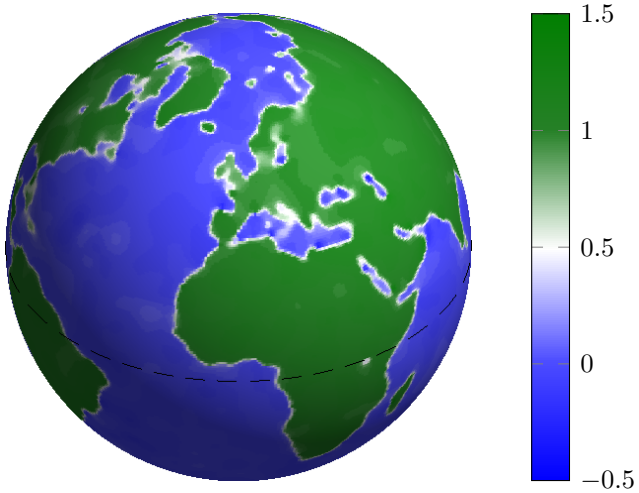


Figure 6.1: Denoising on a triangulated surface. Noisy surface image (a), Gaussian diffusion showing blurring (b) and Perona–Malik nonlinear diffusion (c). Parameters used are $\Delta x = 0.002$, $\lambda = 5$, $t = 0.0015$. Fish geometry obtained as a triangulation from [AIM05], and converted to a closest point representation as in [MR08].

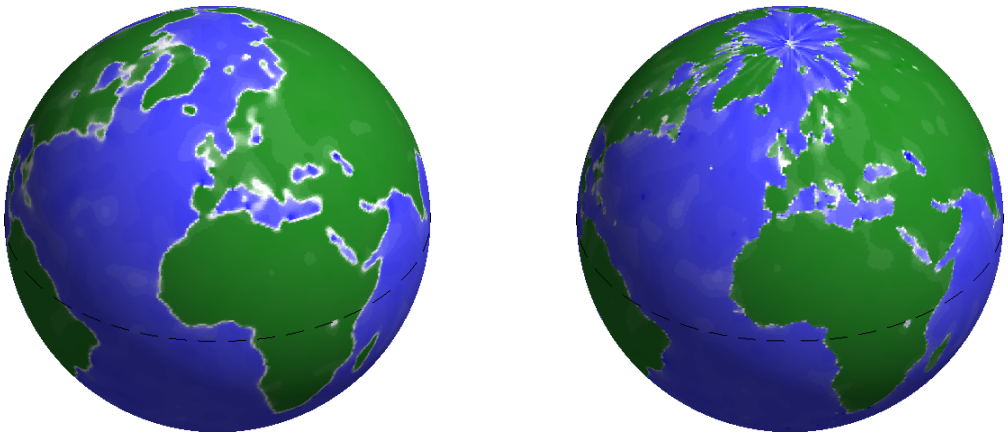


(a) (b)



(c)

Figure 6.2: A noisy image on a sphere (a), and results of denoising at $t = 0.0015$ using surface Gaussian diffusion (b), and Perona–Malik nonlinear diffusion (c). The parameters used are $\Delta x = 0.01$, $\lambda = 5$, $\gamma = 6/\Delta x^2$.



(a) (b)

Figure 6.3: Perona–Malik diffusion on a sphere, using the closest point formulation on a surface (a) and using a naïve map to a 2D version (b). Note the lack of denoising near the pole in (b), a consequence of using the wrong metric.

including the metric terms would solve (6.6) on the plane, before mapping the figure back to the surface. Figure 6.3 shows the difference between the diffusion applied in the embedding space, and the version denoised on a plane. Distortion near the pole can clearly be seen in the planar version.

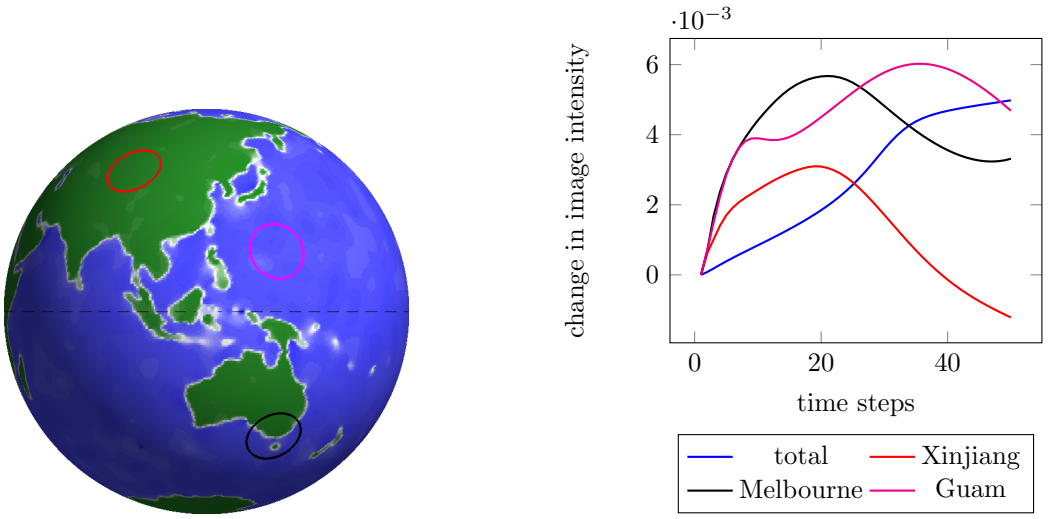
Considering the simpler case of the Laplace–Beltrami operator, the PDE obtained from corrections due to the surface metric would be

