Diffusion-Generated Motion by Mean Curvature for Filaments

Steven J. Ruuth[*], Barry Merriman[†], Jack Xin[‡] and Stanley Osher[§]

Abstract

In previous work, we introduced a simple algorithm for producing motion by mean curvature of a surface, in which the motion is generated by alternately diffusing and renormalizing a characteristic function. This procedure is interesting due to its extreme simplicity, and because it isolates the connection between diffusion and curvature motion. However, it makes sense only for surfaces, i.e. objects of dimension $d-1$ inside of R^d.

In this paper, we generalize diffusion-generated motion to a procedure that can be applied to objects of any dimension k inside of R^d, $k < d$. We focus on generating the curvature motion of filaments, i.e. curves in R^3, since this is an important special case which also illustrates the general ideas. The method for filaments consists of applying diffusion to a complex valued function whose values wind around the filament, followed by normalization. We motivate this approach by considering the essential features of the complex Ginzburg-Landau equation, which is a reaction-diffusion PDE that describes the formation and propagation of filamentary structures. The new

[*] Department of Mathematics, University of California at Los Angeles. (ruuth@math.ucla.edu). The work of this author was partially supported by an NSERC Postdoctoral Scholarship, NSF DMS-9706827 and DMS-9615854.

[†] Department of Mathematics, University of California at Los Angeles. (barry@math.ucla.edu). The work of this author was partially supported by NSF DMS-9706827 and DMS-9615854.

[‡] Department of Mathematics, University of Arizona, Tucson, AZ 85721.(xin@math.arizona.edu). The work of this author was partially supported by NSF DMS-9615854.

[§] Department of Mathematics, University of California at Los Angeles. (sjo@math.ucla.edu). The work of this author was partially supported by NSF DMS-9706827 and DMS-9615854.

algorithm naturally captures topological merging and breaking of filaments without fattening curves. It also gives an improved computational efficiency over direct numerical solution of the Ginzburg-Landau equation.

We justify the new algorithm with asymptotic analysis and numerical experiments for the curvature motion of filaments in R^3. However, we do not have a rigorous proof of its convergence. We also discuss the generalization of this procedure for motion by mean curvature of objects of arbitrary codimension, as well as generalizations that allow for a large class of velocity laws.

1 Introduction

Diffusion-generated motion by mean curvature is a particularly simple and robust algorithm for producing motion by mean curvature of a surface [16, 17]. The major goal underlying this work is to generalize this algorithm from surfaces (dimension $d-1$ inside R^d) to objects of arbitrary dimension k inside R^d. To guide the generalization and connect it to a class of interesting problems, we concentrate on the special case of producing motion by curvature of a curve—or “filament”—in three dimensions, and return to the general dimensional case only towards the end.

The motion of filaments is of particular interest because many physical and mathematical systems exhibit the formation and propagation of filamentary structures. Notable examples include magnetic flux tubes trapped in superconductors, vortex filaments in inviscid fluids, the centers of scroll waves in excitable media, biological polymers such as protein and DNA, and skeleton curves extracted from processing 3-D images in computer vision.

Asymptotic models for these processes often yield equations of motion for a curve moving with a velocity that is a function of its local geometry, i.e. a function of the local normal and binormal direction, curvature, torsion, and higher space and time derivatives of these quantities. For example, studies of models for superconductors and excitable media predict that their vortex filaments evolve asymptotically with a speed proportional to curvature [6, 25].

Given such models, it becomes important to consider algorithms which can realize geometric filament motions in simple, efficient and accurate ways, and which are amenable to mathematical analysis. Designing suitable algorithms is complicated by the fact that in many problems the filaments can merge or break up. It is particularly challenging to find algorithms that retain their simplicity, yet are robust enough to capture these topological transitions.

For surfaces (or, generally, codimension one objects) the level set method of Osher and Sethian [18] was introduced to compute (and define) arbitrary curvature-dependent surface motions, including topological changes. This provides a PDE based method for motion by mean curvature, including the pinch-offs which can occur in three dimensions. Standard numerical PDE methods apply to accurately discretize the equations of motion. However, the original level set method does not directly apply to objects of higher codimension, such as filaments. The level set method was ultimately extended to arbitrary codimension [1]. In this approach, the object is represented by its squared distance function, or any other similar smooth function. The modification leads to two significant practical difficulties. First, the representation is not robust: A perturbation of the level set function φ can inadvertently break up the filament since its representation is given by $\{\vec{x} : \varphi(\vec{x}) < \epsilon\}$ for a small, positive ϵ and $\varphi \geq 0$. Second, the method has the undesirable property that filaments tend to develop interiors whenever mergers occur. See [3] for a detailed discussion on this "fattening phenomenon."

Alternatively, the curvature motion of filaments (or the mean curvature motion of surfaces) may be approximated using reaction-diffusion models such as the complex Ginzburg-Landau equation. These *phase field* models are reviewed in Sections 2.1 and 3.2. Briefly, these methods have the important advantage that they automatically capture the curvature motion of filaments including topological change without fattening curves. When used in computation, however, the spatial discretization must resolve a thin reaction zone in order to accurately compute the motion. Since the width of the front is $O(\epsilon)$, the only remedy is to use a mesh spacing which is much less than ϵ, which can be impractical numerically [17].

To overcome this limitation in the case of surface motion, an algorithm based on an idealization of reaction-diffusion was presented in [16, 17]. This algorithm is described in detail in Section 2.2, but it essentially consists of moving a set boundary by alternately "diffusing" the set—i.e. applying the linear diffusion evolution equation to the set's characteristic function for a short time—and then recovering a new set via a "sharpening" step in which values of the diffused characteristic function are re-normalized to 0 or 1, whichever is closer. This surprisingly simple "diffusion-generated motion by mean curvature" algorithm automatically captures topological change and has a direct extension to a variety of interesting anisotropic motions [10, 23, 11] as well as the motion of triple point junctions and arbitrary networks of surfaces [16, 15, 17, 21]. Furthermore, the method has the advantage that it can be discretized efficiently and accurately using adaptive grid refinement with fast Fourier transforms [22]. Unfortunately, the original method does not apply to objects with higher codimension, such as filaments.

In this present work, we generalize the original diffusion generated motion algorithm to filaments, via a natural idealization of the complex Ginzburg-Landau model. This diffusion-generated filament motion naturally computes the (vector) mean curvature motion, including topological changes without curve fattening. The method also gives improved computational efficiency over reaction-diffusion models since it does not require resolving an artificial small scale parameter, ϵ. Furthermore, the method has a variety of natural extensions. These include the motion by (vector) mean curvature of objects of arbitrary codimension, as well as the anisotropic curvature motion of filaments.

The outline of the paper is as follows. In Section 2, we review diffusion-generated motion and its connection to the Ginzburg-Landau equation. Section 3 begins by reviewing the complex Ginzburg-Landau equation. Using this phase field model as an inspiration and motivation, a diffusion-generated algorithm for the curvature-dependent motion of filaments is derived. In Section 4, we give an asymptotic justification that diffusion-generated filament motion gives motion by curvature in the normal direction. Section 5 reports on a variety of experiments validating our algorithm. Finally, in Section 6 we discuss the extension to arbitrary codimension, extensions to obtain other velocity laws, and other possible variations on this approach.

2 Diffusion-Generated Motion of Surfaces

In this section, we review diffusion-generated motion for surfaces of codimension 1, and its connection to the Ginzburg-Landau equation. Later sections will use this same methodology to develop diffusion based methods for the curvature dependent motion of filaments.

2.1 The Ginzburg-Landau Equation

The real-valued Ginzburg-Landau equation (or "phase-field equation") is a well-known PDE model in which sharp reaction fronts form and propagate. This provides a robust means of describing interface dynamics. The equation has the form

$$u_t = \nabla^2 u - \frac{1}{\epsilon^2} f(u)$$

where $\epsilon > 0$ is a small parameter and f is the derivative of a double well potential (so that the PDE also has a variational formulation as steepest descent minimization of an energy functional). For example, if we take

$$f(u) = (\frac{u^4}{4} - \frac{u^2}{2})' = u(u^2 - 1)$$

and define the interface precisely as the zero level set of u, then in the asymptotic limit as $\epsilon \to 0$ the interface moves with a normal velocity equal to the mean curvature of the front (see, e.g., [8]).

From a theoretical and computational standpoint, this phase field model has the benefit that topological shape changes such as merger and pinch off are treated automatically. Indeed, whenever two curves intersect this approach automatically selects the solution which (locally) produces the most rapid curve shortening. This is unlike solutions to level set PDEs which can develop interiors. See Fig. 1. Unfortunately, if phase field PDEs are used in computation it is necessary to resolve the thin $O(\epsilon)$ wide reaction zone to obtain numerical accuracy, and in turn the model only gives the desired motion in the $\epsilon \to 0$ limit. As a result, the numerical grid size Δx is constrained by $\Delta x \ll \epsilon \ll L$, where L is the scale of the radius of curvature of the interface [17]. This restriction can make calculations too slow for practical use, and in any case is much worse than the natural resolution requirement of interface, $\Delta x \ll L$.

Fortunately, improved efficiency with optimal curve shortening can be obtained by formally "splitting" the reaction-diffusion equation into its component parts. This splitting yields a simple *diffusion-generated motion*, which is the focus of the next section.

