An improvement of a recent Eulerian method for solving PDEs on general geometries*

John B. Greer †

Abstract

We improve upon a method introduced in (Bertalmio et. al. JCP 2001) for solving evolution PDEs on codimension-one surfaces in $\mathbb{R}^N$. As in the original method, by representing the surface as a level set of a smooth function, we use only finite differences on a Cartesian mesh to solve an Eulerian representation of the surface PDE in a neighborhood of the surface. We modify the original method by changing the Eulerian representation to include effects due to surface curvature. This modified PDE has the very useful property that any solution which is initially constant perpendicular to the surface remains so at later times. The change remedies many of problems facing the original method, including a need to frequently extend data off of the surface, uncertain boundary conditions, and terribly degenerate parabolic PDEs. We present numerical examples that include convergence tests in neighborhoods of the surface that shrink with the grid size.

PDEs on surfaces arise in areas including geometry [17, 22], biology [16, 22, 31], fluid dynamics [23, 24], and computer graphics [8, 10, 32, 34]. Until recently, PDEs were computed on surfaces using either surface parametrization or triangulation, both of which have drawbacks. Parametrizations may be difficult to derive for complicated surfaces, and they often require patching to connect different parametrization neighborhoods. Of course a new parametrization must be derived for each different surface. Triangulated surfaces do not have well-defined geometric properties; surface normals are not clearly defined, and curvature quantities are even more ambiguous. Convergence of numerical schemes on triangulated surfaces remains less understood than convergence of numerical schemes on Cartesian grids.

Noting the success of level set methods for dynamically evolving surfaces, Bertalmio et. al. [4] introduced an Eulerian method for solving PDEs on general smooth geometries using only finite differences on a Cartesian mesh. By representing the surface S as the zero level set of a function ϕ defined in $\mathbb{R}^N$, we may rewrite the PDE in Eulerian coordinates and solve this new representation on a Cartesian grid in a neighborhood of S. The Eulerian representation is derived by replacing surface derivatives with projections of derivatives in the embedding Euclidean space. This approach avoids many of the difficulties of parameterized or triangulated surfaces. Implicit representations of surfaces may be computed using one of many different algorithms [13, 18, 19, 36]. Surface properties such as its normal and mean curvature are easily calculated using derivatives of ϕ. PDE computations are simplified, since we use standard finite difference schemes in the embedding space. Finally, the method is more easily adapted to problems involving moving interfaces [2, 35]; for example, we might solve a PDE on a surface that simultaneously evolves.

Results in past works demonstrated applications of this Eulerian method to a wide range of surfaces and PDEs. The authors of [4] applied it to linear diffusion, anisotropic diffusion, and

*Work supported by the National Science Foundation.

†Courant Institute for Mathematical Sciences, New York University, New York, 10012, greer@cims.nyu.edu

reaction diffusion equations on surfaces of varying complexity. The authors of [2] and [35] took advantage of the level set representation to solve PDEs on dynamic interfaces, thus demonstrating the full power of the method. More recently, in [15] fourth order equations were solved on surfaces using implicit representations of the surfaces. Although the method proposed in [4] has been applied mainly to evolution equations, Memoli and Sapiro used the same framework to solve the Eikonal equation on surfaces in [21].

Despite its wide applicability, the method as used in the above papers has a number of drawbacks, especially in the case of diffusion equations on surfaces. The Eulerian representation of the surface PDE is defined on a larger domain in one higher dimension. To reduce the amount of additional work an extra dimension requires, one typically works in a small band around the surface. Computing in this domain requires choosing conditions at the band's boundary, though the correct boundary conditions remain unclear, since the analytic solution is unaffected by boundary conditions. Dirichlet boundary conditions result in a jump in the computed solution at the boundary and lead to large errors at later times [15, 35]. Homogeneous Neumann Boundary conditions lead to a jump at the boundary in the gradient of the computed solution.

In order to solve the PDE in a neighborhood of the surface, initial data must be extended off of the surface to the larger domain. Since the finite difference schemes used will be strongly affected by values of the solution at grid points off of the surface, this data extension must be done carefully. The authors of the above papers extended the surface data by requiring the initial condition to be constant normal to the surface, thus minimizing variations in the solution off of the zero level set and hopefully limiting errors due to discretization. Unfortunately, the solution of the Eulerian PDE does not retain this property at later times, and as a result, long-time calculations of the PDEs often require intermittent re-extension of the data off of the surface [15, 35]. This might be needed to offset effects of boundary conditions [15] or to minimize the effects of discretizations that include points from different level sets of ϕ. The re-extension is analogous to re-extension of velocities off of a moving interface in level set methods [25, 28], and as is the case for level set methods, the frequency and method of re-extension is often ad-hoc.

Applying the method in [2, 4, 35] to diffusion equations on surfaces results in a degenerate diffusion equation in the embedding space, as there is no diffusion in the direction perpendicular to the surface. This occurs with equations as innocuous as the heat equation, since the degeneracy is caused by the embedding method, as opposed to the surface PDE. This degeneracy can have a terrible effect on discretizations. In particular, little is known about the convergence of schemes for degenerate diffusion equations. The authors of [15] noted unusually severe degeneracy when solving fourth order PDEs on surfaces.

We introduce a modification of the method in [2, 4, 35] that is no more difficult to implement than the original method, requires no re-extension of the surface data, has natural boundary conditions associated with the Eulerian representation, and results in diffusion equations that are non-degenerate. We accomplish this by changing the PDE solved in the neighborhood of S. At points on the surface, the PDE we solve is identical to that of the original method, but our method differs at points off the surface. Our Eulerian formulation of the surface PDE satisfies the following property: given initial data that is constant in the direction perpendicular to the surface, the solution retains this property at later times.

We first discuss the original method and its drawbacks, focusing on degeneracy problems, the need to re-extend surface data, and the lack of natural boundary conditions. These disadvantages are clearly illustrated by applying the method to the heat equation on the unit circle, which we examine in Section 3. This example motivates our modification to the original method, which we fully justify in Section 5. In sections 6 and 7, we discuss numerical examples including advection

and diffusion equations on curves in $\mathbb{R}^2$ and surfaces in $\mathbb{R}^3$. We include convergence tests done in a band around each surface that shrinks with the grid size. As far as we know, the convergence tests in [2] and [35] are not done in domains that shrink with the grid size, and it is unclear how accurate the original method is in that situation. Obviously, computing in a narrow band is most advantageous when we can shrink the width of the band with the grid cell width.

1 Eulerian Representations of PDEs defined on surfaces.

We first describe the method developed in [2, 4, 35]. Let S be a closed embedded codimension-one manifold in $\mathbb{R}^N$. Let $\phi : \mathbb{R}^N \to \mathbb{R}$ be the signed distance function to S with the convention $\phi < 0$ inside S and $\phi > 0$ outside S. Means of producing such an implicit representation of S (e.g. via Fast Marching Method) may be found in [7, 25, 28]. Many geometric properties of S and the other level sets of ϕ have elegant expressions in terms of the derivatives of ϕ. The outer normal to each level set is given by $\nabla\phi$, and each level set's curvature tensor in Eulerian coordinates may be calculated by the formula

$$\mathrm{K} = -D^2\phi, \tag{1}$$

where $D^2\phi$ is the Hessian of ϕ. Local projections onto tangent planes to each level set are given by

$$\mathrm{P} = \mathrm{I} - \nabla\phi \otimes \nabla\phi, \tag{2}$$

where I denotes the identity matrix. So that we may compute PDEs on S, we will use P to embed surface derivatives in $\mathbb{R}^N$. Our method differs from that of [4] in the choice of embedding.

1.1 Embedding surface derivatives – the method of [4].

Suppose $\tilde{f}$ is a smooth function defined in an open domain containing S and let f be the restriction of $\tilde{f}$ to S. We use $\nabla_S f$ to denote the gradient of f intrinsic to the surface S, and we use $\nabla\tilde{f}$ to denote the standard Euclidean gradient of $\tilde{f}$. Since surface derivatives of f are given by projections of Euclidean derivatives of $\tilde{f}$ onto the tangent space of S [9, 29], we can define $\nabla_S f$ using P and $\nabla\tilde{f}$:

$$\nabla_S f = \mathrm{P}(x)\nabla\tilde{f}. \tag{3}$$

Since ϕ and P are defined in a domain $\Omega \subset \mathbb{R}^3$ containing S, (3) makes sense in all of Ω. At each $x \in \Omega$, $\nabla_S f$ defined as above corresponds to the gradient of f intrinsic to the level set of ϕ through x. Once ∇_S is established, we may use it to calculate higher derivatives; for example, the Laplace-Beltrami operator is given by

$$\Delta_S f = \nabla_S \cdot \nabla_S u = (\mathrm{P}\nabla)\cdot\left(\mathrm{P}\nabla\tilde{f}\right) = \nabla\cdot\left(\mathrm{P}\nabla\tilde{f}\right). \tag{4}$$

From here on, we ignore use of the tilde, always assuming that we differentiate quantities defined not only on the surface, but in the embedding space as well. We discuss how to extend surface data to the embedding space in Section 2.