$$u_t = \operatorname{div}_{\mathcal{S}}(\nabla_{\mathcal{S}} u) = \sum_{i,j=1}^{k} \frac{1}{\sqrt{G}} \sum_{i=1}^{k} \frac{\partial}{\partial x^i} \left(\sqrt{G} \sum_{j=1}^{k} g^{ij} \frac{\partial u}{\partial x_j} \right), \tag{6.16}$$

where $G = \det(g_{i,j})$, and the metric can be induced from the local parameterization $X \in C^2(V, \mathbb{R}^n)$, $\theta \in V$ of the surface

$$g_{i,j}(\theta) = \frac{\partial X}{\partial \theta_i}(\theta) \cdot \frac{\partial X}{\partial \theta_j}(\theta), \qquad i, j = 1, ..., n. \tag{6.17}$$

A similar equation for the Perona–Malik scheme involves further corrections due to the nonlinear diffusion coefficient. In contrast, the closest point method is simple to formulate without knowledge of the surface metric, for a large class of surfaces which do not necessarily admit a parameterization.



(a) (b)

Figure 6.4: Solution of Perona–Malik denoising, with regions of integration marked (a) and change in average intensity $\int_\Omega u(s,t)ds - \int_\Omega u_0(s)ds$ in each region (b).

Conservation of intensity

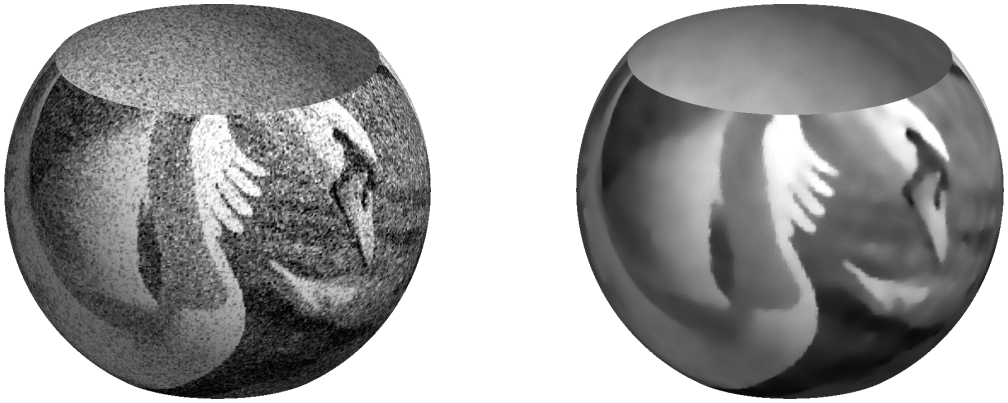
We now examine the effect of the algorithm on the average intensity of the image. Integrating (6.8) over the surface $\mathcal{S}$, we have

$$\int_{\mathcal{S}} u_t = \int_{\mathcal{S}} \operatorname{div}_{\mathcal{S}}(g(|\nabla u|)\nabla u)ds = 0, \tag{6.18}$$

since $\mathcal{S}$ has no boundary. The global average image intensity should therefore remain constant over time. Figure 6.4 shows the world map example above, with the total average image intensity over the sphere plotted over time. We see that the change is of order $10^{-2} = \Delta x$. In addition, we plot the change in image intensity in smaller subregions. One region each of high and low intensity ($u_0 = 1$ and $u_0 = 0$ respectively) is chosen, as well as a region intersecting an edge. Since the image is close to piecewise linear, we expect the average intensity to be approximately preserved in these regions, as in the global case.

Surface with boundary: Fishbowl

In the derivation of the method, the surface $\mathcal{S}$ was defined for simplicity to be a submanifold without boundary. The definition of the Euclidean closest point function can easily be extended to surfaces with boundary (see Section 2.6.2 and [RM08, MBR11]). We show results on a fishbowl surface, which has a boundary which is a circle in



(a) (b)

Figure 6.5: Perona–Malik diffusion of a texture-mapped image on a surface with a boundary. Original noisy surface image (a) and denoised image (b). Parameters used are $\Delta x = 0.007$, $\lambda = 5$, $\gamma = 6/\Delta x^2$. Swan image used with permission from Harry Biddle [BvGMM13].

$\mathbb{R}^3$. Using the standard matrix implementation of the closest point method will impose a first-order homogeneous Neumann boundary condition at the surface boundary [RM08], which is an appropriate boundary condition for many image processing tasks.

Figure 6.5 shows the results of Perona–Malik diffusion on a surface with boundary.

6.3 Variational methods in image processing

6.3.1 Overview

Variational methods for image processing determine an optimal image $\hat{u}$, the function which minimizes a certain energy based on a given observed image u_0. We define an energy functional

$$J(u) := R(u) + H(u), \tag{6.19}$$

where the term $R(u)$ is called the regularizing term, while $H(u)$ is a fidelity term, which keeps the solution sufficiently close to the original image u_0.

Variational methods differ in their choice of the regularizing term, and the function spaces chosen for the fidelity and minimization. A typical choice is to minimize over $BV(\Omega)$, with an L^2 or L^1 fidelity term. This leads to minimization problems of the form

$$\hat{u} = \operatorname*{argmin}_{u \in BV(\Omega)} |J(u)|, \tag{6.20}$$

which can be solved, for example, by obtaining the associated Euler-Lagrange equations and using a gradient descent scheme.