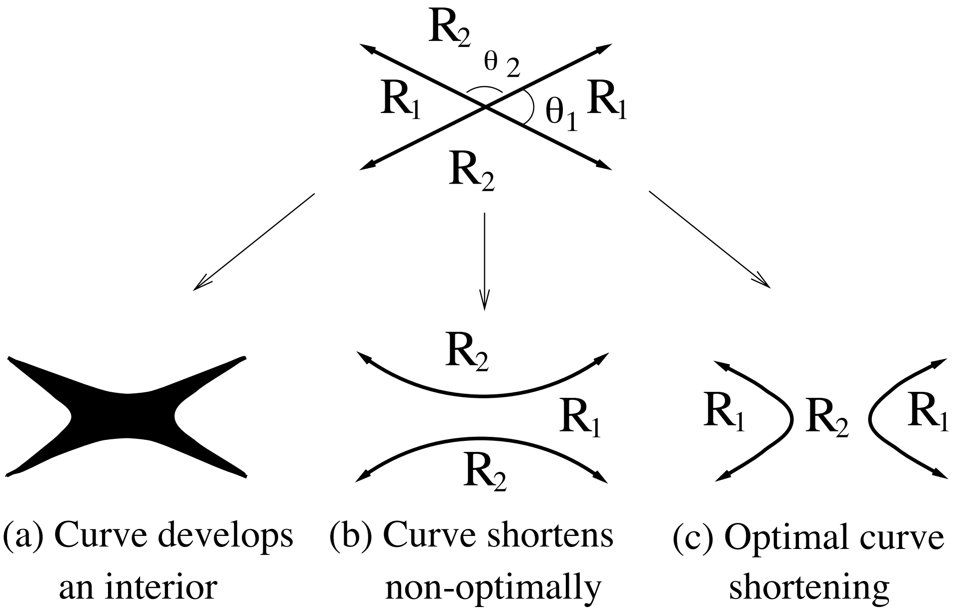

R2
θ 2
R1
θ1
R1
R2
R2
R1
R2
R1
R2
R1
(a) Curve develops an interior
(b) Curve shortens non-optimally
(c) Optimal curve shortening

Figure 1: Under curvature motion, methods which capture the interface produce different results when intersections occur. (a) For nonsmooth initial data, curves can develop interiors using level set methods. (b) A non-optimal curve shortening solution. (c) If $\theta_2 > \theta_1$ then the most rapid curve shortening occurs when region one pinches off. Remarkably, this "optimal curve shortening" solution is automatically selected by the phase field approach as well as the diffusion-generated algorithm discussed in Section 2.2.

2.2 Diffusion-Generated Motion by Mean Curvature

Rather than obtaining motion by mean curvature of a surface via a singular reaction-diffusion model, it is possible to use an idealized reaction-diffusion process that achieves the desired motion directly, without introducing artificial small scales [16, 17]. The algorithm can be motivated by considering a formal time splitting of the reaction-diffusion processes. If we start from any initial data function $\chi(\vec{x}, t_0)$, and apply the reaction kinetics for a short time Δt,

$$\begin{aligned} \bar{\chi}_t &= -\frac{1}{\epsilon^2}\bar{\chi}(|\bar{\chi}|^2 - 1) \\ \bar{\chi}(\vec{x}, 0) &= \chi(\vec{x}, t_0) \end{aligned}$$

we obtain an intermediate result $\bar{\chi}(\vec{x}, \Delta t)$, with values that have been strongly driven towards the stable equilibrium values ±1. We follow this by the diffusion process

$$\begin{aligned} \chi_t &= \nabla^2 \chi \\ \chi(\vec{x}, 0) &= \bar{\chi}(\vec{x}, \Delta t) \end{aligned}$$

for a time Δt, to obtain the reaction-diffusion update $\chi(\vec{x}, t_0 + \Delta t)$. If we take the formal $\epsilon \to 0$ limit in the reaction process, the reaction step drives all values completely to the nearest equilibrium value, and reduces to the simple normalization procedure

$$\bar{\chi} = \begin{cases} 1 & \text{if } \chi \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

which can also be written as

$$\bar{\chi} = \frac{\chi}{|\chi|}.$$

In so doing, we obtain the diffusion-generated motion by mean curvature algorithm[1] [16, 17].

ALGORITHM **DGM**
GIVEN: An initial region R.

[1] We have rephrased the algorithm slightly from the authors' original description to facilitate the generalization of Section 3.2.

```
BEGIN
    (1) “Initialize”: Set $\chi$ equal to *any* function that is positive inside
        and negative outside $R$.
    (2) Repeat for all steps:
        (a) “Normalize”: $\bar{\chi} = \frac{\chi}{|\chi|}$
        (b) “Diffuse”: Starting from $\bar{\chi}$, evolve $\chi$ for a time $\Delta t$ according to $\chi_t = \nabla^2 \chi$.
END
```

The location of the interface is given by the zero level set of χ.

It is important to note that formal derivation given in no way proves this scheme produces motion by mean curvature in the continuous time limit, nor does it even strongly suggest that this might be true. This is because the $\epsilon \to 0$ limit, followed by the continuous time limit $\Delta t \to 0$, is by no means guaranteed to be equivalent to the standard convergent time-splitting, which in this case would require Δt and ϵ go to zero simultaneously in such a way that $\Delta t << \epsilon$. Thus while the above formalism yields the diffusion generated motion algorithm, it in no way guarantees that this algorithm works, i.e. gives the same interface motion (motion by mean curvature) as the singular limit of the reaction-diffusion PDE. Indeed, such naive limit-taking is quite likely to give totally wrong behavior in a singular perturbation problem. Nonetheless, the resulting algorithm does indeed produce motion by mean curvature in the $\Delta t \to 0$ continuous time limit, but this fact requires totally new justification, independent of the properties of reaction-diffusion model that motivated the scheme.

Because of this, to appreciate why the method works it is better to forget the formal derivation, and simply note that the diffusion step of the algorithm causes a curvature-dependent blurring of the set boundary. Thus the updated boundary will be displaced by a curvature dependent distance, and a formal analysis of the diffusion equation [15, 16, 17] shows this should result in precisely motion by mean curvature. Indeed, a variety of rigorous proofs have been given to show this simple algorithm converges to motion by mean curvature in any number of dimensions as the time step size goes to zero [7, 2, 11].

This algorithm has several remarkable properties. Similar to the phase field approach, motion by mean curvature is obtained without ever directly computing the mean curvature and topological mergers such as pinch off are captured with no special algorithmic procedures. Indeed, it is easily seen that whenever two curves intersect this approach automatically selects the solution which (locally) produces the most rapid curve shortening. Furthermore, diffusion generated motion enjoys an important advantage over the phase field approach: Phase-field models introduce an artificial small length scale—the width of the reaction zone—which, for numerical work, must be resolved by a computational grid or all accuracy is lost for the computed interface motion. (In fact, if the grid spacing is not smaller than ϵ, the initial fronts are actually frozen in place, and no front motion occurs at all [17].) In contrast, diffusion-generated motion has no such artificial small scales. For numerical work, the computational method need only resolve the natural length scales in the problem, i.e. the curvatures of the evolving surface. Thus diffusion-generated motion has in effect passed to the asymptotic limit of the phase-field class of models, obtaining a simplified and more accurate evolution scheme in the process.

3 Diffusion-Generated Motion of Filaments

We have seen that the real-valued Ginzburg-Landau equation gives an intuition for deriving a diffusion-generated algorithm for motion by mean curvature for surfaces (of codimension one). In this section, we review the complex Ginzburg-Landau model for evolving filaments with a normal velocity equal to the (vector) curvature. Then, following the formal derivation of the of the previous section, we idealize this reaction-diffusion model to obtain a diffusion-generated algorithm for the curvature-dependent motion of filaments in three dimensions. Later sections will justify our proposed method with asymptotics and numerical experiments, and also extend it to arbitrary dimensions and codimensions.

3.1 The Complex Ginzburg-Landau Equation

The complex Ginzburg-Landau equation is:

$$u_t = \Delta u - \frac{1}{\epsilon^2} u(|u|^2 - 1), \tag{1}$$

where $u(x,t)$ is a complex scalar and $0 < \epsilon \ll 1$. is a basic model for understanding the motion of phase defects (singularities). For $\vec{x} \in R^3$ the defects are generically supported on the one dimensional curve (filament) where $|u^\epsilon|$ vanishes. Equation (1) is the magnetic field-free case of the time dependent Ginzburg-Landau system, which models the mixed states in type-II superconductors where magnetic flux carrying normal filaments are embedded in a superconducting matrix, [6]. The complex scalar u^ϵ is an order parameter, representing normal phase if $|u^\epsilon|$ is close to zero and super phase if close to one, and $\epsilon \ll 1$ is the effective diameter of the magnetic flux core. Equation (1) is also a generic amplitude equation describing instabilities near bifurcation points in dissipative systems, known as the Landau-Stuart equation, [13].

Asymptotic analysis can be used to extract the $\epsilon \to 0$ limiting behavior of solutions. For initial data $u(x, 0)$ vanishing on a filament Γ_0 and having winding number one around it, formal asymptotic derivation [19] shows that solution evolves to leading order as a complex scalar vanishing along the filament Γ_t which is generated from Γ_0 as motion by curvature along the normal. If the filaments are nearly parallel, rigorous results are established in [14] on their dynamics on the $O(\log \frac{1}{\epsilon})$ time scale.