We may now use (3) to derive expressions of surface PDEs that depend only on P and Euclidean derivatives. Consider the heat equation on a surface S:

$$u_t = \Delta_S u \tag{5}$$

combined with the initial condition

$$u(y,0) = f(y) \text{ for } y \text{ on } S. \tag{6}$$

Assuming ϕ is a signed distance function, we use (4) to write (5) in terms of Euclidean derivatives:

$$u_t = \nabla \cdot (\mathrm{P}\, \nabla u)\,. \tag{7}$$

Since equation (7) is defined in an open subset of $\mathbb{R}^N$, we can compute it on a Cartesian grid. We refer to (7) as an Eulerian representation of the surface PDE (5). Note that solving (7) in $\mathbb{R}^N$ is equivalent to solving (5) on each level set of ϕ. As remarked in [4], a wide variety of PDEs defined on surfaces may be similarly represented as PDEs that depend on appropriate projections of derivatives in the embedding space. Examples showcasing the generality of this method may be found in [2, 4, 15, 21, 35].

1.2 Embedding surface derivatives – the modification.

Equation (7) is an obvious embedding of (5), but it is not the only one. Since we only need to solve exactly (7) at points on the surface, there is some leniency in choosing the PDE to solve at points off of the surface. Our contribution is a change of the above method that relies on our freedom to choose a "useful" PDE to solve at points off of the zero level set of ϕ. In the PDEs derived by the methods of [4], we recommend replacing the projection matrix P by

$$\tilde{\mathrm{P}} = (\mathrm{I} - \phi \mathrm{K})^{-1}\, \mathrm{P}. \tag{8}$$

To solve the heat equation on a surface, for example, we propose solving

$$u_t = \left(\tilde{\mathrm{P}}\nabla\right) \cdot \left(\tilde{\mathrm{P}}\nabla u\right) \tag{9}$$

instead of (7). This modification relies on two assumptions. First, ϕ must be a signed distance function. This differs from the original method, which has no such requirement, though the authors of [4] and [15] pointed out that distance functions are preferred. Second, we may only compute in a domain that is contained within a band around the surface with a width satisfying

$$|\phi(x)| \leq c < \frac{1}{\kappa_{\max}}, \tag{10}$$

where $\kappa_{\max}$ is the maximum achieved on the surface by the magnitude of any of the surface's principal curvatures. This restriction on the domain is also required for the methods in [2, 4, 15, 35], since it ensures that $\nabla\phi$ is well defined (when ϕ is a signed distance function). Assumption (10) ensures that $(\mathrm{I} - \phi\mathrm{K})$ is invertible. The inversion itself may be done easily as a preprocessing step, just as computing P is done.

Before discussing the advantages of the modification, we provide a brief overview of the technical details involved in computing Eulerian representations of PDEs on surfaces. In Section 3, we expound the reasons for replacing P with $\tilde{\mathrm{P}}$; in particular, we demonstrate the shortcomings of the original embedding that the modified embedding remedies.

2 Computing Eulerian representations of PDEs on surfaces.

Our discussion in Section 1 ignored technical issues that arise in the numerical implementation of any Eulerian representation of a surface PDE. We must establish the computational domain in the embedding space, and since the original PDE and initial condition are defined only on the surface, we must specify how to extend the initial data to this domain. Although the surface is assumed to have no boundary, the new computational domain in the embedding space will necessarily have a

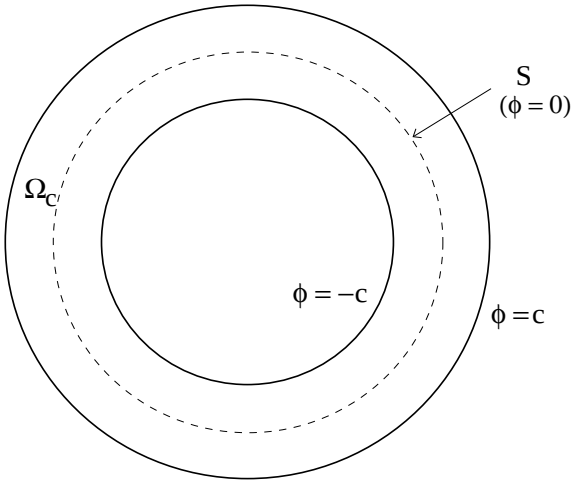

Figure 1: A typical computational domain.

boundary, and boundary conditions must be prescribed for numerical solution. We briefly discuss these issues, referring to earlier works [2, 4, 15, 35] for elaboration, and emphasize that they are not unique to our modification, but are intrinsic to any similar Eulerian representation of surface PDEs.

2.1 Computational domain.

Although we have added a dimension to the computational domain by solving a PDE defined on the space containing S, we limit the added cost by computing only in a small band around the surface. Choosing $c > 0$, we compute in the tubular neighborhood

$$\Omega_c = \{x : |\phi(x)| \leq c\}. \tag{11}$$

The choice of c generally depends on the size of the numerical stencil and on the complexity of the working surface. This restriction to a band is similar in spirit to the local level set method [1, 26]. When ϕ is a signed distance function, c must satisfy the upper bound (10) so that $\nabla\phi$ is properly defined throughout Ω_c.

2.2 Extension of initial data.

Since the initial data, u_0, is assumed to be defined only on the surface S, we must extend it to the other level sets of ϕ within Ω_c. We do so by requiring

$$\nabla u_0 \cdot \nabla\phi = 0 \quad \text{for} \quad x \in \Omega_c. \tag{12}$$

In other words, we require u_0 to be constant perpendicular to the surface. Such an extension can be easily computed by solving (12) directly using a fast marching method [28], or by solving

$$(u_0)_t + sign(\phi)\,\nabla u_0 \cdot \nabla\phi = 0 \tag{13}$$

to steady state [25]. Note that restricting the computational domain to satisfy (10) ensures well-posedness of (12).

2.3 Boundary conditions.

Consider equation (7) defined in Ω_c, which has a boundary given by level sets of ϕ. Since no information is transferred between level sets of ϕ, boundary conditions are not required to solve the PDE analytically within Ω_c. This no longer remains true when using finite differences to solve (7) on a Cartesian grid in Ω_c. Now boundary conditions are important – they are necessary for computation and certainly affect the solution. Computations in past works used either Dirichlet or homogeneous Neumann boundary conditions [2, 15, 35].

3 Contrasting the original and modified formulations.

We next discuss the major challenges facing the original method, many of which are discussed in [15] in greater detail. These challenges will be resolved by our modified method. Past results demonstrate the success of using an Eulerian formulation of surface PDEs for computation, but they also demonstrate areas needing improvement. Intermittent re-extension of surface data is needed for long time calculations, proper boundary conditions are not known, and well-behaved surface PDEs may have an Eulerian formulation that is poorly behaved. In this section we explore the causes of these drawbacks by applying the method to the heat equation on the unit circle. The same example demonstrates the success of our recommended modification of the method.

3.1 Heat equation on the unit circle – the original method.

Parameterizing the unit circle by arc-length s, the heat equation is

$$\begin{aligned} u_t &= u_{ss} \qquad \text{in} \quad [0,2\pi)\times[0,\infty) \\ u(s,0) &= u_0(s) \end{aligned} \tag{14}$$

with periodic boundary conditions in s. The signed distance function for the unit circle is of course

$$\phi(x,y) = \sqrt{x^2+y^2} - 1.$$

We may select our computational domain to be the annulus

$$\left\{(x,y) : \left|\sqrt{x^2+y^2} - 1\right| \le c\right\} \tag{15}$$

for any positive $c < 1$. We extend the initial data to this embedding domain by noting that

$$u_0(s) = u_0(\theta) \tag{16}$$

in polar coordinates (r,θ), and requiring (16) to hold in the entire annulus. Using the original method of [4], the Eulerian representation of (14) involves solving (7) in the domain (15) with initial condition (16).