6.3.2 Denoising with TV

A popular denoising method was introduced by Rudin, Osher and Fatemi [ROF92, RO94], using the total variation $|Du|(\Omega)$ of the image as the regularizing term $R(u)$. This method is known as TV denoising or ROF denoising.

The L^2 norm is used for the fidelity term,

$$H(u) = \frac{\lambda}{2}\|u - u_0\|^2_{L^2(\Omega)}, \tag{6.21}$$

where u_0 is the observed image, and λ is a penalty parameter. The energy functional is thus given by

$$J(u) = |Du|(\Omega) + \frac{\lambda}{2}\|(u - u_0)\|^2_{L^2(\Omega)}, \tag{6.22}$$

and the resulting gradient descent scheme in Ω is

$$u_t = \operatorname{div}\left(\frac{\nabla u}{|\nabla u|}\right) + \lambda(u - u_0). \tag{6.23}$$

As in Perona–Malik diffusion, the equation contains a nonlinear diffusion term, where the diffusivity coefficient $1/|\nabla u|$ becomes small at sharp edges of u. The minimizer of $J(u)$ exists and is unique in $\mathrm{BV}(\Omega)$ for $\lambda > 0$ [AV94, CL97].

We now consider a surface image $u : \mathcal{S} \to \mathbb{R}$, and write the denoising equation as

$$u_t = \operatorname{div}_{\mathcal{S}}\left(a(|\nabla_{\mathcal{S}} u|)\nabla_{\mathcal{S}} u\right) + \lambda(u - u_0), \tag{6.24}$$

with scalar diffusivity $a(|\nabla_{\mathcal{S}} u|) = \frac{1}{|\nabla_{\mathcal{S}} u|}$. This has a similar form to the surface nonlinear diffusion equations studied in the previous chapter, with an additional regularizing term.

In the formulation of Chapter 3, the continuous embedding equation associated with (6.24) is

$$v_t = E\operatorname{div}\left(a(E|\nabla v|)\nabla v\right) - \lambda(v - v_0) - \gamma(v - Ev), \tag{6.25}$$

with initial condition $v_0 = Eu_0$. We have again used the fact that the second extension of the gradient is unnecessary when using the Euclidean closest point ecp (Section 5.1 and [MM12]). Note that this equation involves two penalty terms, the first relating the solution to the initial observed image v_0, and a second term constraining the solution to be a closest point extension.

Since the diffusion coefficient $1/|\nabla v|$ is degenerate in regions of constant v, the coefficient is modified to $1/|\nabla v|_\epsilon$, with

$$|\nabla v|_\epsilon := \sqrt{\epsilon^2 + |\nabla v|^2}, \tag{6.26}$$

for some positive ϵ.

The discretization is similar to the schemes above for curvature-based and nonlinear diffusion. Results are shown in Figure 6.6 for both total variation and Perona–Malik denoising on the fishbowl surface, using a spatial grid size $\Delta x = 0.007$, Perona–Malik threshold $\lambda_{PM} = 2$, total variation fidelity parameter $\lambda_{TV} = 10$, and final time $t = 0.0008$.

The colour surface image u in this case is given by a function $u : \mathcal{S} \to \mathbb{R}^3$, where the 3 components represent RGB values. For both the total variation and Perona–Malik models, we evolve the RGB channels separately at each time step, using an average of the diffusion coefficient across the channels. This approach can also be generalized to different colour spaces, such as LUV.

6.3.3 Penalty terms in variational image processing

The gradient descent scheme on the surface involves a penalty term (called the fidelity) of the form

$$\lambda(u - u^0), \qquad \lambda \in \mathbb{R}. \tag{6.27}$$

The fidelity term ensures that the solution is not too far from the observed image u^0. This can be derived from an unconstrained minimization problem, with the assumption that the noise in the image has zero mean [CL97]. The value of the parameter λ is often chosen by the user, for example chosen to be inversely proportional to the variance σ^2 of the noise [COS01].

Applying the extension operator to the surface PDE results in an extended fidelity term of the form $\lambda(v - Ev^0) = \lambda(v - v^0)$, where we have assumed that the initial condition is given by $v^0 = Ev^0 = Eu^0$.

We now compare this to the penalty term from the closest point method constraint,

$$\gamma(v - Ev) \qquad \gamma \in \mathbb{R}, \tag{6.28}$$

which forces the solution to be a closest point extension Ev. We have shown in Chapter 3 that this constraint is enforced exactly, however note that the λ-fidelity term is only weakly enforced (otherwise the result would be equal to the initial condition).

Figure 6.6: Perona–Malik and total variation diffusion on a colour image. Original surface image (a), image with noise in the embedding space (b), result of Perona–Malik denoising (c), and result of total variation denoising (d). Parameters used are $\Delta x = 0.007$, $\lambda_{PM} = 2$, $\lambda_{TV} = 10$. Photo courtesy of Toni Lankerd.

The value of γ does not affect the solution in the continuous case, but is rather chosen to ensure consistency and stability of the numerical scheme.

The methods may be extended by allowing the parameters to be spatially varying. If the parameter λ is set to zero in some regions, this corresponds to total variation inpainting [CS01], which we will demonstrate in the following section.