Numerically, small ϵ introduces small length and time scales into the dynamics. Consequently, an accurate direct simulation of (1) has to resolve the core size and reaction rate, an expensive task in three dimensions. However, it turns out we can capture the desired limiting filament dynamics with a complex diffusion-generated motion algorithm obtained by idealizing the effect of the strong reaction in (1).

3.2 Complex Diffusion-Generated Motion

Similar to the case of diffusion-generated motion, a formal splitting method can be applied to the complex Ginzburg-Landau equation to obtain an algorithm for motion by mean curvature of filaments. In the reaction step, an initial complex-valued $\chi(\vec{x}, t_0)$ is driven towards one of its stable equilibrium values $e^{i\theta}$ by the reaction kinetics,

$$\begin{aligned} \bar{\chi}_t &= -\frac{1}{\epsilon^2}\bar{\chi}(|\bar{\chi}|^2 - 1) \\ \bar{\chi}(\vec{x}, 0) &= \chi(\vec{x}, t_0) \end{aligned}$$

for a time Δt to obtain an intermediate result $\bar{\chi}(\vec{x}, \Delta t)$. This result is subsequently diffused for a time Δt,

$$\begin{aligned} \chi_t &= \nabla^2 \chi \\ \chi(\vec{x}, t) &= \bar{\chi}(\vec{x}, \Delta t) \end{aligned}$$

to obtain the desired update $\chi(\vec{x}, t_0 + \Delta t)$. By replacing the reaction step by its formal limit as $\epsilon \to 0$, it becomes the simple normalization to a unit complex number

$$\bar{\chi} = \frac{\chi}{|\chi|}$$

and we obtain the following method for (hopefully) evolving filaments with a normal velocity equal to curvature:

```
ALGORITHM **CDGM**
GIVEN: An initial filament.

BEGIN
  (1) "Initialize": Set $\chi$ so that its "center of winding" coincides with the filament.
      I.e. set $\chi$ so that its winding number is nonzero around any closed curve
      that winds around the filament. See next section.
  (2) Repeat for all steps:
      (a) "Normalize": $\bar{\chi} = \frac{\chi}{|\chi|}$
      (b) "Diffuse": Starting from $\bar{\chi}$, evolve $\chi$ for a time $\Delta t$ according to $\chi_t = \nabla^2 \chi$.
END
```

The location of the interface is given by the zero contour of χ (or, equivalently, its center of winding, though this is more difficult to locate in practice).

As we shall see in the analysis of Section 4, this simple splitting method captures the leading order behavior of the complex Ginzburg-Landau equation: I.e., it produces a normal velocity equal to the curvature of the filament without ever directly computing curvature. Topological mergers are also captured with no special algorithmic procedures. In particular, filaments do not develop interiors (unlike level set methods for filament motion — see [3]) and a good agreement with optimal curve shortening (see Fig. 1) is observed. Furthermore, the method enjoys some important computational and theoretical advantages over phase field methods. For example, the ALGORITHM **CDGM** does not introduce any small spatial scale ϵ. Thus, it need only resolve the natural length scales of the problem (i.e., the curvature of the evolving filament) rather than the width, ϵ, of the reaction zone. Also note that the method has the advantage that the diffusion step is entirely linear with the only nonlinear part being the final, trivial, normalization step.

We now complete our description of the algorithm with a discussion on the initialization of χ.

3.3 Initialization of χ

To apply the ALGORITHM **CDGM**, an initial value of χ is required. This value may be determined in a number of ways, depending on the problem.

If the filament is already defined implicitly as the zero of a function u arising from a complex Ginzburg-Landau equation, then we simply set $\chi = u$ to initialize. But, in general, we need to construct a $\chi : R^3 \to C$ which implicitly captures the position of the filament. We use the same representation as in the complex Ginzburg-Landau equation. Specifically, we construct a complex-valued χ so that the winding number of $\chi(\vec{x})$ (with respect to zero in the complex plane) is nonzero when $\vec{x}$ moves around any closed loop that encircles the filament. Along other loops, however, the winding number must be zero to avoid creating spurious filaments. See Fig. 2.

In any such construction, the basic idea is to mimic the way $e^{i\theta}$ winds around the origin in the complex plane. Thus we will define $\chi(\vec{x}) = e^{i\theta(\vec{x})}$, where $\theta(\vec{x})$ is some angle variable defined on all of R^3 that winds around the filament similar to the way the polar angle winds around the z axis in a cylindrical coordinate system. There are a number of convenient ways to define such an angular variable relative to a given filament. For example, we may define a planes and reference axis through each point on the filament, so that the planes fill out R^3(see Fig. 3a). The initialization on a particular plane is then given by $\chi(\vec{x}) = \exp(i\theta(\vec{x}))$ where $\theta(\vec{x})$ the angle function in that plane, measured relative to the reference axis in the plane. As shown in Fig. 3b, this type of initialization is particularly straightforward whenever the filament can be represented as a function of z in some coordinate system. Looping structures such as rings and linked rings are also easily initialized in this manner. See Section 5.

Alternatively, whenever two surfaces can be found whose intersection gives the desired filament, a simple shape-based initialization can be used (see Fig. 4). Here, we set a function χ_{Re} equal to $+1$ inside and -1 outside

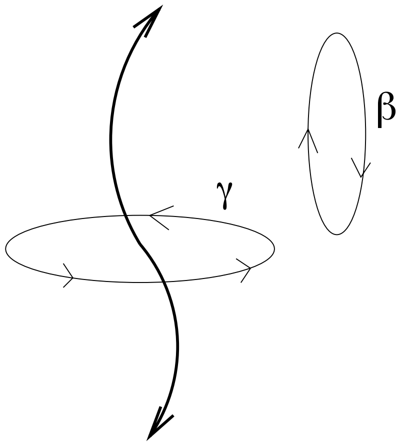


Figure 2: An initial χ is chosen so that its winding number (with respect to zero) is nonzero around any closed curve γ that winds around the filament. For all other curves (e.g., β), the winding number is zero.

the first surface. Similarly, we set χ_{Im} equal to $+i$ inside and $-i$ outside the second surface. The initial value of χ is then given naturally as the sum of χ_{Re} and χ_{Im}. Notice that the winding number condition is always satisfied by this representation. Of course, from a practical standpoint, the choice of surfaces will have a significant impact on the truncation error constant. The most desirable surfaces would intersect orthogonally and be as flat as possible—unnecessarily curved choices will introduce additional curvature-dependent errors (all of which vanish as $\Delta t \to 0$).

We now direct our attention to the convergence analysis of our proposed algorithm.

4 Analysis of Diffusion-Generated Motion

In this section, we present formal analyses which show that the diffusion generated motion algorithm for filaments does produce a time discrete approximation to motion by vector mean curvature. We hope these non-rigorous arguments will encourage the development of rigorous convergence proofs, as they did in the case of diffusion generated motion by mean curvature for surfaces.

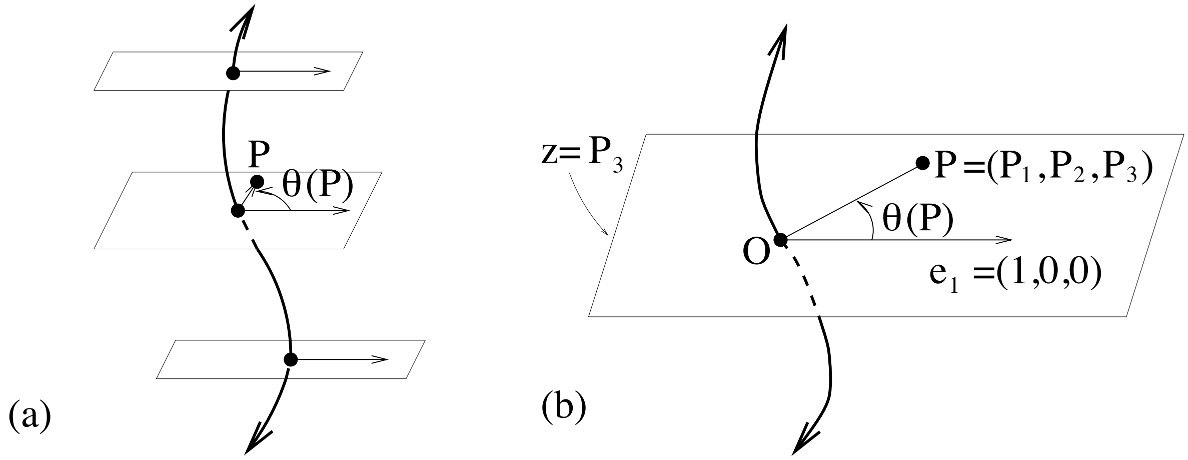
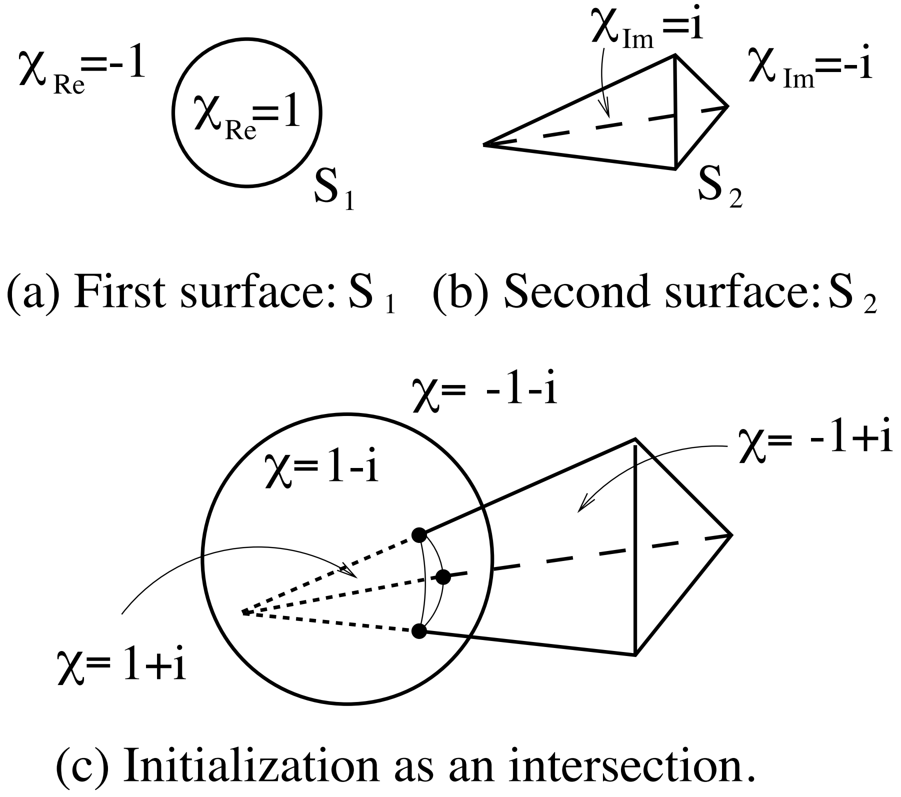