Next write the Eulerian formulation (7) in polar coordinates $u(r,\theta,t)$:

$$u_t = \frac{1}{r^2}u_{\theta\theta} \qquad \text{in} \quad \{(r,\theta) : |r-1| \le c\}\times[0,\infty). \tag{17}$$

We immediately see from (17) that although the initial data depends only on θ, the solution will depend on r at all later times; for example, letting $u_0 = \sin(n\theta)$, the solution of (17) is

$$u(r,\theta,t) = e^{-\left(\frac{n}{r}\right)^2 t}\sin(n\theta).$$

Fixing Dirichlet conditions on the band's boundaries results in a discontinuity in u at the boundary at later times, whereas fixing $\frac{\partial}{\partial r}u = 0$ at the boundary results in a discontinuity in u_r at the boundary. While we may select appropriate boundary conditions for this particular problem, our selection will not easily generalize to other PDEs and surfaces. Also note that the coefficient of diffusion is largest for level sets away from the surface of interest (the circles of smaller radius). Diffusion on these level sets will have a detrimental effect on any explicit schemes, since time steps will be limited by the level sets in the band with highest curvature.

3.2 Heat equation on the unit circle – the modified method.

The drawbacks of solving (17) result from the original method's embedding of surface derivatives. When S is the unit circle, writing the embedding in polar coordinates gives

$$\nabla_S u \;=\; \mathrm{P}\,\nabla u = \frac{1}{r}u_\theta.$$

In addition to expressing the metric on S, the zero level set of ϕ, the embedding establishes a metric on each of the nonzero level sets of ϕ. In fact, the established metric on each level set is the standard metric induced by $\mathbb{R}^N$. This results in slower diffusion on outer circles than on inner ones, rapidly producing changes in the values of u in the radial direction.

A better embedding of the surface derivatives would define a more useful metric on the nonzero level sets of ϕ. Only the zero level set should have a metric induced by the Euclidean metric in $\mathbb{R}^N$; any other level set should inherit its metric from the zero level set of ϕ. This is precisely what our modified matrix $\tilde{\mathrm{P}}$ does. For the unit circle, the modified embedding of the surface gradient is given by

$$\tilde{\mathrm{P}}\,\nabla u = u_\theta.$$

Thus the embedded derivative depends on arc-length only on the unit circle. For general surfaces, the metric prescribed by $\tilde{\mathrm{P}}$ to each level set is inherited from distances on the zero level set of ϕ.

Written in polar coordinates, our modified Eulerian representation of the heat equation on the unit circle is

$$u_t = u_{\theta\theta} \qquad \text{in} \quad \{(r,\theta) : |r-1| \le c\} \times [0,\infty). \tag{18}$$

If the initial condition depends only on θ, then the solution of (18) clearly depends only on θ at all later times. In computations, homogeneous Neumann Boundary conditions are then appropriate, since $u_r = 0$ throughout the domain.

3.3 Advantages of the new formulation.

In general, by replacing P with $\tilde{\mathrm{P}}$ in the Eulerian formulation of each surface PDE, the solutions have the property that given initial data u_0 satisfying (12), the solution u satisfies

$$\nabla u \cdot \nabla \phi = 0 \tag{19}$$

at all later times. This immediately remedies two of the original method's major drawbacks. The function u remains constant normal to the level sets of ϕ, thus removing any need for ad-hoc re-extension methods. Eliminating re-extensions reduces the amount of computational work needed and prevents accumulating errors due to changes in u on the surface caused by the re-extension method.

Since u satisfies (19) at all times, Neumann Boundary conditions are the obvious conditions to enforce at the boundaries of the band. In practice, we recommend an even simpler technique

for computing near the boundary of the band. Instead of fixing boundary conditions, we assume u satisfies (19) in a neighborhood of the band. Whenever the numerical scheme's stencil calls for a grid point outside the band, we use (19) to prescribe the value of u at that grid point. This is different from (and easier than) applying Neumann Boundary conditions, since we ignore the exact location of the boundary. The extrapolation is advantageous since the band boundaries are typically curved and difficult to locate exactly.

The modified method we present here has a third advantage when computing diffusion equations on surfaces. Since u is constant in the direction normal to level sets of ϕ, we may add diffusion in that direction without affecting the solution within the band. By doing this when solving diffusion equations on surfaces, we derive a non-degenerate diffusion equation. We discuss this further in Section 5.3.

4 Outline of the modified method.

We briefly outline our modified method, noting the changes from the original method used in [2, 4, 15, 35].

1. Compute ϕ, the signed distance function to the surface. The original method does not require that ϕ be a signed distance function, though the authors of [4] discussed advantages of using signed distance functions.

2. Choose the computational domain. This should be a band around the surface that is small enough to satisfy (10), which ensures that $\nabla\phi$ is well defined. This step is the same in the original method.

3. Derive the Eulerian representation of the surface PDE as in [4], but replace the projection matrix P with the modified projection matrix $\tilde{\mathrm{P}}$ defined by (8).

4. Extend the initial data off of the surface by requiring (12). Our restriction on the band's width ensures that (12) is well posed. This step is the same in the original method.

5. For diffusion equations on the surface, add diffusion in the direction normal to the surface. This ensures that the embedded equation is non-degenerate. The amount of diffusion added is optional, and should be chosen for convenience; examples are discussed in sections 5, 6, and 7. Adding this diffusion to the original method results in an equation that is inconsistent with the surface PDE, but we may add it to the modified method, since u remains constant normal to the surface. One may think of the added diffusion as a means of damping any numerical errors in this direction.

6. Compute the Eulerian representation of the surface PDE using standard finite differences on a Cartesian mesh in the computational domain. Use (19) to extrapolate the solution to grid points outside of the band that the scheme may call for. Boundary conditions are more troublesome for the original method, in which (19) is not satisfied at all times.

The remainder of this paper consists of two main parts. In Section 5 we give simple, but rigorous, arguments supporting our formal discussion in Section 3. Readers who are more interested in applications may prefer skipping to sections 6 and 7, where we provide numerical examples in both two and three dimensions.

5 Analysis of the modified embedding.

This section is devoted to showing that unlike $P(x)\nabla$, $\tilde{\mathrm{P}}(x)\nabla$ is an operator that does not depend on the distance of x from the surface. As a result, by using $\tilde{\mathrm{P}}\nabla$ to embed ∇_S, we derive an Eulerian representation of each surface PDE that depends only on surface coordinates and not on the distance orthogonal to the surface. In other words, the solutions of our Eulerian representation of dynamic surface PDEs have the following useful property: If the initial condition $u(x,0) = u_0(x)$ satisfies

$$\nabla\phi \cdot \nabla u_0 = 0, \tag{20}$$

then u satisfies

$$\nabla\phi \cdot \nabla u(t) = 0 \tag{21}$$

for all $t \geq 0$.

For convenience of notation, we use $\tilde{D}u$ to denote our modified embedding of the gradient, that is

$$\tilde{D}u := \tilde{\mathrm{P}}\nabla u.$$

We similarly use $\tilde{D}^2 u$ to denote the embedding of the Hessian,

$$\tilde{D}^2 u := \left(\tilde{\mathrm{P}}\nabla\right)\tilde{D}u.$$

We prove the following theorem.

Theorem 1. *Let $F(x,r,p,X)$ be a C^1 mapping*

$$F : \Omega_c \times \mathbb{R} \times \mathbb{R}^N \times S(N) \to \mathbb{R},$$

where $S(N)$ denotes symmetric $N \times N$ matrices. Suppose $u(x,t)$ is a $C^1\left([0,\infty), C^2\left(\Omega_c\right)\right)$ solution of

$$u_t = F(x,u,\tilde{D}u,\tilde{D}^2 u) \tag{22a}$$

$$u(x,0) = u_0(x). \tag{22b}$$

Also suppose u_0 satisfies (20) *and*

$$\nabla_x F \cdot \nabla\phi = 0. \tag{23}$$

Then u satisfies (21) *for all $t \geq 0$.*

Remark. Theorem 1 is obviously not as general as it could be. We may easily extend it to higher order PDEs, for example. For simplicity however, we prove only Theorem 1, which applies to all the numerical examples in sections 6 and 7.