6.3.4 TV inpainting

Image inpainting is a term for repairing or editing an image by filling in missing or damaged regions. It is a form of interpolation, since information in the surrounding regions is used to reconstruct data in the missing area. This was traditionally done by hand; now digital inpainting is an area of active research. It has been used in removing scratches from old photographs, removing obstructions or superimposed text, as well as image compression (for reviews see [BCHS06, CS05]). Many different techniques have been proposed, based on PDEs, variational methods, or stochastic approaches.

Suppose an image function is given with only partial data. This could be due to an obstruction such as superimposed text, or where the underlying image data has been lost. The goal is to reconstruct values for the image function in the entire region, in a way that well approximates, in some sense, the original image.

Problem 6.3.1 (Inpainting)**.** *Let Ω be an image domain, and $D \subset \Omega$ be the inpainting region. Given partial data $u_0(x) : D^c \to \mathbb{R}^m$ on $D^c = \Omega \backslash D$, reconstruct an image function $u(x) : \Omega \to \mathbb{R}^m$ on the entire domain Ω, such that $u|_{D^c} = u_0$.*

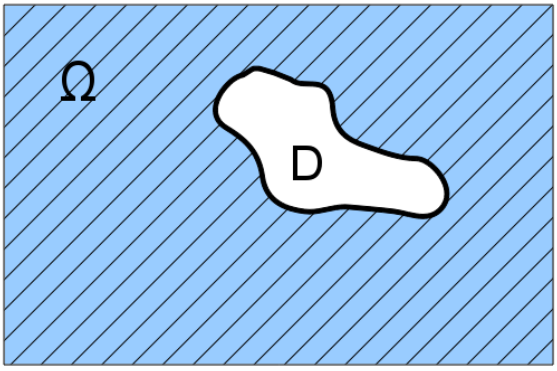


Figure 6.7: Image domain Ω and inpainting domain $D \subset \Omega$.

In 2D image inpainting, Ω is usually a rectangle, while D is a (typically user-defined) region in the shape of the scratch or superimposed text, for example. For

surface inpainting, Ω is a k-dimensional surface $\mathcal{S}$ embedded in $\mathbb{R}^n$, typically with $k = 2$ and $n = 3$.

Depending on the image model, we could relax the restriction that $u|_{D^c} = u_0$, and allow this to hold only approximately, for example if the function u_0 is assumed to be noisy and we perform simultaneous inpainting and denoising. The choice of regularizer for the function u determines the values in the region D.

Note that the term surface inpainting could also refer to reconstructing damaged regions of the surface itself (as in e.g. [CDD+04, VCBS03]). Our applications are restricted to the case of reconstructing a damaged image defined on a given fixed surface.

Here we consider TV inpainting [CS01], closely related to the TV denoising used above. To construct an energy functional $J(u)$, the regularizing term $R(u)$ is again chosen to be the total variation of the energy, as for TV denoising.

The penalty parameter $\lambda = \lambda(y)$ is now spatially varying over the surface, related to the characteristic function of the nondamaged region,

$$\lambda(y) = \begin{cases} \lambda_0 \gg 1, & y \in \Omega \backslash D \\ 0, & y \in D. \end{cases} \tag{6.29}$$

On the surface, the damaged region (where $\lambda = 0$) is also referred to as the *inpainting mask*. After extending the equation as in the TV denoising case (6.25), the resulting equation in $B(\mathcal{S})$ is

$$v_t = E\text{div}\,(a(E|\nabla v|)\nabla v) - E\lambda(v - v_0) - \gamma(v - Ev). \tag{6.30}$$

Note that we now need to extend the function λ into the embedding space with an extension operator, resulting in

$$\bar{\lambda}(x) = E\lambda = \begin{cases} \lambda_0 \gg 1, & \text{cp}(x) \in \Omega \backslash D \\ 0, & \text{cp}(x) \in D. \end{cases} \tag{6.31}$$

Care must be taken to ensure that the interpolation involved in calculating the discrete extension of the initial surface function u^0 does not result in damaged values (see Figure 6.8). A function u evaluated at $\text{cp}(x) \in \mathcal{S} \backslash D$ near the boundary ∂D may still involve some values $u(y)$ with $y \in D$. We therefore expand the mask by an additional p grid points in each dimension to avoid using values in D. Some advantages of narrow inpainting regions may be lost due to this expansion.

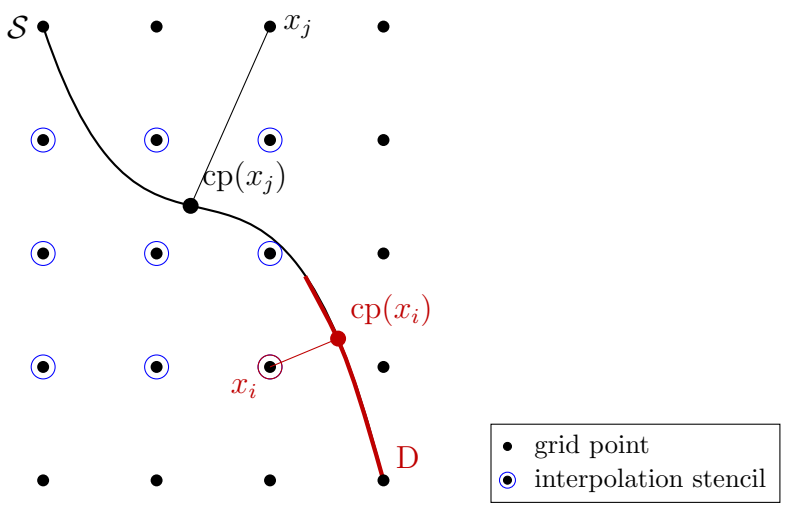


Figure 6.8: Extension of the inpainting mask: although $\mathrm{cp}(x_j)$ lies in $\mathcal{S}\backslash D$, the interpolation stencil of $\mathrm{cp}(x_j)$ contains a damaged value x_i with $\mathrm{cp}(x_i)$ in D, the inpainting region.