Figure 3: (a) In our examples, a plane and a reference axis are defined for each point on the filament. The initialization for a particular plane is then given by $\chi(\mathbf{P}) = \exp(i\theta(\mathbf{P}))$ where $\theta(\mathbf{P})$ is an angle function that winds around the filament. (b) Whenever the curve can be represented as a function of z in some coordinate system, this initialization step is particularly straightforward. For each grid point $\mathbf{P} = (P_1, P_2, P_3)$ we restrict ourselves to the plane $z = P_3$ and set $\mathbf{O}$ equal to the intersection of the plane with the curve. A consistent initialization is then obtained by setting $\chi(\mathbf{P}) = \exp(i\theta(\mathbf{P}))$ where $\theta(\mathbf{P})$ is the angle between $\mathbf{OP}$ and the fixed vector $\mathbf{e_1} = (1, 0, 0)$.



(a) First surface: S_1 (b) Second surface: S_2

(c) Initialization as an intersection.

Figure 4: Consider a filament which is the intersection of two surfaces. To initialize χ: (a) Set χ_{Re} equal to $+1$ inside the first surface and -1 outside. (b) Set χ_{Im} equal to $+i$ inside the second surface and $-i$ outside. (c) We initialize χ by summing the contributions from χ_{Re} and χ_{Im}. I.e., $\chi = \chi_{Re} + \chi_{Im}$.

We present two alternative approaches: heuristic analysis that uses a variety of shortcuts to deduce the motion law, and a detailed asymptotic analysis that yields the motion law.

4.1 Heuristic Analyses

Here we present two short, formal calculations which "show" that diffusion generates motion by vector mean curvature for a filament. These calculations allow us to quickly extract the motion law generated by diffusion, without going through the full details of asymptotic analysis. This is particularly useful for exploring novel diffusion-generated algorithms.

4.1.1 Surface Heuristics

We first briefly review each calculation for the easier case of diffusion generated mean curvature motion of surfaces [16, 17]. (Note that the distributional argument for surfaces given below has not been presented before.) We then show how the arguments extend to the more complicated case of filaments.

Consider a surface Γ, which is the boundary of an open set Ω. We represent Γ as the "zero level" of the characteristic function of the set,

$$\chi(\vec{x}) = \left\{ \begin{array}{rl} 1 & \text{if } \vec{x} \in \Omega \\ -1 & \text{otherwise} \end{array} \right. .$$

Our goal is to show that diffusion applied to this characteristic function will move its zero level with a normal velocity equal to the mean curvature, initially.

We can formally derive this directly from the diffusion equation

$$\chi_t = \nabla^2 \chi.$$

by suitable evaluation of the terms. There are two distinct approaches that yield simple, informative results. The first relies on writing the Laplacian out in a special geometric coordinate system. This has the advantage of providing a great deal of insight into why the method works, but becomes complicated in higher dimensions. The second approach is based on the formal calculus of distributions, which makes the calculations simple and mechanical in any number of dimensions, but is not as intuitively enlightening.

Geometric Coordinates for Laplacian The approach that is most direct and also provides the most insight is to write out the Laplacian in a coordinate system adapted to the local geometry of the surface, Γ. This is especially easy in the plane, i.e. for a 1-D curve Γ moving in R^2. In that case, suppose we want to determine the velocity $v_n(\vec{P})$ at a particular point $\vec{P} \in \Gamma$. To analyze the motion set up an (r, θ) polar coordinate system with origin at the center of the local circle of curvature of $\vec{P}$. Note that in this coordinate system, $\vec{P}$ is located at $r = R$, the radius of curvature at $\vec{P}$. Using the standard polar coordinate form of ∇^2 the diffusion equation becomes

$$\chi_t = \frac{1}{r}\chi_r + \chi_{rr} + \frac{1}{r^2}\chi_{\theta\theta}.$$

Because the polar coordinate circles are aligned (tangent to second order) with the level contours of χ initially, we have $\chi_{\theta\theta} = 0$ and (for short times) the equation becomes the purely one dimensional advection-diffusion equation

$$\chi_t - \frac{1}{r}\chi_r = \chi_{rr}$$

This evolution causes the initial radial step-function profile $\chi(r)$ to advect radially with the spatially variable speed $v = -1/r$, and simultaneously diffuse symmetrically into a hyperbolic tangent profile. Evaluating this at the point of interest $\vec{P}$ where initially $\chi = 0$ and $1/r = \kappa$ is the curvature of Γ, we get that initially the zero level of χ will move normal to itself at speed $v_n = -\kappa$, which is motion by mean curvature with the proper sign for a stable, well-posed motion. Note that the radial diffusion term does not effect the motion of the zero level initially, due to the symmetry of the step function $\chi(r)$ about its zero.

Note that this analysis shows that diffusion always induces an initial mean curvature motion in the contours of the diffusing function, as well as other effects that redistribute the function values. The special symmetry of the characteristic function causes all these other terms to vanish at its zero level. Thus diffusion generated motion relies on the generic short time behavior of diffusion and the special symmetry of the initial data to achieve its effect.

Distributional Calculus Expressing the Laplacian in geometric coordinates provides a great deal of insight, but it becomes complicated in higher dimensions and also requires a certain amount of pre-existing intuition to interpret the resulting formulas. There is an alternative approach that yields the result for a (codimension one) surface in any number of dimensions using only formal calculus. In this approach, we evaluate the terms of the diffusion equation as idealized distributional derivatives, since this captures the dominant singular behavior of the solution. To do this completely formally, we assume that χ is at all times the characteristic function of a set whose boundary $\Gamma(t)$ is moving with a normal velocity v_n. I.e. assume $\chi = H(d)$, where $H(x) = x/|x|$ is the sign function, and $d(\vec{x})$ is the signed distance to the moving interface $\Gamma(t)$, with sign such that $d > 0$ inside the set. To facilitate the calculations, we note the following relations. As distributions, $H' = 2\delta$, and $\delta(d)$ is a delta function concentrated on the interface $\Gamma(t)$. The unit normal vector field to $\Gamma(t)$ is given by $\hat{n} = -\nabla d$, the mean curvature is given by $\kappa = \nabla \cdot \hat{n}$, and the normal velocity is $v_n = d_t$. Further, these definitions naturally extend these quantities defined on $\Gamma(t)$ to all of space, where they are in general the normal, mean curvature and normal velocity of the level contours of d. We can now readily compute the distributional derivatives

$$\chi_t = 2v_n\delta,$$

$$\nabla\chi = -2\delta\hat{n},$$

and

$$\nabla \cdot \nabla\chi = -2\kappa\delta + 2\delta'.$$

Using these formulas, the diffusion equation becomes the relation

$$2v_n\delta = -2\kappa\delta + 2\delta'.$$

If we evaluate this relationship at the interface of interest, Γ, where $d = 0$, equating the coefficients of the delta functions gives

$$v_n = -\kappa$$

which is precisely motion by mean curvature of the surface (with the proper sign for stable, well-posed motion). Note that we are justified in neglecting the δ' term at $d = 0$ because $\delta'(0) = 0$, while $\delta(0) = \infty$, formally.

4.1.2 Filament Heuristics

In the case of filament motion, we have the curve initially represented by a complex valued function of the form

$$\chi(\vec{x}) = e^{i\theta(\vec{x})}$$

where $\theta(\vec{x})$ is a real angular coordinate on R^3 that increases by $2\pi m$ around any loop about the filament, where m is the nonzero integer winding number of χ. We identify the filament as the place where χ "vanishes", which in general means the locus of points in R^3 that χ winds around. Compare this to the case for surfaces above, where the "zero level" of the discontinuous characteristic function clearly meant the points in space at where the sign of χ changes.

Following the analysis for surfaces, we will deduce the effect that diffusion has on the location of the zero of χ by direct, formal evaluation of the diffusion equation

$$\chi_t = \nabla^2\chi.$$

In order to provide a clear intuition, we assume that χ winds around the filament uniformly[2] . Specifically, we assume that $\theta = m\phi(\vec{x}, t)$ where $\phi(\vec{x}, t)$ denotes the polar angle between $\vec{x}$ and the Frenet normal to the filament, in the plane normal to the filament that contains $\vec{x}$. Detailed asymptotics for initializations based on parallel planes (see Figure 3b) are given in Section 4.2.