5.1 Notation.

We remind the reader that ϕ is assumed to be a signed distance function to a closed embedded codimension-one manifold S in $\mathbb{R}^N$. We again use Ω_c to denote the tubular neighborhood of S given by (11). To ensure that the distance function ϕ is smooth within Ω_c, we require c to satisfy (10). Given a point x in Ω_c, let β be the signed distance of x from the closest point x_0 on the zero level set of ϕ. Assuming

$$x_0 = \alpha := (\alpha_1, ..., \alpha_{N-1})$$

in some local coordinate system of the zero level set, we use the coordinates $x = (\alpha_1, ..., \alpha_{N-1}, \beta)$ within Ω_c. Letting $\Gamma_0(\alpha)$ denote the zero level set of ϕ, this coordinate system for Ω_c is given by

$$x(\alpha, \beta) = \Gamma_0(\alpha) + \beta\nu(\alpha), \tag{24}$$

where $\nu(\alpha)$ is the outward pointing unit normal to Γ_0 at $\Gamma_0(\alpha)$ (note that $\nu(\alpha) = \nabla\phi(\Gamma_0(\alpha))$).

We define

$$\mathrm{K}(x) = -D^2\phi(x),$$

so that K gives the curvature tensor of each level set in Eulerian coordinates. Using the coordinate system (24), we define

$$\mathrm{K}_0(\alpha_1, ..., \alpha_{N-1}, \beta) = \mathrm{K}(\alpha_1, ..., \alpha_{N-1}, 0). \tag{25}$$

In other words, at any point $x \in \Omega_c$ $\mathrm{K}_0(x)$ gives the curvature tensor of Γ_0 at its closest point to x.

Our modification to P makes use of a well known relationship between K and K_0 :

$$\mathrm{K}_0 = \mathrm{K}(\mathrm{I} - \phi\mathrm{K}). \tag{26}$$

See exercises 6 of Section 1-7 and 11 of Section 3-5 in [9] for examples of (26). Evans and Spruck used the same property to analyze the PDE satisfied by the signed distance function to a surface moving by mean curvature [12]. Equation (26) implies that

$$(\mathrm{I} + \beta\mathrm{K})^{-1} \;=\; \mathrm{I} - \beta\mathrm{K}_0 \;=\; \mathrm{K}_0\mathrm{K}^{-1},$$

thus the modified projection matrix,

$$\tilde{\mathrm{P}} = \mathrm{K}_0\mathrm{K}^{-1}\mathrm{P}, \tag{27}$$

accounts for the difference in curvature between the level set of ϕ through x and the zero level set of ϕ. In our analysis of its properties, we write $\tilde{\mathrm{P}}$ as

$$\tilde{\mathrm{P}} = (\mathrm{I} - \beta\mathrm{K}_0)\,\mathrm{P}, \tag{28}$$

but in computations, it is typically more convenient to calculate $\tilde{\mathrm{P}}$ by (8), since

$$\mathrm{K} = -D^2\phi.$$

5.2 Proof of Theorem 1.

Once we prove that the modified embedding of surface derivatives does not depend on distances from the surface, the theorem easily follows.

Lemma 1. *Suppose u satisfies* (12) *and* $\mathbf{w}$ *is a* C^1 *vector field satisfying*

$$\frac{\partial}{\partial\beta}\mathbf{w} = 0. \tag{29}$$

Then

$$\frac{\partial}{\partial\beta}(\tilde{D}u \cdot \mathbf{w}) = 0.$$

In particular,

$$\frac{\partial}{\partial\beta}\tilde{D}u = 0.$$

Proof. Since $\nabla(\nabla u \cdot \nabla \phi) = 0$,

$$D^2 u \nabla \phi = -D^2 \phi \nabla u = K \nabla u. \tag{30}$$

Using (26), (29), and (30), we calculate

$$\begin{aligned}
\partial_\beta(\tilde{P}\nabla u \cdot w) &= (\tilde{P}\nabla u)_\beta \cdot w \\
&= \left(\left(\partial_\beta \tilde{P}\right)\nabla u + \tilde{P}(\partial_\beta \nabla u)\right) \cdot w \\
&= \left(-\mathrm{K}_0 P \nabla u + (1-\beta \mathrm{K}_0) P D^2 u \nabla \phi\right) \cdot w \\
&= \left(-\mathrm{K}_0 P \nabla u - (1-\beta \mathrm{K}_0) P D^2 \phi \nabla u\right) \cdot w \\
&= (-\mathrm{K}_0 P \nabla u + (1-\beta \mathrm{K}_0) K \nabla u) \cdot w \\
&= 0.
\end{aligned}$$

■

Corollary 1. *If* $\nabla u \cdot \nabla \phi = 0$,

$$\frac{\partial}{\partial \beta} \tilde{D}^2 u = 0$$

Proof. Repeated application of Lemma 1 gives

$$\frac{\partial}{\partial \beta}\left(\tilde{D}^2 u\right)_{i,j} = \left(\tilde{D}\left(\tilde{D}u \cdot e_i\right) \cdot e_j\right)_\beta = 0.$$

■

Remembering that $u_\beta = 0$ at $t = 0$, we calculate

$$\begin{aligned}
(u_\beta)_t &= \frac{\partial}{\partial \beta} F\left(x, u, \tilde{D}u, \tilde{D}^2 u\right) \\
&= \nabla_x F \cdot \nabla \phi + \frac{\partial F}{\partial r} u_\beta + \nabla_p F \cdot \left(\tilde{D}u\right)_\beta + \left\langle \nabla_X F, \left(\tilde{D}^2 u\right)_\beta \right\rangle \\
&= 0
\end{aligned}$$

by (23), Lemma 1, and Corollary 1. Theorem 1 follows.

5.3 Computing a non-degenerate PDE.

We gain an additional advantage when applying the modified method to diffusion equations. Consider solving the heat equation on a surface,

$$u_t = \Delta_S u,$$

and assume the initial condition u_0 satisfies (20). For simplicity, we examine the two dimensional case, for which the curvature tensor in Euclidean coordinates satisfies

$$\mathrm{K}(x) = \kappa(x)\mathrm{P}(x),$$

where κ is the curvature at x of the level curve of ϕ through x. We similarly have

$$\mathrm{K}_0(x) = \kappa_0(x)\mathrm{P}(x),$$

where $\kappa_0(x)$ is the curvature of the zero level set of ϕ through its closest point to x. Using this simplification, we expand $\Delta_s u$,

$$\Delta_s u = \left(\tilde{\mathrm{P}}\nabla\right)\cdot\left(\tilde{\mathrm{P}}\nabla u\right) = (\beta\kappa_0 - 1)\beta\nabla\kappa_0\cdot\mathrm{P}\nabla u + (1-\beta\kappa_0)^2\nabla\cdot(\mathrm{P}\nabla u), \tag{31}$$

and rewrite the heat equation:

$$u_t = (\beta\kappa_0 - 1)\beta\nabla\kappa_0\cdot\mathrm{P}\nabla u + (1-\beta\kappa_0)^2\nabla\cdot(\mathrm{P}\nabla u). \tag{32}$$

Equation (32) may be difficult to solve numerically, since it is a degenerate diffusion equation (there is no diffusion in the direction of $\nabla\phi$). We avoid solving (32) by using Theorem 1; because u remains constant along gradient lines of ϕ, we may add diffusion in this direction to (32):

$$u_t = (\beta\kappa_0 - 1)\beta\nabla\kappa_0\cdot\mathrm{P}\nabla u + (1-\beta\kappa_0)^2\nabla\cdot(\mathrm{P}\nabla u) + C(x)\nabla\cdot(\nabla\phi\otimes\nabla\phi\,\nabla u)\,. \tag{33}$$

If $C(x)\geq\nu$ for some $\nu>0$, equation (33) is a strictly parabolic diffusion equation, thus standard numerical convergence results apply [27]. We emphasize that equations (32) and (33) are equivalent only for functions satisfying (21), since

$$\nabla\phi\otimes\nabla\phi\,\nabla u = 0$$

in that case. We may choose $C(x)$ for convenience. Picking $C(x) = (1-\beta\kappa_0)^2$ for this example results in

$$u_t = (\beta\kappa_0 - 1)\beta\nabla\kappa_0\cdot\nabla u + (1-\beta\kappa_0)^2\Delta u. \tag{34}$$

In higher dimensions, the expression for $\Delta_S u$ is not as clean as in (32) or (34), but the same idea applies – Theorem 1 allows us to add diffusion in the direction normal to S without changing the solution on the surface.