Examples

The numerical discretization of the TV inpainting method is very similar to that of TV denoising. We demonstrate the results of the scheme on a surface image with superimposed text. Figure 6.9 shows the original image on a fishbowl surface, followed by the damaged image, the dilated inpainting mask, and finally the result of inpainting at time $t = 0.03$. The spatial resolution is $\Delta x = 0.007$, with penalty parameters $\lambda_0 = 10^4$ and $\gamma = \frac{6}{\Delta x^2}$. A further example is shown in Figure 6.10.

6.3.5 Minimization in the embedding space

Steepest descent methods can be very slow due to nonlinearity and poor conditioning [GO09]. Fast solvers exist for energy minimization problems in image processing, for example the split Bregman method [GO09], and Chambolle's dual space method [Cha04].

Starting from a minimization problem on the surface of the form

$$\min_{u \in BV(\mathcal{S})} J(u), \tag{6.32}$$

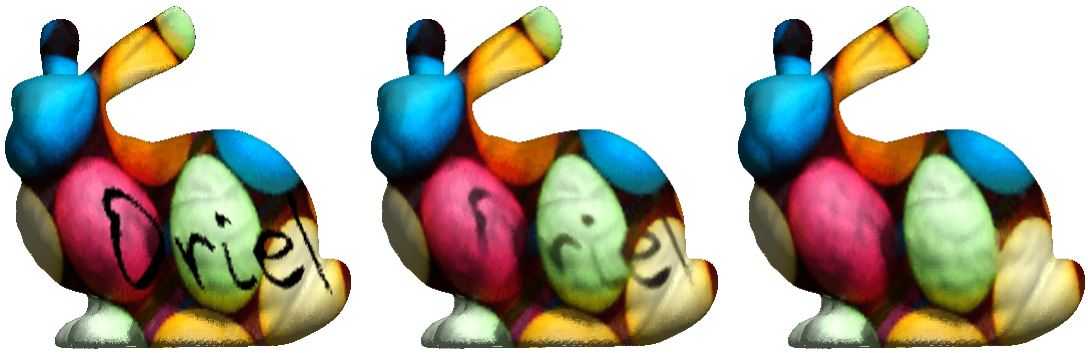
we could convert this to a modified minimization problem in the embedding space,

$$\min_{v \in BV(B(\mathcal{S}))} \tilde{J}(v), \tag{6.33a}$$

(a) (b)

(c) (d)

Figure 6.9: Total variation inpainting on a surface image. Original image texture-mapped onto a fishbowl surface (a), image damaged with superimposed text (b), inpainting mask (c), and results of TV inpainting (d). Parameters used are $\Delta x = 0.007$, $\lambda_0 = 10^4$.



(a) (b) (c)

Figure 6.10: Total variation inpainting on the Stanford bunny. Damaged colour image (a), and results of TV inpainting at time $t = 0.01$ (b) and $t = 0.03$ (c). Parameters used are $\Delta x = 0.007$, $\lambda_0 = 10^4$.

subject to a constraint that

$$v = Ev. \tag{6.33b}$$

Fast solvers could then be developed for (6.33) directly. The study of this, and the effect on the extension operators on the minimization problem, is a subject of further work.

Chapter 7

Conclusions

In this thesis, we have introduced a new formulation of a closest point method for solving partial differential equations (PDEs) on surfaces. Given a surface PDE, we construct an equation in a narrow band surrounding the surface, through a suitable extension of the surface solution. A solution of the new embedding equation, when restricted to the surface, is shown to correspond to a solution of the original surface PDE. This is achieved through the introduction of a penalty term that ensures that the solution in the embedding space remains a closest point extension. Similar to the original closest point method of Ruuth and Merriman, the method is simple and very general with respect to surface geometry, dimension and codimension.

Compared to previous approaches to constructing an implicit closest point method, this method has an advantage in that the numerical scheme can be formulated for more general PDEs, such as variable-coefficient, higher-order, and semi-linear PDEs. Because the method allows a method of lines discretization, it can be used with either implicit or explicit time-stepping (and, although not the focus here, for elliptic problems).

Numerical studies established convergence for the Laplace–Beltrami operator, and examples demonstrated the effectiveness of the method for diffusion and reaction-diffusion equations on various parameterized and triangulated surfaces. We have also investigated the use of the method in higher-order problems, for example, in the treatment of surface biharmonic problems. Stability was demonstrated through numerical examples and a study of the matrix operators.

The potential for application to models of biological and physical systems was demonstrated with several nonstandard surface diffusion problems, including diffusion based on the curvature of the surface. The closest point representation is related in a straightforward way to the mean curvature of the surface, thus leading to simple schemes for curvature-dependent problems. We have also applied the closest point

penalty method to several common image processing problems, such as denoising with nonlinear diffusion, as well as total variation inpainting.

The simplicity of the resulting numerical schemes, constructed using a straightforward closest point approach, makes the method well-suited to a broad range of applications on surfaces with possibly complex geometries.

References

[AF03] R. A. Adams and J. J. F. Fournier. *Sobolev Spaces*. Academic Press, 2003.