Laplacian Heuristic Since we only need to track the motion of the zero of the complex χ, we can simplify the analysis slightly by working only with the amplitude of χ and ignoring the phase. Thus we write

$$\chi(\vec{x}, t) = A(\vec{x}, t)e^{i\theta(\vec{x},t)}$$

where $A = |\chi|$, and plug this form into the diffusion equation. The real part of this equation yields the amplitude evolution equation

$$A_t = \nabla^2 A - |\nabla\theta|^2 A.$$

[2] Heuristically, this assumption is reasonable since each diffusion step helps to enforce this type of symmetry near the zero of χ.

When viewed in this decomposition, we see there is a reaction term present that instantaneously drives the amplitude A to 0 at the location of the singularity of $|\nabla\theta|$, which in turn occurs at the center of winding of θ, i.e. at the filament location. (Note that the presumed winding of θ implies that $|\nabla\theta|$ blows up at the filament like m/d, where d is the distance to the filament.) Thus, as expected, the amplitude vanishes at the filament. This decomposition shows how this is enforced by the winding number.

All that remains is to write out the Laplacian in suitable geometric coordinates, and show that it has a term corresponding to advection with a velocity that reduces to $\vec{v} = \kappa\hat{n}$ at the filament, where κ is the curvature and $\hat{n}$ is the Frenet normal vector. Suitable coordinates can be defined as follows: let s be the arclength coordinate along the filament. At a given s value along the filament, there is a plane normal to the filament, and the Frenet normal $\hat{n}$ and binormal $\hat{b}$ unit vectors in this plane define associated Cartesian planar coordinates p, q. Thus (s, p, q) define an orthogonal curvilinear coordinate system (at least, near the filament). In this coordinate system, calculation (see [12] for partial details) shows that the the Laplacian is given by

$$\nabla^2 A = H[H[A]] - \frac{\kappa}{1-\kappa p}A_p + A_{pp} + A_{qq}$$

where $\kappa(s)$ is the curvature of the filament, and H is the differential operator

$$H[f] = \frac{1}{1-\kappa p}(f_s - \tau f_\phi)$$

where $\tau(s)$ is the torsion along the filament, and ϕ is the polar angle coordinate in the (p, q) plane. Thus the amplitude equation becomes becomes

$$A_t + \frac{\kappa}{1-\kappa p}A_p = H^2 A + A_{pp} + A_{qq} - |\nabla\theta|^2 A. \qquad (2)$$

Consider the short time effects of the terms of this equation: the singular reaction term drives A to 0 at the filament, and the (p, q) diffusion smoothes this profile into a cylindrically symmetric well. Because the resulting A is constant along the filament, and cylindrically symmetric, it has no s or ϕ dependence, and the H terms vanish. Thus, none of these terms actually produce any initial motion of the $A = 0$ location. The remaining terms in the equation, evaluated at the filament where $p = 0$, reduce to

$$A_t + \kappa A_p = 0$$

which convects the values of A in the (p, q) plane, in the p direction, with speed κ. Thus these terms move the zero of A—and hence the filament—by the vector mean curvature, initially. See the analysis related to Eq. (14) in Section 4.2 for further details on the asymptotic properties of the reaction-diffusion equation (2).

Distribution Heuristic Here we evaluate the terms in the filament diffusion equation as distributional derivatives. The distributional calculus is more delicate in the case of filaments, because the delta functions that result are concentrated only along certain directions. For example, let us compute the distributional gradient of χ. We will do this in the (s, p, q) coordinates defined in the previous analysis. Note that χ has no s dependence, so

$$\chi_s = 0.$$

Next, consider the behavior of χ along the p axis, which passes through the filament in the normal direction. Restricted to this line, χ is constant with the value $e^{i\theta(0)}$ along the $\phi = 0$ ray, and $e^{i\theta(\pi)}$ along the $\phi = \pi$ ray, so its derivative along this direction is a delta function,

$$\chi_p(p, q = 0) = [\chi]_n \delta(p)$$

where $[\chi]_n = e^{i\theta(0)} - e^{i\theta(\pi)}$ is the jump in χ at $p = 0$. Similarly, along the q axis, which passes through the filament in the binormal direction, we get

$$\chi_q(p = 0, q) = [\chi]_b \delta(q)$$

where $[\chi]_b = e^{i\theta(\pi/2)} - e^{i\theta(3\pi/2)}$. Along lines that do not pass through the filament, χ varies smoothly, which makes a smooth contribution to the p and q derivatives. Thus the complete distributional gradient in (s, p, q) coordinates is given by

$$\nabla\chi = [\chi]_n \delta(p) \hat{n}(s) + [\chi]_b \delta(q) \hat{b}(s) + O(\text{smooth})$$

where O(smooth) is the “smooth” (i.e. non-delta function, or non-singularly supported) part of the gradient. If we now take the divergence of this, and use that $\nabla \cdot \hat{n}(s) = -\kappa$, $\nabla \cdot \hat{b}(s) = 0$ at the filament (see [12]), we get

$$\nabla^2\chi = [\chi]_n \delta'(p) - \kappa[\chi]_n \delta(p) + O(\text{smooth})$$

Similarly, if we consider the initial time behavior of χ along the p axis, it is simply a step function propagating along at a speed v_n, so we get

$$\chi_t(p, q = 0) = -v_n[\chi]_n\delta(p),$$

while off this axis it varies smoothly in time, hence in general

$$\chi_t = -v_n[\chi]_n\delta(p) + O(\text{smooth}).$$

Thus the diffusion equation $\chi_t = \nabla^2\chi$ becomes

$$-v_n[\chi]_n\delta(p) = [\chi]_n\delta'(p) - \kappa[\chi]_n\delta(p) + O(\text{smooth}).$$

As for the case of surfaces, we equate the coefficients of the delta function terms to obtain

$$v_n = \kappa,$$

which is the desired motion by mean curvature.

4.2 Asymptotic Analysis of Diffusion Generated Motion

We next present a detailed matched asymptotic analysis of the ALGORITHM **CDGM** and show that the algorithm indeed captures the motion by curvature along filament normal direction. As a byproduct, we also obtain the order of convergence. We find both outer solution away from the filament core and inner solution taking account of the core structure before finally matching the two asymptotics. The inner solution reveals how the zero amplitude is generated without the Ginzburg-Landau nonlinearity, a conspired effort of linear diffusion and imposed topological winding number. The zero amplitude is what the numerical algorithm captures to follow the evolution of the filament. We also compare the behavior of the filament core in the algorithm with that of the complex Ginzburg-Landau equation (1). During the diffusive step, the filament core size enlarges in time like $O(\sqrt{\Delta t})$, while the Ginzburg-Landau filament core size remains $O(\epsilon)$ for all time.

Let us consider the diffusion effect on a complex scalar function of the form $\chi_0 = \exp\{i\Theta_0\}$, where Θ_0 is the phase function (counting the angle) about a space curve (the filament) Γ_0. Initially, Θ_0 is as described in Section 3.2, see also Fig. 3(b). Let us examine the effect of short time diffusion on

χ_0, especially its phase. Suppose the filament is parameterized by z, that is $\Gamma_0 : (\gamma_1(z), \gamma_2(z), z)$, then ($\vec{x} = (x, y, z)$, $\vec{\xi} = (\xi, \eta, \zeta)$):

$$\chi(t, \vec{x}) = (4\pi t)^{-3/2} \int_{R^3} \exp\{-|\vec{\xi} - \vec{x}|^2/4t\} \exp\left\{ i \arctan\left(\frac{\eta - \gamma_2(\zeta)}{\xi - \gamma_1(\zeta)}\right)\right\} d\vec{\xi}.$$

Changing variables: $\vec{\xi} = \vec{x} + \tau\vec{\xi'}$, and $\tau = \sqrt{t}$, we have (ignoring the primes):

$$\chi(t, \vec{x}) = (4\pi)^{-3/2} \int_{R^3} \exp\{-|\vec{\xi}|^2/4\} \exp\left\{ i \arctan\left(\frac{y + \tau\eta - \gamma_2(z + \zeta\tau)}{x + \tau\xi - \gamma_1(z + \zeta\tau)}\right)\right\} \; d\vec{\xi}. \tag{3}$$

If $|x - \gamma_1(z)| \geq \tau^\epsilon$, for $\epsilon \in (0, 1/4)$, we can expand the arctan function in small τ. Some details are:

$$\begin{aligned} \frac{y + \tau\eta - \gamma_2(z + \zeta\tau)}{x + \tau\xi - \gamma_1(z + \zeta\tau)} &= \frac{y - \gamma_2(z)}{x - \gamma_1(z)} + \tau\left(\frac{\eta - \gamma_2'(z)\zeta}{x - \gamma_1(z)} - \frac{y - \gamma_2(z)}{x - \gamma_1(z)} \cdot \frac{\xi - \gamma_1'(z)\zeta}{x - \gamma_1(z)}\right) + O(\tau^2) \\ &\equiv F_0 + \tau F_1 + O(\tau^2); \end{aligned} \tag{4}$$

and:

$$\begin{aligned} \exp\{i \arctan(F_0 + \tau F_1 + O(\tau^2))\} &= \exp\left\{ i \left(\arctan F_0 + \frac{\tau F_1}{1 + F_0^2} + O(\tau^2)\right)\right\} \\ &= \exp\{i \arctan F_0\} \left(1 + i\tau \frac{F_1}{1 + F_0^2} + O(\tau^2)\right) \end{aligned} \tag{5}$$

Note that F_0 is independent of $\vec{\xi}$ and F_1 is linear in $\vec{\xi}$, so when integrated against the heat kernel (now unit Gaussian) in (3), the $O(\tau)$ term vanishes, and we have:

$$\bar{\chi} = \chi(t, \vec{x})/|\chi(t, \vec{x})| = \exp\{i(\arctan F_0 + O(t))\}, \tag{6}$$

for small t and $|x - \gamma_1(z)| \geq t^{\epsilon/2}$, $\epsilon \in (0, 1/4)$. The $O(\tau^2) = O(t)$ term in (4) is quadratic in $\vec{\xi}$ and is not zero in general when integrated against the unit Gaussian, hence (6) is in fact optimal. It says that the small time diffusion on the phase is to introduce an $O(t)$ correction away from the filament.