6 Numerical examples in 2D.

6.1 Diffusion on the unit circle.

We return to the heat equation on the unit circle, (14). As discussed in Section 5.3, we derive equation (34) by adding diffusion in the radial direction, but since the circle has constant curvature, our equation in the embedding space is further simplified:

$$u_t = \frac{1}{(1+\phi\kappa)^2}\Delta u = \frac{1}{(1+\phi\Delta\phi)^2}\Delta u. \tag{35}$$

Using polar coordinates, we pick

$$u_0 = \sin\theta$$

for initial data, so the solution of (14) is

$$u(\theta,t) = \exp^{-t}\sin\theta. \tag{36}$$

Our distance function ϕ is defined in $[-2,2]\times[-2,2]$ on a uniform Cartesian grid. We compute in a band of fixed width, $|\phi(x)|\leq\frac{1}{10}$. We note that for our embedding of (14), u satisfies (36) throughout the band, and not only on the unit circle. We use the 'pseudo-Neumann' boundary conditions discussed in Section 3.3 – values of u at grid points outside the band are calculated using (21). We use the standard centered discretization of the Laplacian to compute (35), and we

(a) (b)

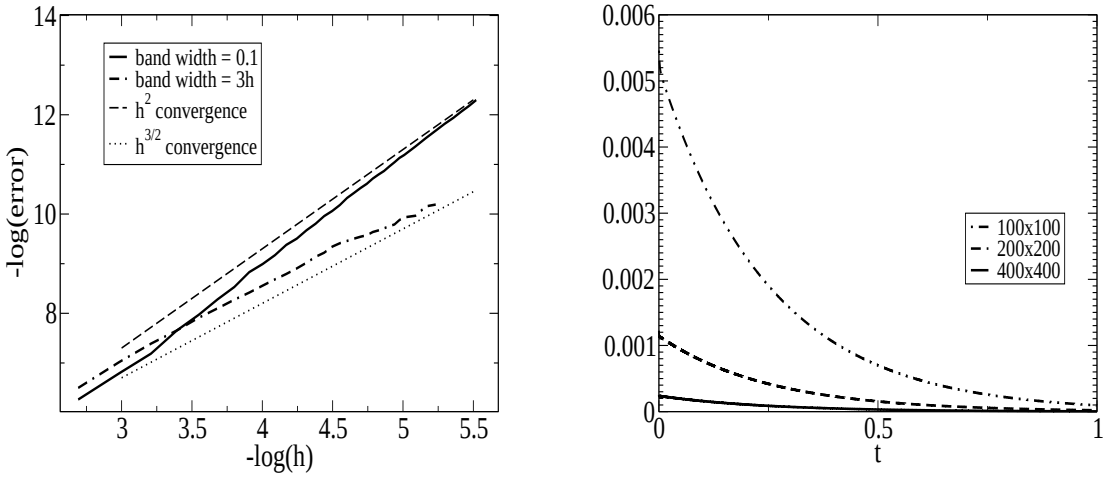

Figure 2: (a) Convergence test for the heat equation on the unit circle. The plot shows relative error computed at $t = 1$ using the l^∞ norm within Ω_c. We demonstrate results for both a fi xed band and one that shrinks with the grid size. (b) Graph of $\| \nabla u(t) \cdot \nabla \phi \|_\infty$ for different grid sizes.

use explicit time stepping with time step $k = \frac{1}{10}h^2$, where h is the grid cell width. Figure 2 graphs the error at $t = 1$ for various grid sizes. We use $M \times M$ grids where M varies from 20 to 1000. We observe $O(h^2)$ convergence of the relative error, which is computed using the l_∞ norm within the band.

Figure 2 also displays results from a slight change of the above experiment – we allow the band's width to shrink with the grid size. For each value of the grid cell width h, we compute in the band $|\phi| \leq 3h$. To our knowledge, none of the examples in [2, 4, 15, 35] use a band width that scales with the grid size. We do not observe $O(h^2)$ convergence, instead observing approximately $O(h^{\frac{3}{2}})$ convergence. The reason for the reduced rate is the subject of current investigation.

Figure 2 also shows the graph of $\| \nabla u \cdot \nabla \phi(t) \|_{l_\infty} = \| u_r(t) \|_{l_\infty}$ for various grid sizes. The added diffusion in the direction of $\nabla \phi$ inherent in (35) damps any numerical variations of u in this direction.

6.2 Advection on the unit circle.

We next solve an advection equation on the circle. We do not add any diffusion in the direction perpendicular to the surface. One purpose of this test is to check the growth of variations of u in this direction due to numerical error. We solve

$$u_t + u_\theta = 0 \tag{37}$$

with the initial condition

$$u(\theta, 0) = \cos^2 \theta. \tag{38}$$

We measure the l_∞ norm of the difference between our computed solution and the exact solution of (37) and (38),

$$u(\theta, t) = \cos^2(\theta - t).$$

(a) (b)

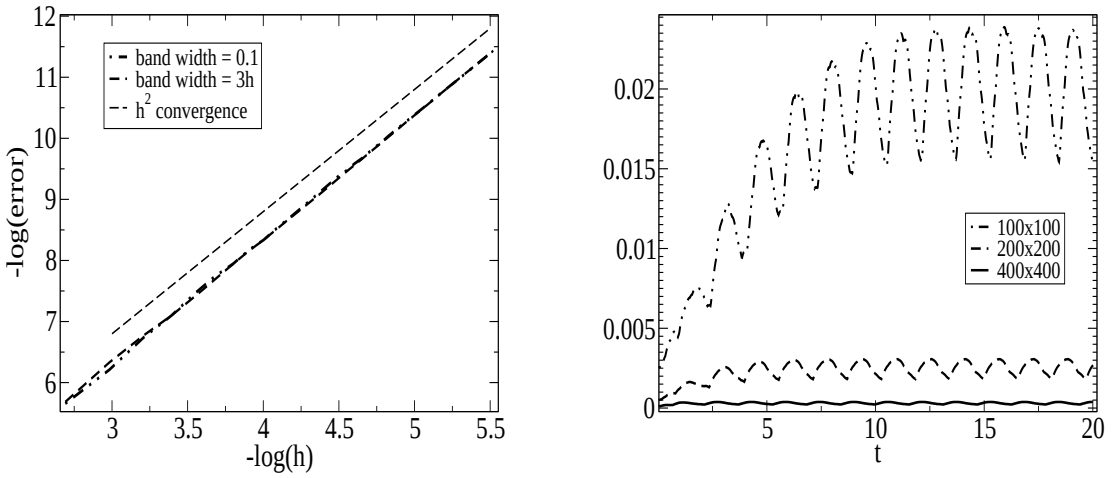

Figure 3: (a) Convergence test for the linear advection on the unit circle. The plot shows relative error computed at $t = 1$ using the l^∞ norm within Ω_c. (b) Graph of $\| \nabla u(t) \cdot \nabla \phi \|_\infty$ for different grid sizes.

The Eulerian representation of (37) that we solve is

$$u_t + V \cdot \tilde{\mathrm{P}} \nabla u = 0. \tag{39}$$

with velocity

$$V = (V_1, V_2) = (-\frac{y}{\sqrt{x^2+y^2}}, \frac{x}{\sqrt{x^2+y^2}}).$$

We use the Lax-Wendroff scheme [20] separately in each spatial direction and combine the two directions using Strang splitting [30] in time. This method is second order in both space and time. We use a time step of $k = \frac{1}{2}h$, where h is the uniform grid size on $[-2,2] \times [-2,2]$. Figure 3 shows $O(h^2)$ convergence for computations in the fixed band $|\phi(x)| \leq \frac{1}{10}$ and in a band that scales with h, $|\phi(x)| \leq 3h$. Figure 3 also shows graphs of $\| \nabla u \cdot \nabla \phi(t) \|_{l_\infty} = \| u_r(t) \|_{l_\infty}$ for various grid sizes. For each grid size, we used the fixed band $|\phi(x)| \leq \frac{1}{10}$.We observe an initial growth of variations of u in the direction of $\nabla \phi$, however these variations quickly stabilize.