[AIM04] "Genus3", AIM@SHAPE Shape Repository, 2004. https://shapes.aim-at-shape.net.

[AIM05] "Fish", AIM@SHAPE Shape Repository, 2005. https://shapes.aim-at-shape.net.

[AMT+12] S. Auer, C. B. Macdonald, M. Treib, J. Schneider, and R. Westermann. Real-time fluid effects on surfaces using the Closest Point Method. *Computer Graphics Forum*, 31(6):1909–1923, 2012.

[AS96] L. Ambrosio and H. M. Soner. Level set approach to mean curvature flow in arbitrary codimension. *Journal of Differential Geometry*, 43(4):693–737, 1996.

[AV94] R. Acar and C. R. Vogel. Analysis of bounded variation penalty methods for ill-posed problems. *Inverse Problems*, 10(6):1217–1229, 1994.

[BAL07] M. Ben-Artzi and P. G. LeFloch. Well-posedness theory for geometry-compatible hyperbolic conservation laws on manifolds. *Annales de l'Institut Henri Poincare (C) Non Linear Analysis*, 24(6):989–1008, 2007.

[BCHS06] M. Bertalmío, V. Caselles, G. Haro, and G. Sapiro. PDE-based image and surface inpainting. In N. Paragios, Y. Chen, and O. D. Faugeras, editors, *Handbook of Mathematical Models in Computer Vision*, pages 33–61. Springer, 2006.

[BCOS01] M. Bertalmío, L. Cheng, S. Osher, and G. Sapiro. Variational problems and PDEs on implicit surfaces. *Journal of Computational Physics*, 174(2):759–780, 2001.

[BG88] M. Berger and B. Gostiaux. *Differential Geometry: Manifolds, Curves, and Surfaces*, volume 115 of *Graduate Texts in Mathematics*. Springer-Verlag, New York, 1988.

[Bid12] H. Biddle. Nonlinear diffusion filtering on surfaces. Master's thesis, University of Oxford, 2012.

[Bra08] J. Brandman. A level-set method for computing the eigenvalues of elliptic operators defined on compact hypersurfaces. *Journal of Scientific Computing*, 37(3):282–315, 2008.

[BT04] J.-P. Berrut and L. N. Trefethen. Barycentric Lagrange interpolation. *SIAM Review*, 46(3):501–517, 2004.

[Bur09] M. Burger. Finite element approximation of elliptic partial differential equations on implicit surfaces. *Computing and Visualization in Science*, 12(3):87–100, 2009.

[BvGMM13] H. Biddle, I. von Glehn, C. B. Macdonald, and T. März. A volume-based method for denoising on curved surfaces. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2013)*, pages 529–533, 2013.

[CDD+04] U. Clarenz, U. Diewald, G. Dziuk, M. Rumpf, and R. Rusu. A finite element method for surface restoration with smooth boundary conditions. *Computer Aided Geometric Design*, 21(5):427–445, 2004.

[CDR04] U. Clarenz, U. Diewald, and M. Rumpf. Processing textured surfaces via anisotropic geometric diffusion. *IEEE Transactions on Image Processing*, 13(2):248–261, 2004.

[Cha04] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97, 2004.

[CHL08] D. A. Calhoun, C. Helzel, and R. J. LeVeque. Logically rectangular grids and finite volume methods for PDEs in circular and spherical domains. *SIAM Review*, 50(4):723–752, 2008.

[CL97] A. Chambolle and P.-L. Lions. Image recovery via total variation minimization and related problems. *Numerische Mathematik*, 76(2):167–188, 1997.

[CM14] Y.-J. Chen and C. B. Macdonald. The Closest Point Method and multigrid solvers for elliptic equations on surfaces. 2014. Submitted.

[COS01] T. Chan, S. Osher, and J. Shen. The digital TV filter and nonlinear denoising. *IEEE Transactions on Image Processing*, 10(2):231–241, 2001.

[CS01] T. Chan and J. Shen. Mathematical models for local nontexture inpaintings. *SIAM Journal on Applied Mathematics*, 62(3):1019–1043, 2001.

[CS03] T. Chan and J. Shen. On the role of the BV image model in image restoration. *Contemporary Mathematics*, 330:25–41, 2003.

[CS05] T. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. SIAM, 2005.

[DDE05] K. Deckelnick, G. Dziuk, and C. M. Elliott. Computation of geometric partial differential equations and mean curvature flow. *Acta Numerica*, 14:139–232, 2005.

[DDEH10] K. Deckelnick, G. Dziuk, C. M. Elliott, and C. J. Heine. An h-narrow band finite-element method for elliptic equations on implicit surfaces. *IMA Journal of Numerical Analysis*, 30(2):351, 2010.

[DE07a] G. Dziuk and C. M. Elliott. Finite elements on evolving surfaces. *IMA Journal of Numerical Analysis*, 27(2):262–292, 2007.

[DE07b] G. Dziuk and C. M. Elliott. Surface finite elements for parabolic equations. *Journal of Computational Mathematics*, 25(4):385–407, 2007.

[DE08] G. Dziuk and C. M. Elliott. Eulerian finite element method for parabolic PDEs on implicit surfaces. *Interfaces and Free Boundaries*, 10(1):119–138, 2008.

[DE10] G. Dziuk and C. M. Elliott. An Eulerian approach to transport and diffusion on evolving implicit surfaces. *Computing and Visualization in Science*, 13(1):17–28, 2010.