We just derived this from a special coordinate system parameterized by z, however, the conclusion is independent of the choice of the coordinates. In fact, we will be mainly concerned with $\vec{x}$ in the vicinity or $O(t^{\epsilon/2})$ distance from the filament, so a local Frenet coordinate is more convenient. For small t, the form of the phase function away from the $O(t^{\epsilon/2})$ neighborhood of the filament is a higher order term, as seen from the fast decay of heat kernel. Hence local Frenet coordinate gives the same leading order behavior.

Now we look at $\chi(t,\vec{x})$ using Frenet coordinate attached on the filament and develop the inner asymptotic expression. Let us adopt the framework in [5] and define $\Gamma : \vec{X}(s,t) = (X,Y,Z)(s,t)$, where s is the arclength of Γ_0 at $t=0$. A space vector $\vec{x} = \vec{X} + r\hat{r}$, where $\hat{r} = \hat{r}(\theta,s,t)$, and $\hat{r}$ the radial unit vector on the Frenet plane spanned by $(\hat{n},\hat{b})$, the normal and binormal unit vectors. Let $\varphi = \hat{r}\cdot\hat{n}$, and $\theta_0 = \theta_0(s,t)$ obey: $\theta_{0,s} = -\sigma T$, where $\sigma = |\vec{X}_s|$, and $T = -\sigma^{-1}\hat{b}_s\cdot\hat{n}$, being the torsion of the filament. Let $\theta = \varphi - \theta_0$, then (r,θ,s) form orthogonal curvilinear coordinates, and:

$$d\vec{x} = \hat{r}dr + r\hat{\theta}d\theta + h_3\hat{\tau}ds,$$

where $\hat{\tau}$ the tangential unit vector, $h_3 = \sigma[1-\kappa r\cos(\theta+\theta_0)]$, and $\kappa = \sigma^{-1}|\hat{\tau}_s|$ the filament curvature.

The heat equation in the (r,θ,s) coordinates is:

$$\left[\frac{\partial}{\partial t} - \dot{X}\cdot\nabla_{r,\theta,s} - \frac{r}{h_3}(\hat{r}_t\cdot\hat{\tau})\frac{\partial}{\partial s} - (\hat{r}_t\cdot\hat{\theta})\frac{\partial}{\partial\theta}\right]\chi$$
$$= (rh_3)^{-1}\left[\frac{\partial}{\partial r}(rh_3\frac{\partial}{\partial r}) + \frac{\partial}{\partial\theta}(h_3r^{-1}\frac{\partial}{\partial\theta}) + \frac{\partial}{\partial s}(rh_3^{-1}\frac{\partial}{\partial s})\right]\chi \qquad (7)$$

where: $\nabla_{r,\theta,s} = \frac{\partial}{\partial r}\hat{r} + r^{-1}\frac{\partial}{\partial\theta}\hat{\theta} + h_3^{-1}\frac{\partial}{\partial s}\hat{\tau}$. The right hand side of (7) is equal to:

$$\left[\frac{\partial^2}{\partial r^2} + r^{-1}\frac{\partial}{\partial r} - \frac{\kappa\cos\varphi}{1-\kappa r\cos\varphi}\frac{\partial}{\partial r} + r^{-2}\frac{\partial^2}{\partial\theta^2} + \frac{\kappa\sin\varphi}{1-\kappa r\cos\varphi}r^{-1}\frac{\partial}{\partial\theta} + h_3^{-2}\frac{\partial^2}{\partial s^2} - \frac{h_{3,s}}{h_3}\frac{\partial}{\partial s}\right]\chi.$$

We look at a point x of $O(\delta)$ away from Γ, and expand χ as:

$$\chi \sim Ae^{iS} = (A_0 + \delta A_1 + \cdots)(\eta,\tau,s,\theta,t)e^{i(S_0+\delta S_1+\cdots)(\eta,\tau,s,\theta,t)}, \qquad (8)$$

where $\eta = r/\delta$, $\tau = t/\delta^2$, and the size of the small parameter δ will be determined later. The heat equation in (η,τ,s,θ) is:

$$\left[\delta^{-2}\frac{\partial}{\partial\tau} - \dot{X}\cdot\nabla_{\eta\delta,\theta,s} - \delta\eta h_3^{-1}(\hat{r}_t\cdot\hat{\tau})\frac{\partial}{\partial s} - \hat{r}_s\cdot\hat{\theta}\frac{\partial}{\partial\theta}\right]\chi$$

$$\begin{aligned} &= [\delta^{-2}(\frac{\partial^2}{\partial\eta^2} + \eta^{-1}\frac{\partial}{\partial\eta}) - \delta^{-1}\frac{\kappa\cos\varphi}{1-\delta\eta\kappa\cos\varphi}\frac{\partial}{\partial\eta} \\ &+ \delta^{-2}\eta^{-2}\frac{\partial^2}{\partial\theta^2} + \delta^{-1}\frac{\kappa\sin\varphi}{1-\delta\eta\kappa\cos\varphi}\eta^{-1}\partial_\theta + h_3^{-2}\frac{\partial^2}{\partial s^2} - \frac{h_{3,s}}{h_3}\partial_s]\chi. \qquad (9)\end{aligned}$$

Plugging (8) into (9) and keeping leading orders $O(\delta^{-2})$ and $O(\delta^{-1})$, we find:

$$\begin{aligned} \delta^{-2}(A_\tau + iS_\tau A) - \delta^{-1}\dot{X}\cdot\nabla_{\eta,\theta}A &= \delta^{-2}[\Delta A + 2i\nabla A\cdot\nabla S - |\nabla S|^2 A + iA\Delta S] \\ &+ \delta^{-1}[-\kappa\hat{n}\cdot\nabla A + iA\kappa\hat{n}\cdot\nabla S]. \qquad (10)\end{aligned}$$

Collecting imaginary and real parts, we have:

$$S_\tau - \Delta S + 2\frac{\nabla A}{A}\cdot\nabla S + \delta(\kappa\hat{n} - \dot{X})\cdot\nabla S = O(\delta^2), \qquad (11)$$

$$A_\tau - \Delta A + \delta(\kappa\hat{n} - \dot{X})\cdot\nabla A + |\nabla S|^2 A = O(\delta^2), \qquad (12)$$

with initial data for A being 1, and for S the angle variable θ. Here we suppose that the initialization can be expressed as local Frenet coordinates near the filament and that its phase is equal to θ. Otherwise, there is an initial layer during which the phase adjusts itself to θ. Notice that a small interval of t is magnified by δ^{-2} for τ, and so other phase initialization may well have relaxed to θ. The topological constraint on S is that its winding number about the origin is 1, also ∇S tends to zero at ρ infinity which helps to ensure the limit of A equal to one at ρ infinity. In (11)-(12), the coupling term is $\frac{\nabla A}{A}\cdot\nabla S$. To leading order, we have:

$$\begin{aligned} S_{0,\tau} - \Delta S_0 + 2\frac{\nabla A_0}{A_0}\cdot\nabla S_0 &= 0, \\ A_{0,\tau} - \Delta A_0 + |\nabla S_0|^2 A_0 &= 0, \qquad (13)\end{aligned}$$

which has solution $A_0 = A_0(|\eta|, \tau) = A_0(\rho, \tau)$, and $S_0 = \theta$. The coupling term is zero, and the reduced A_0 equation becomes:

$$A_{0,\tau} = A_{0,\rho\rho} + \frac{1}{\rho}A_{0,\rho} - \frac{1}{\rho^2}A_0, \qquad (14)$$

with $A_0(\rho, 0) = 1$. The dynamics of equation (14) is best understood in terms of its self-similar solution:

$$A_0(t,\rho) = A(\frac{\rho^2}{t}) \equiv A(z),$$

satisfying the ODE:

$$A_{zz} + (\frac{1}{z} + \frac{1}{4})A_z - \frac{1}{4z^2}A = 0, \tag{15}$$

with the boundary conditions: $A(z)$ is regular near $z \sim 0$, $A \to 1$ as $z \to +\infty$.

Expanding A for small z, we find two linearly independent local solutions:

$$A_1 = z^{1/2}(c_0 + c_1 z + \cdots), \;\; c_0 > 0, \tag{16}$$

and

$$A_2 = z^{-1/2}\,(b_0 + b_1 z + \cdots) + \log z\,(b'_0 + b'_1 z + \cdots), \quad b_0 > 0,$$

with the latter removed due to the regularity condition at $z = 0$.