6.3 Advection-diffusion equation on the unit circle.

We combine the above examples and solve

$$u_t + u_\theta = u_{\theta\theta} \tag{40}$$

with initial condition

$$u(\theta, 0) = \sin(2\theta). \tag{41}$$

The equation we solve is

$$u_t + V \cdot \tilde{\mathrm{P}} \nabla u = \frac{1}{(1+\phi\Delta\phi)^2} \Delta u. \tag{42}$$

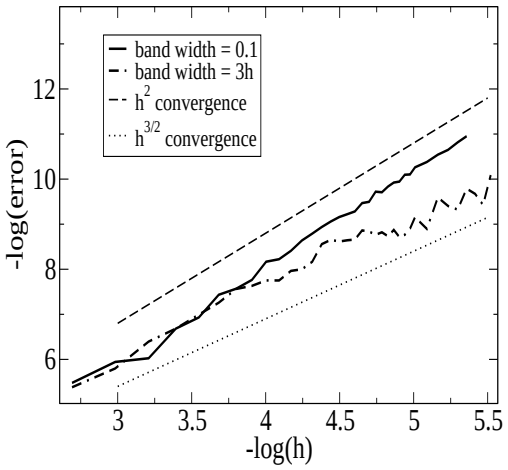

Figure 4: Convergence test for the linear advection-diffusion equation (40) on the unit circle. The plot shows relative error computed at $t = 1$ using the l^∞ norm.

We compare our solution of (42) with the exact solution

$$u(\theta,t) = e^{-4t}\sin(2(\theta - t)).$$

We combine centered differencing for all spatial derivatives in (42) with explicit time stepping, resulting in a scheme that is second order in space and first order in time. We use a time step of $k = \frac{1}{10}h^2$. Our results, plotted in Figure 4, demonstrate $O(h^2)$ convergence for a fixed band $|\phi(x)| \leq \frac{1}{10}$, and approximately $O(h^{\frac{3}{2}})$ convergence for computations in a band scaling with h, $|\phi(x)| \leq 3h$.

7 Numerical examples in 3D.

7.1 Diffusion on the unit sphere.

We next demonstrate the heat equation on a sphere. Since the sphere has constant curvature, the equation we solve is

$$u_t = \frac{1}{1 + \frac{\phi}{2}\Delta\phi}\Delta u. \tag{43}$$

Using spherical coordinates (r,θ,η), we pick

$$u_0(\eta) = \cos(\eta) \tag{44}$$

as an initial condition. Since this function is an eigenmode of the Laplace-Beltrami on the sphere, we calculate

$$u(\eta,t) = e^{-2t}\cos(\eta) \tag{45}$$

as the exact solution of (43) with initial condition (44). We compare our numerical solution of (43) with the exact solution (45) using the l_∞ norm. We compute on a cubic grid with grid cells

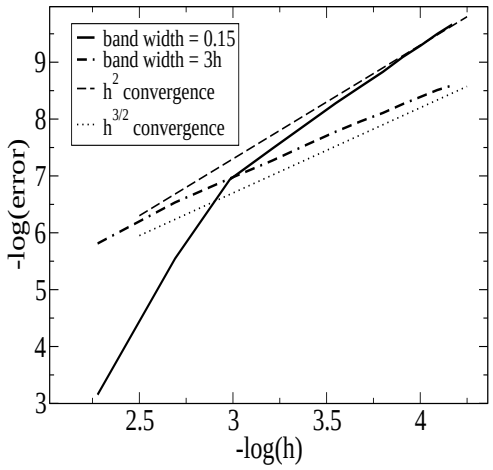

Figure 5: Convergence test for the heat equation (40) on the unit sphere. The plot shows relative error computed at $t = 1$ using the l^∞ norm.

of width h on each side. We use the standard centered discretization of the Laplacian combined with explicit time stepping using a time step of $k = \frac{1}{10}h^2$. Figure 5 demonstrates convergence of our scheme for both a band of fi xed width $|\phi| \leq \frac{3}{20}$ and a band that decreases with h, $|\phi| \leq 3h$. We observe $O(h^2)$ convergence for the band of constant width and approximately $O(h^{\frac{3}{2}})$ for the band width proportional to h.

7.2 Advection on a torus.

We compute an advection equation on a curved Torus with the signed distance function

$$\phi(x,y,z) = \sqrt{z^2 + \sqrt{x^2+y^2} - 1} - \frac{1}{2}. \tag{46}$$

Adopting the parametrization

$$S(\theta,\eta) = \left(\left(\frac{1}{2}\cos\eta + 1\right)\cos\theta, \left(\frac{1}{2}\cos\eta + 1\right)\sin\theta, \frac{1}{2}\sin\eta\right), \tag{47}$$

we solve

$$u_t + u_\eta = 0 \tag{48a}$$

$$u(\eta,0) = f(\eta). \tag{48b}$$

Letting

$$g(x) = \frac{e^{\frac{1}{x-1}} - e^{-\frac{1}{x}}}{e^{-\frac{1}{x}} + e^{\frac{1}{x-1}}},$$

we set the initial condition f to be

$$f(\eta) = \begin{cases} g(\frac{\eta+\pi}{\pi}) & -\pi \leq \eta \leq 0 \\ g(\frac{\pi-\eta}{\pi}) & 0 < \eta < \pi \end{cases}, \tag{49}$$

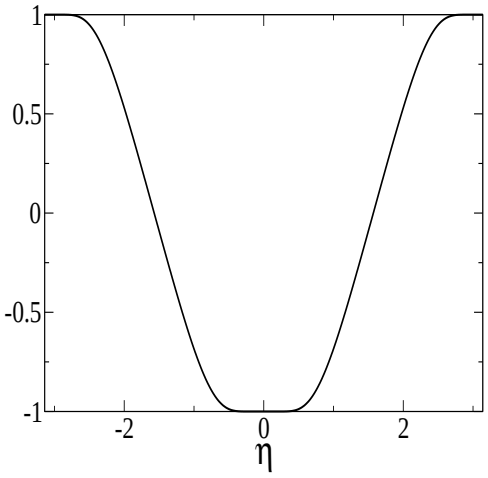

Figure 6: Graph of $f(\eta)$ defined by (49).

so that f is a smooth function of period 2π consisting of an infinite number of Fourier modes (see Figure 6). The exact solution of (48) is

$$u(\eta,t) = f(\eta - t).$$

We again use the Lax-Wendroff scheme separately in each spatial direction and combine the directions using Strang splitting [20, 30]. For the convergence test, we shrink the width of the band by requiring $|\phi| \leq 3h$. We observe $O(h^2)$ convergence of the relative error, as may be seen in Figure 7.

7.3 Allen-Cahn Equation on a torus.

For our final example, we compute the Allen-Cahn equation,

$$u_t = \varepsilon^2 \Delta_S u + u^3 - u, \tag{50}$$

on the torus in the previous example. Equation (50) was introduced in [3] to model the evolution of an order parameter representing the phase of a polycrystalline material. In this case, those phases are represented by $u = \pm 1$. A well known property of solutions to (50) is that the diffuse-interface between these two phases nearly moves by mean curvature. Evans, Soner, and Souganidis proved in [11] that under appropriate scalings, as $\varepsilon \to 0$ the limit of this interface motion is indeed motion by mean curvature.

We perform two sets of computations in the narrow band $|\phi(x)| \leq 3h$, where h is the cell width of the cubic $120 \times 120 \times 120$ grid on $[-2,2] \times [-2,2] \times [-2,2]$. We set $\varepsilon = 2h$ for both examples. When ε is chosen to be on the order of the grid size, care must be taken to check that no artificial pinning of interfaces occurs [5, 6]. There is no such pinning in our simulations, however, and we observe interfaces with a width of approximately five to seven grid cells. This width is comparable to calculations of diffuse-interface motion in other works [14, 15, 33].

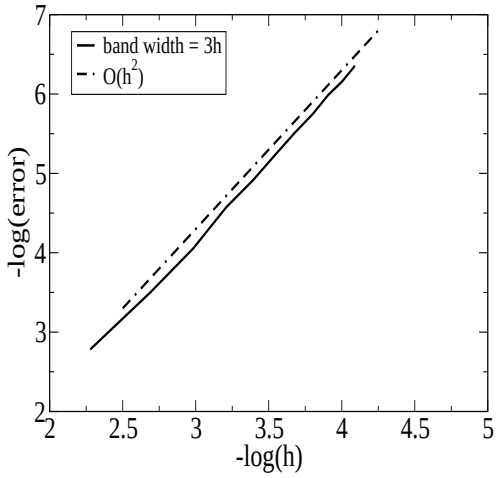

Figure 7: Convergence test for the linear advection on a curved torus in $\mathbb{R}^3$. The plot shows relative error computed at $t = 1$ using the l^∞ norm.