[DE13] G. Dziuk and C. M. Elliott. Finite element methods for surface PDEs. *Acta Numerica*, 22:289–396, 2013.

[DHT14] T. A. Driscoll, N. Hale, and L. N. Trefethen, editors. *Chebfun Guide*. Pafnuty Publications, Oxford, 2014.

[DMS13] A. Dedner, P. Madhavan, and B. Stinner. Analysis of the discontinuous Galerkin method for elliptic problems on surfaces. *IMA Journal of Numerical Analysis*, 33(3):952–973, 2013.

[Dzi88] G. Dziuk. Finite elements for the Beltrami operator on arbitrary surfaces. In *Partial Differential Equations and Calculus of Variations*, volume 1357 of *Lecture Notes in Mathematics*, pages 142–155. Springer, 1988.

[EF72] E. Evans and Y. C. Fung. Improved measurements of the erythrocyte geometry. *Microvascular Research*, 4(4):335–347, 1972.

[ESV12] C. M. Elliott, B. Stinner, and C. Venkataraman. Modelling cell motility and chemotaxis with evolving surface finite elements. *Journal of the Royal Society Interface*, 9(76):3027–3044, 2012.

[FH05] M. S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. In *Advances in Multiresolution for Geometric Modelling*, pages 157–186. Springer, 2005.

[FW12] E. J. Fuselier and G. B. Wright. Scattered data interpolation on embedded submanifolds with restricted positive definite kernels: Sobolev error estimates. *SIAM Journal on Numerical Analysis*, 50(3):1753–1776, 2012.

[FW13] E. J. Fuselier and G. B. Wright. A high-order kernel method for diffusion and reaction-diffusion equations on surfaces. *Journal of Scientific Computing*, 56(3):535–565, 2013.

[GBS06] J. B. Greer, A. L. Bertozzi, and G. Sapiro. Fourth order partial differential equations on general geometries. *Journal of Computational Physics*, 216(1):216–246, 2006.

[Giu84] E. Giusti. *Minimal Surfaces and Functions of Bounded Variation*, volume 80 of *Monographs in Mathematics*. Birkhäuser Verlag, Basel, 1984.

[GO09] T. Goldstein and S. Osher. The split Bregman method for L1-regularized problems. *SIAM Journal on Imaging Sciences*, 2(2):323–343, 2009.

[Gre06] J. B. Greer. An improvement of a recent Eulerian method for solving PDEs on general geometries. *Journal of Scientific Computing*, 29(3):321–352, 2006.

[GS83] P. Gray and S. K. Scott. Autocatalytic reactions in the isothermal, continuous stirred tank reactor: isolas and other forms of multistability. *Chemical Engineering Science*, 38(1):29–43, 1983.

[GS03] H. Grossauer and O. Scherzer. Using the complex Ginzburg-Landau equation for digital inpainting in 2D and 3D. In *Scale Space Methods in Computer Vision*, pages 225–236. Springer, 2003.

[GT01] D. Gilbarg and N. S. Trudinger. *Elliptic Partial Differential Equations of Second Order*. Springer, 2001.

[GWC+04] X. Gu, Y. Wang, T. Chan, P. M. Thompson, and S.-T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(8):949–958, 2004.

[Hir76] M. W. Hirsch. *Differential Topology*, volume 33 of *Graduate Texts in Mathematics*. Springer, 1976.

[HNW93] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I: Nonstiff Problems*. Springer, 1993.

[HZQW10] Y. Hong, D. Zhu, X. Qiu, and Z. Wang. Geometry-based control of fire simulation. *The Visual Computer*, 26(9):1217–1228, 2010.

[Kim97] R. Kimmel. Intrinsic scale space for images on surfaces: the geodesic curvature flow. In *Scale-Space Theory in Computer Vision*, pages 212–223. Springer, 1997.

[KTT13] T. Kim, J. Tessendorf, and N. Thuerey. Closest point turbulence for liquid surfaces. *ACM Transactions on Graphics (TOG)*, 32(2):15, 2013.

[LC11] R. Lai and T. Chan. A framework for intrinsic image processing on surfaces. *Computer Vision and Image Understanding*, 115(12):1647–1661, 2011.

[Lee09] J. M. Lee. *Manifolds and Differential Geometry*, volume 107 of *Graduate Studies in Mathematics*. AMS, 2009.

[LeV07] R. J. LeVeque. *Finite Difference Methods for Ordinary and Partial Differential Equations*. SIAM, 2007.

[LLZ11] S. Leung, J. Lowengrub, and H. Zhao. A grid based particle method for solving partial differential equations on evolving surfaces and modeling high order geometrical motion. *Journal of Computational Physics*, 230(7):2540–2561, 2011.

[LR01] R. J. LeVeque and J. A. Rossmanith. A wave propagation algorithm for the solution of PDEs on the surface of a sphere. In *Hyperbolic Problems: Theory, Numerics, Applications*, volume 141 of *ISNM International Series of Numerical Mathematics*, pages 643–652. Birkhäuser Basel, 2001.

[LXZ08] D. Liu, G. Xu, and Q. Zhang. A discrete scheme of Laplace–Beltrami operator and its convergence over quadrilateral meshes. *Computers & Mathematics with Applications*, 55(6):1081–1093, 2008.

[LZ09] S. Leung and H. Zhao. A grid based particle method for moving interface problems. *Journal of Computational Physics*, 228(8):2993–3024, 2009.