Hence we see from (16) that the desired solution is strictly increasing in a small neighborhood of zero. By the maximum principle on positive solutions of equation (15), such a solution cannot experience an interior maximum, and so must be nondecreasing towards $z \to \infty$. Finite time blowup cannot occur due to boundedness of coefficients for z away from zero.

It remains only to analyze what limit A approaches as $z \to \infty$, a positive finite number or infinity. Making the change of variables:

$$A = e^{-\frac{1}{2}\log z - z/8}B,$$

we have:

$$B_{zz} - q(z)B = B_{zz} - (\frac{1}{64} + \frac{1}{8z})B = 0.$$

By a result of P. Hartman (p. 382, [9]), we have two linearly independent solutions:

$$B \sim q^{-1/4}\exp\{\pm\int^z \sqrt{q(s)}\,ds\}, \qquad z \to \infty,$$

where:

$$q^{1/2}(s) = \left(\frac{1}{64} + \frac{1}{8s}\right)^{1/2} \sim \frac{1}{8}\left(1 + \frac{4}{s} - \frac{8}{s^2} + \cdots\right),$$

and:

$$q^{-1/4}(s) = \left(\frac{1}{64} + \frac{1}{8s}\right)^{-1/4} \sim \frac{1}{\sqrt{8}}\left(1 - \frac{2}{s} + \frac{10}{s^2} + \cdots\right).$$

So the two linearly independent solutions are:

$$B_{1,2} \sim \left(1 - \frac{2}{z} + \frac{10}{z^2} + \cdots\right)\exp\{\pm\frac{1}{8}(z + 4\log z + \frac{8}{z} + \cdots)\}.$$

Or in terms of A:

$$\begin{aligned} A_1 &\sim (1 - \frac{2}{z} + \cdots)\exp\{\frac{1}{z} + \cdots\} = (1 - \frac{2}{z} + \cdots)(1 + \frac{1}{z} + \cdots) \\ &= (1 - 1/z + \cdots) = e^{-1/z}(1 + \alpha_2 z^{-2} + \cdots), \end{aligned} \tag{17}$$

and:

$$A_2 \sim e^{-\log z - z/4}(1 - \frac{2}{z} + \cdots)e^{\frac{8}{z}} = \frac{1}{z}e^{-z/4}(1 + \frac{6}{z} + \cdots).$$

Hence up to a multiplicative constant, the asymptotic behavior of A is that $A(z)$ converges to a finite positive constant as $z \to \infty$. We have $A'(z) \geq 0$, and in fact $A'(z) > 0$ for any finite z. Letting $w = A'(z)$, we see that w satisfies the differential inequality:

$$w_{zz} + (1/z + 1/4)w_z - (\frac{1}{z^2} + \frac{1}{4z^2})w = -\frac{1}{2z^3}A < 0,$$

implying via maximum principle that the nonnegative function w cannot achieve an interior minimum 0, thus $w > 0$.

We normalize A so that $A(+\infty) = 1$. By maximum principle, such a solution A ($A(0) = 0$, $A'(z) > 0$, $A(+\infty) = 1$) is the unique classical solution. This normalized self-similar solution is selected with initial condition 1 for equation (14).

With the order $O(\delta)$ terms turned on, the system (11)-(12) is coupled, however, the coupling tends to zero as $\rho \to \infty$ since $\nabla A \to 0$ and $A \to 1$. For $\tau \in [0, \tau_0]$, τ_0 a fixed positive number, as $\rho \to \infty$, the S approaches its steady state in τ denoted by S_∞ obeying the equation:

$$-\Delta S_\infty + \delta(\kappa\hat{n} - \dot{X}) \cdot \nabla S_\infty = O(\delta^2), \tag{18}$$

subject to the constraint that its winding number is one and also $\nabla S_\infty = o(1)$ as $\rho \to \infty$. Solution of (18) to $O(\delta)$ is:

$$S_\infty = \eta \int_0^\theta [-G_\eta + \delta(\kappa\hat{n} - \dot{X}) \cdot (\cos\theta, \sin\theta)G]\, d\theta, \tag{19}$$

where:

$$G = -\exp\{\delta(\kappa\hat{n} - \dot{X}) \cdot (\eta\cos\theta, \eta\sin\theta)\}K_0(\delta\eta|\kappa\hat{n} - \dot{X}|/2), \tag{20}$$

with $K_0(x) = -\log\frac{x}{2} - const. + \cdots$, K_0 the zeroth order modified Bessel function, see [19].

Combining (19)-(20), we calculate:

$$\begin{aligned} S_\infty &= \theta + \delta(\kappa\hat{n} - \dot{X})(\eta\sin\theta, \eta(1-\cos\theta))K_0(\delta\eta|\kappa\hat{n} - \dot{X}|/2), \\ &= \theta + r(\kappa\hat{n} - \dot{X})(\sin\theta, (1-\cos\theta))K_0(|\kappa\hat{n} - \dot{X}|r/2). \end{aligned} \tag{21}$$

Now we select δ to be fixed small number $\geq O(t^{\epsilon/2})$. Consequently, $r = O(\delta)$; the phase corrections due to the above two calculations (6) and (21) match to give the relation:

$$|\kappa\hat{n} - \dot{X}|\; K_0(|\kappa\hat{n} - \dot{X}|/2) = O(t), \tag{22}$$

implying:

$$\dot{X} - \kappa\hat{n} = O\left(\frac{t}{\log(t^{-1})}\right), \tag{23}$$

for small t. We have shown that after the first diffusing step of the algorithm, the filament motion is motion along normal direction by curvature to the leading order.

At the subsequent normalizing step, $\bar{\chi} = e^{i\Theta}$, with $\Theta = \theta + O(\delta t)$, δt the time step of the algorithm, locally in the Frenet coordinate near the filament Γ. This follows from the inner solution structure as shown in (21). As commented before, a similar outer solution calculation again infers that the phase correction at the following diffusing step is $O(\delta t)$ to leading order. Iterating this argument, we see that the ALGORITHM **CDGM** of Section 3.2 captures the leading filament motion law of the Ginzburg-Landau equation, that is motion by curvature along the normal [19].

4.3 Comparison with Ginzburg-Landau Filaments

The algorithm in Section 3.2 mimics the action of the complex Ginzburg-Landau (CGL) nonlinearity (which is stiff numerically) by repeated normalizing steps (2a). The diffusing steps (2b) recovers the quantity $\kappa\hat{n} - \dot{X}$; however, the inner solution, especially the amplitude, is time dependent; and the core size increases in time. In contrast, the inner solution of the CGL is quasi-steady (only depending on r/δ), and the filament core size remains $O(\epsilon)$ all the time.

The diffusive aspect of the algorithm and that of the CGL do share some common features. The CGL phase obeys linear diffusion equation away from the filament core region ([19], [14]), which makes the phase change by $O(t)$ for small time, similar to the algorithm.

The phase singularity (zero amplitude) in the algorithm is generated by the imposed phase winding number and the linear diffusion. In CGL, the stiff nonlinearity is a major source of zero amplitude. It remains to find out how to extend the approach to model a phase singularity in the Schroedinger filaments (motion by curvature along binormal). A straightforward use of the linear equation $iu_t = \Delta u$ only produces oscillation near the core region.

5 Numerical Experiments

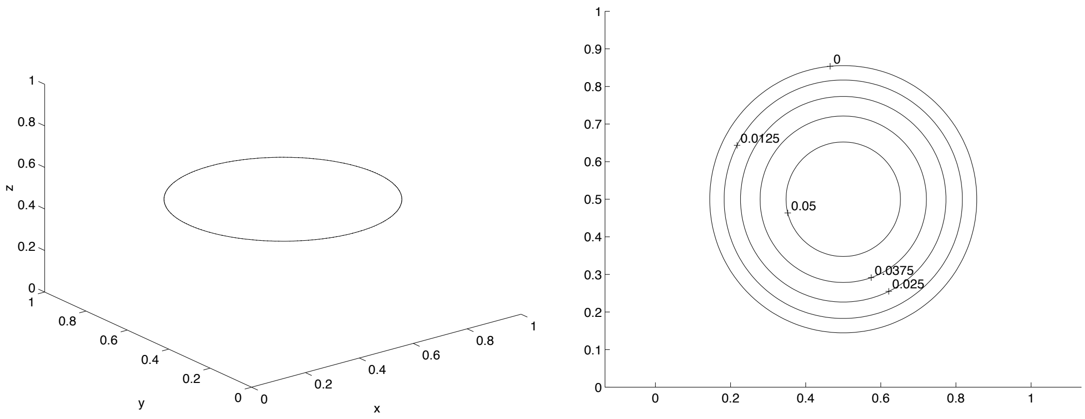
In this section, we report on various experiments using our algorithm. For simplicity, all results are derived using a pseudospectral spatial discretization (see, e.g., [24]). More accurate results may be obtained using adaptive resolution with fast Fourier transform techniques. See [22] for a detailed discussion on these methods for the ALGORITHM **DGM**.

5.1 A Ring

To begin, consider the curvature motion of the circular ring displayed in Fig. 5.