For the first example, we set the initial condition to be

$$f(\theta,\eta) = \begin{cases} 1 & \sqrt{(\theta-\frac{\pi}{2})^2+(\eta-\frac{\pi}{2})^2} \le \frac{4}{5} \\ -1 & \text{otherwise} \end{cases}, \tag{51}$$

so that f is 1 inside a convex region, and -1 outside that region. Figure 8 displays the results. Although the initial data is discontinuous, the diffusion equation quickly creates a smooth interface separating the two regions. The convex region shrinks slowly at first, but the dynamics accelerate as the interface asymptotically approaches a circle before disappearing.

In the second example, we use the more complicated initial condition

$$f(\theta,\eta) = \frac{2}{5}\left(\sin(14\theta)\cos(8\theta) + \cos(11\theta)\cos(6\eta)\right). \tag{52}$$

In this case u first evolves into a function that mostly takes on the values ± 1 with smooth interfaces between them. The interfaces then move by mean curvature until reaching the steady state shown in the last frame of Figure 9. The phases have separated into rings around the torus. Although the boundaries of these rings are curved in the embedding space, they have zero curvature on the surface of the torus and thus remain stationary.

8 Conclusions.

We have introduced a modification to a recent method for computing PDEs on surfaces by representing the surface as the level set of a signed distance function ϕ. By taking into account the curvatures of the different level sets of ϕ, the modification results in PDEs that are better behaved than those derived from the original method in [2, 4, 15, 21, 35]. We have not considered dynamically evolving surfaces as in [2, 35], but believe this can be done as was the case for the original method. It will be most effective to move the surface by an external velocity field that preserves

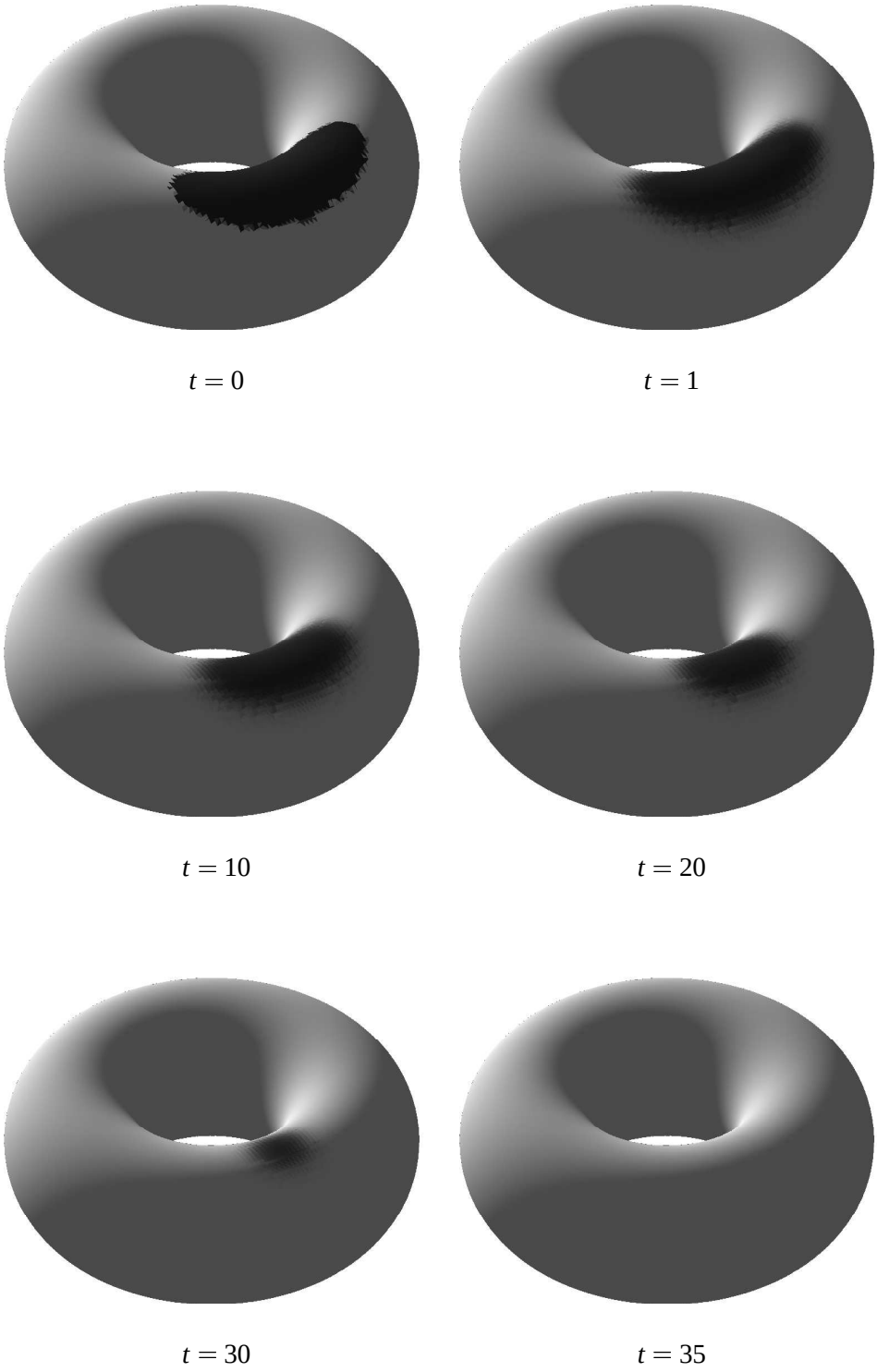

Figure 8: Allen-Cahn equation. Motion by mean curvature shrinks convex regions asymptotically to a circle before disappearing. We used the Matlab routine *isosurface* to render the surface. Any jagged appearance of the interface is due to the visualization.