[LZ13] J. Liang and H. Zhao. Solving partial differential equations on point clouds. *SIAM Journal on Scientific Computing*, 35(3):A1461–A1486, 2013.

[Mac08] C. B. Macdonald. *The Closest Point Method for time-dependent processes on surfaces*. PhD thesis, Simon Fraser University, 2008.

[MBR11] C. B. Macdonald, J. Brandman, and S. J. Ruuth. Solving eigenvalue problems on curved surfaces using the Closest Point Method. *Journal of Computational Physics*, 230(22):7944–7956, 2011.

[MCC02] T. G. Myers, J. P. F. Charpin, and S. J. Chapman. The flow and solidification of a thin fluid film on an arbitrary three-dimensional surface. *Physics of Fluids*, 14(8):2788–2803, 2002.

[MM05] K. W. Morton and D. F. Mayers. *Numerical Solution of Partial Differential Equations*. Cambridge University Press, 2005.

[MM12] T. März and C. B. Macdonald. Calculus on surfaces with general closest point functions. *SIAM Journal on Numerical Analysis*, 50(6):3303–3328, 2012.

[MMC14] T. März, C. B. Macdonald, and Y.-J. Chen. Consistency and stability of closest point iterations. In preparation, 2014.

[MMR13] C. B. Macdonald, B. Merriman, and S. J. Ruuth. Simple computation of reaction-diffusion processes on point clouds. *Proceedings of the National Academy of Sciences*, 110(23):9209–9214, 2013.

[MR08] C. B. Macdonald and S. J. Ruuth. Level set equations on surfaces via the Closest Point Method. *Journal of Scientific Computing*, 35(2–3):219–240, 2008.

[MR09] C. B. Macdonald and S. J. Ruuth. The implicit Closest Point Method for the numerical solution of partial differential equations on surfaces. *SIAM Journal on Scientific Computing*, 31(6):4330–4350, 2009.

[Mun96] R. Munafo. Reaction-Diffusion by the Gray-Scott Model: Pearson's Parameterization. http://www.mrob.com/pub/comp/xmorphia, 1996. Accessed: May 2014.

[Mur03] J. D. Murray. *Mathematical Biology II: Spatial Models and Biomedical Applications*, volume 18 of *Interdisciplinary Applied Mathematics*. Springer, 2003.

[OF03] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer, 2003.

[OR10] M. A. Olshanskii and A. Reusken. A finite element method for surface PDEs: matrix properties. *Numerische Mathematik*, 114(3):491–520, 2010.

[ORG09] M. A. Olshanskii, A. Reusken, and J. Grande. A finite element method for elliptic equations on surfaces. *SIAM Journal on Numerical Analysis*, 47(5):3339–3358, 2009.

[OS88] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *Journal of Computational Physics*, 79(1):12–49, 1988.

[Pea93] J. E. Pearson. Complex patterns in a simple system. *Science*, 261(5118):189–192, 1993.

[Pir12] C. Piret. The orthogonal gradients method: A radial basis functions method for solving partial differential equations on arbitrary surfaces. *Journal of Computational Physics*, 231(14):4662–4675, 2012.

[PM90] P. Perona and J. Malik. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.

[RM08] S. J. Ruuth and B. Merriman. A simple embedding method for solving partial differential equations on surfaces. *Journal of Computational Physics*, 227(3):1943–1961, 2008.

[RO94] L. I. Rudin and S. Osher. Total variation based image restoration with free local constraints. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 94)*, pages 31–35, 1994.

[ROF92] L. I. Rudin, S. Osher, and E. Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena*, 60(1-4):259–268, 1992.

[Ruu95] S. J. Ruuth. Implicit-explicit methods for reaction-diffusion problems in pattern formation. *Journal of Mathematical Biology*, 34(2):148–176, 1995.

[Set96] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[Sim84] L. Simon. *Lectures on Geometric Measure Theory*. Centre for Mathematical Analysis, Australian National University, 1984.

[SK05] A. Spira and R. Kimmel. Enhancing images painted on manifolds. In *Scale Space and PDE Methods in Computer Vision*, pages 492–502. Springer, 2005.

[SWKF14] V. Shankar, G. B. Wright, R. M. Kirby, and A. L. Fogelson. A radial basis function (RBF) finite difference method for the simulation

of reaction-diffusion equations on stationary platelets within the augmented forcing method. *International Journal for Numerical Methods in Fluids*, 75(1):1–22, 2014.

[TMR09] L. Tian, C. B. Macdonald, and S. J. Ruuth. Segmentation on surfaces with the Closest Point Method. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 09)*, pages 3009–3012, 2009.

[Tur91] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *SIGGRAPH Computuer Graphics*, 25(4):289–298, 1991.

[VCBS03] J. Verdera, V. Caselles, M. Bertalmío, and G. Sapiro. Inpainting surface holes. In *Proceedings of the IEEE International Conference on Image Processing (ICIP 2003)*, volume 2, pages 903–906, 2003.

[vGMM14] I. von Glehn, T. März, and C. B. Macdonald. An embedded method-of-lines approach to solving partial differential equations on surfaces. 2014. Submitted.

[Wei98] J. Weickert. *Anisotropic Diffusion in Image Processing*. Teubner, Stuttgart, 1998.

[XZ03] J.-J. Xu and H. Zhao. An Eulerian formulation for solving partial differential equations along a moving interface. *Journal of Scientific Computing*, 19(1-3):573–594, 2003.