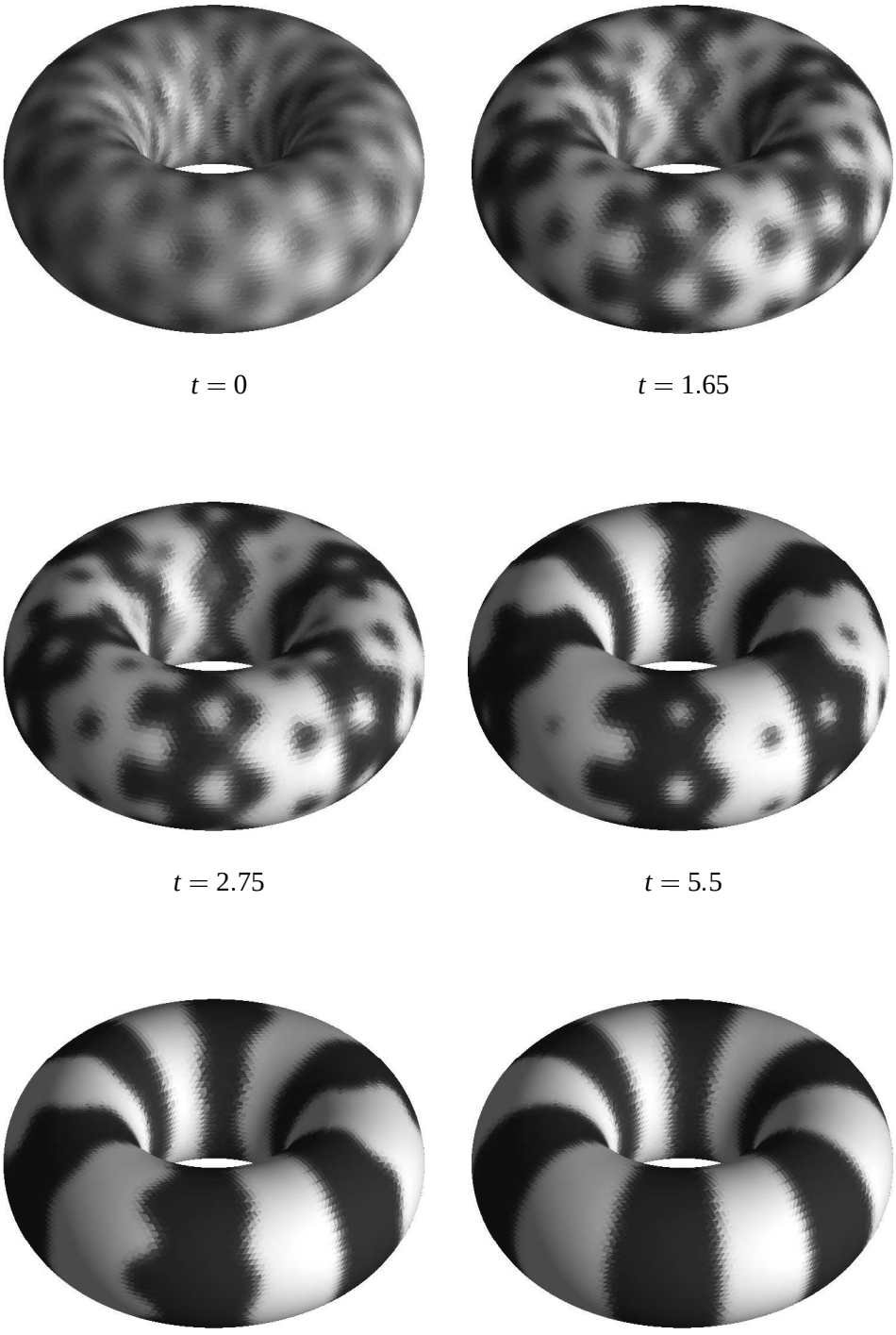

Figure 9: Allen-Cahn equation. Although the rings around the torus in the final frame have nonzero curvature in the embedding space, they have zero curvature on the torus and are thus a steady state. We rendered the surface using the Matlab routine *isosurface*.

the property $|\nabla\phi| = 1$ so that ϕ remains a signed distance function. This may be done by extending the surface velocity as in [2] and [35]. We also wonder if these ideas have analogues for more typical level set methods, where a decision is often made on the PDE to be solved in the embedding space. Perhaps there are more appropriate extensions of interface velocities or initial conditions to be discovered.

Acknowledgements.

Thanks to Robert V. Kohn for valuable discussion and suggestions. The author completed this work at the Courant Institute for Mathematical Sciences while receiving a National Science Foundation Postdoctoral Research Fellowship.

References

[1] D. Adalsteinsson and J. A. Sethian. A fast level set method for propagating interfaces. *J. Comput. Phys.*, 118(2):269–277, 1995.

[2] D. Adalsteinsson and J. A. Sethian. Transport and diffusion of material quantities on propagating interfaces via level set methods. *J. Comput. Phys.*, 185(1):271–288, 2003.

[3] S. M. Allen and J. W. Cahn. A macroscopic theory for antiphase boundary motion and its application to antiphase domain coarsing. *Acta. Metal.*, 27:1085–1095, 1979.

[4] M. Bertalmío, L. T. Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. *J. Comput. Phys.*, 174(2):759–780, 2001.

[5] J. W. Cahn, J. Mallet-Paret, and E. S. Van Vleck. Traveling wave solutions for systems of ODEs on a two-dimensional spatial lattice. *SIAM J. Appl. Math.*, 59(2):455–493, 1998.

[6] A. Carpio and L. L. Bonilla. Depinning transitions in discrete reaction-diffusion equations. *SIAM J. Appl. Math.*, 63(3):1056–1082, 2003.

[7] D. L. Chopp. Some improvements of the fast marching method. *SIAM J. Sci. Comput.*, 23(1):230–244, 2001.

[8] U. Diewald, T. Preufer, and M. Rumpf. Anisotropic diffusion in vector fi eld visualization on Euclidean domains and surfaces. *IEEE Trans. Visualization Comput. Graphics*, 6:139–149, 2000.

[9] M. P. do Carmo. *Differential geometry of curves and surfaces*. Prentice-Hall Inc., Englewood Cliffs, N.J., 1976. Translated from the Portuguese.

[10] J. Dorsey and P. Hanrahan. Digital materials and virtual weathering. *Sci. Am.*, 282(2):46, 2000.

[11] L. C. Evans, H. M. Soner, and P. E. Souganidis. Phase transitions and generalized motion by mean curvature. *Comm. Pure Appl. Math.*, 45(9):1097–1123, 1992.

[12] L. C. Evans and J. Spruck. Motion of level sets by mean curvature. II. *Trans. Amer. Math. Soc.*, 330(1):321–332, 1992.

[13] S. F. Frisken, R. N. Perry, A. Rockwood, and T. Jones. Adaptively sampled fi elds: A general representation of shape for computer graphics. *ACM SIGGRAPH*, 2000.

[14] K. Glasner. A diffuse interface approach to Hele-Shaw flow. *Nonlinearity*, 16(1):49–66, 2003.

[15] J. B. Greer, A. L. Bertozzi, and G. Sapiro. Fourth order partial differential equations on general geometries. UCLA CAM preprint, 05-17.

[16] D. Halpern, O. E. Jensen, and J. B. Grotberg. A theoretical study of surfactant and liquid delivery into the lung. *J. Appl. Physiology*, 85:333–352, 1998.

[17] M. Hofer and H. Pottmann. Energy-minimizing splines in manifolds. *ACM Trans. Graphics*, 2004.

[18] H. Hoppe and M. Eck. Automatic reconstruction of b-spline surfaces of arbitrary topological type. *ACM SIGGRAPH*, 1996.

[19] V. Krishnamurthy and M. Levoy. Fitting smooth surfaces to dense polygon meshes. *ACM SIGGRAPH*, pages 313–324, 1996.

[20] P. Lax and B. Wendroff. Systems of conservation laws. *Comm. Pure Appl. Math.*, 13:217–237, 1960.

[21] F. Mémoli and G. Sapiro. Fast computation of weighted distance functions and geodesics on implicit hyper-surfaces. *J. Comput. Phys.*, 173(2):730–764, 2001.

[22] F. Memoli, G. Sapiro, and P. Thompson. Implicit brain imaging. *Human Brain Mapping*, 23:179–188, 2004.

[23] T. G. Myers and J. P. F. Charpin. A mathematical model for atmospheric ice accretion and water flow on a cold surface. *International Journal of Heat and Mass Transfer*, 47(25):5483–5500, 2004.

[24] T. G. Myers, J. P. F. Charpin, and S. J. Chapman. The flow and solidifi cation of a thin fluid fi lm on an arbitrary three-dimensional surface. *Phys. Fluids*, 14(8):2788–2803, 2002.

[25] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153 of *Applied Mathematical Sciences*. Springer-Verlag, New York, 2003.

[26] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local level set method. *J. Comput. Phys.*, 155(2):410–438, 1999.

[27] R. D. Richtmyer and K. W. Morton. *Difference methods for initial-value problems*. Robert E. Krieger Publishing Co. Inc., Malabar, FL, second edition, 1994.

[28] J. A. Sethian. *Level set methods and fast marching method. Evolving interfaces in computational geometry, fluid mechanics, computer vision, and materials sciences*, volume 3 of *Cambridge Monographs on Applied and Computational Mathematics*. Cambridge University Press, Cambridge, second edition, 1999.

[29] L. Simon. *Lectures on geometric measure theory*, volume 3 of *Proceedings of the Centre for Mathematical Analysis, Australian National University*. Australian National University Centre for Mathematical Analysis, Canberra, 1983.

[30] G. Strang. On the construction and comparison of difference schemes. *SIAM J. Numer. Anal.*, 5:506–517, 1968.

[31] A. Toga. *Brain Warping*. Academic Press, New York, 1998.

[32] G. Turk. Generating textures on arbitrary surfaces using reaction-diffusion. *Comput. Graphics*, 25(4), 1991.

[33] B. P. Vollmayr-Lee and A. D. Rutenberg. Fast and accurate coarsening simulation with an unconditionally stable time step. *Phys Rev E*, 68, 2003.

[34] A. Witkin and M. Kass. Reaction-diffusion textures. *Computer Graphics(SIGGRAPH)*, 25(4):299, 1991.

[35] J. Xu and H. K. Zhao. An Eulerian formulation for solving partial differential equations along a moving interface. *J. Sci. Comput.*, 19(1-3):573–594, 2003. Special issue in honor of the sixtieth birthday of Stanley Osher.

[36] G. Yngve and G. Turk. Creating smooth implicit surfaces from polygonal meshes. *Technical Report GIT-GVU-99-42, Graphics, Visualization, and Usability Center. Georgia Institute of Technology*, 1999.