Since a circle may be represented as the intersection of a plane with a cylinder, it is clear that we may initialize χ according to the shape-based method described in Section 3.3. Specifically, we set χ_{Re} equal to $+1$ inside and -1 outside the cylinder and χ_{Im} equal to $+i$ below and $-i$ above the plane. This gives the initial value of χ as a sum $\chi = \chi_{Re} + \chi_{Im}$ as shown in Fig. 6a. Using this initialization, with a time step size of $\Delta t = 0.0001$ and a mesh spacing of $\Delta x = 1/128$ we obtain a close agreement with exact solution for curvature motion. Indeed, the relative error in the change of area enclosed by the ring is always less than 4% in this calculation. See Fig. 6b.

An alternative initialization appropriate for generalization to interlinked rings is also easily derived (see Fig. 7a): For each grid point $\mathbf{P}$ in the domain, the nearest point $\mathbf{O}$ to the ring is determined. The initial value of χ is then given by $\exp(i\theta(\mathbf{P}))$ where $\theta(\mathbf{P})$ is the angle between $\mathbf{OP}$ and the outward



(a) Ring in 3D. (b) The cross section $z = 0.5$.

Figure 5: (a) Initial ring. (b) Cross sectional contour plot of the ring shrinking under curvature motion at various times t. Note that the exact solution for this problem is easily obtained using the fact that the radius of the circle obeys the ordinary differential equation, $\dot{r} = -2\pi/r$.

normal at **O**. Once again, a close agreement with the exact solution for curvature motion is obtained. For example, the relative error in the change of area enclosed by the ring is always less than 4% when $\Delta t = 0.0001$ and $\Delta x = 1/128$. See Fig. 7b.

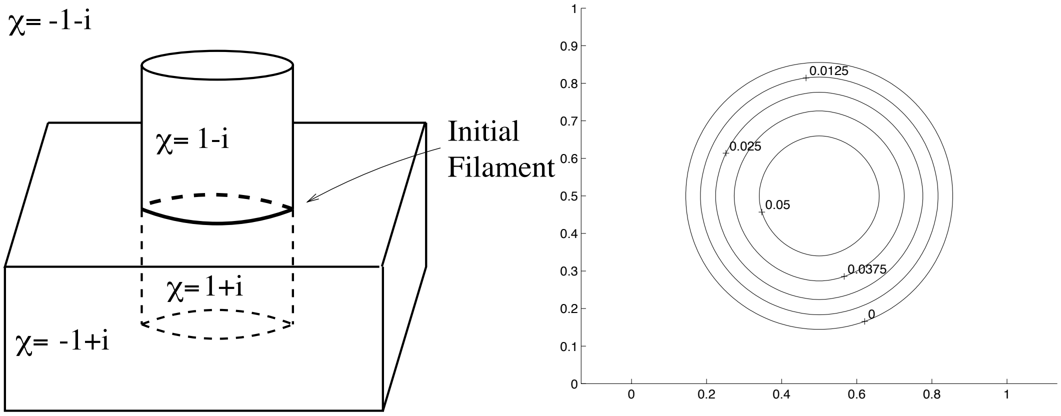
5.2 A Spiral

For our second example, consider the curvature motion of a periodic spiral,

$$\begin{aligned} x &= 0.5 + 0.3\cos(2\pi s), \\ y &= 0.5 + 0.3\sin(2\pi s), \\ z &= s. \end{aligned}$$

Here, it is easily shown that the exact solution is also a spiral, but with a radius that shrinks according to the ordinary differential equation, $\dot{r} = -r/(r^2 + \frac{1}{4\pi^2})$.

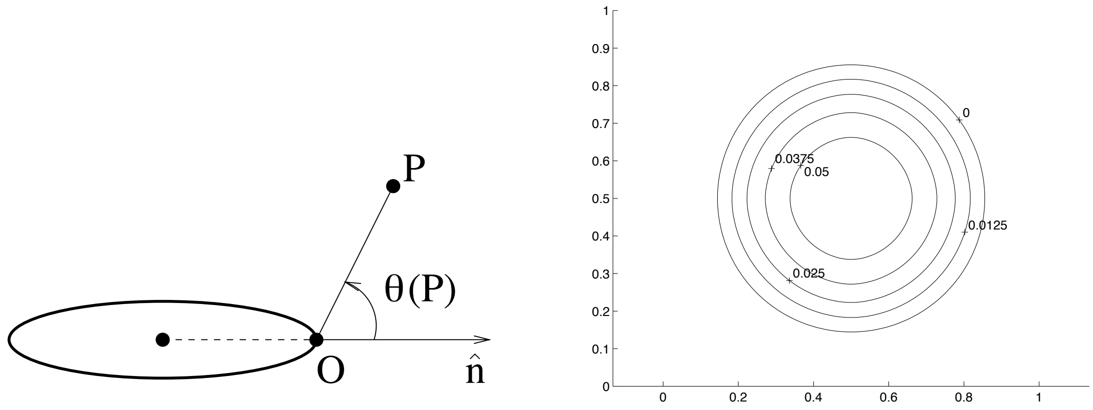
Since this spiral is naturally represented as a function of z, it is straightforward to initialize χ according to the method described in Fig. 3b. Using



(a) Initialization.

(b) Numerical result.

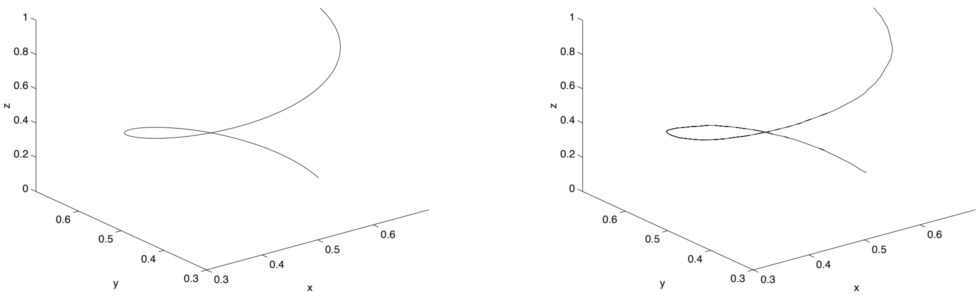
Figure 6: (a) Because a circle can be represented as the intersection of a cylinder with a plane, it is straightforward to initialize a ring: Simply set χ_{Re} equal to $+1$ inside and -1 outside the cylinder and χ_{Im} equal to $+i$ below and $-i$ above the plane. The initial value of χ is then given by a sum: $\chi = \chi_{Re} + \chi_{im}$. In our simulation, a cylinder with a radius of 0.35 was used to produce a ring with the desired initial radius. (b) Numerical result at various times t using a time step size of $\Delta t = 0.0001$ and a mesh spacing of $\Delta x = 1/128$.



(a) Initialization.

(b) Numerical result.

Figure 7: (a) Initialization of χ for a ring. For each grid point $\mathbf{P}$, the nearest point $\mathbf{O}$ to the ring is determined. A consistent initialization is then obtained by setting $\chi(\mathbf{P}) = \exp(i\theta(\mathbf{P}))$ where $\theta(\mathbf{P})$ is the angle between $\mathbf{OP}$ and the outward normal at $\mathbf{O}$. (b) Numerical result at various times t using a time step size of $\Delta t = 0.0001$ and a mesh spacing of $\Delta x = 1/128$.



(a) Exact solution.

(b) Numerical result.

Figure 8: A periodic spiral (initial radius of 0.3) evolved with a normal velocity equal to curvature: (a) Exact solution. Note that the exact solution for this problem is easily obtained using the fact that the radius of the spiral obeys the ordinary differential equation, $\dot{r} = -r/(r^2 + \frac{1}{4\pi^2})$. (b) Numerical result derived using the ALGORITHM **CDGM**.

this initialization, the location of the spiral filament was approximated over a time $t = 0.05$ using a time step size of $\Delta t = 0.000025$ and a mesh spacing of $\Delta x = 1/128$. As in the case of a shrinking ring, the ALGORITHM **CDGM** gives a very close agreement with the exact solution. See Fig. 8.

5.3 Connected Rings

For our next example, consider the curvature motion of two interlinked rings. Here, the rings shrink and eventually merge to form a closed loop, as shown in Fig. 10. In this example, we take our "exact solution" to be a front tracking calculation with 200 nodes along the curve. The change of topology was chosen to agree with the optimal curve shortening solution and was verified with a simulation of the complex Ginzburg-Landau equation.

To initialize χ, we determine the nearest ring and assign a phase angle θ exactly as in the case of a single ring. This phase angle is then shifted by an amount α according to the relative position of the distal ring. See Fig. 9.

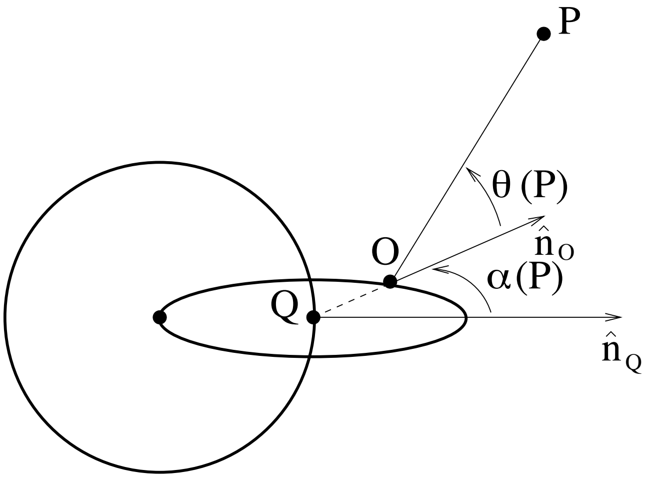


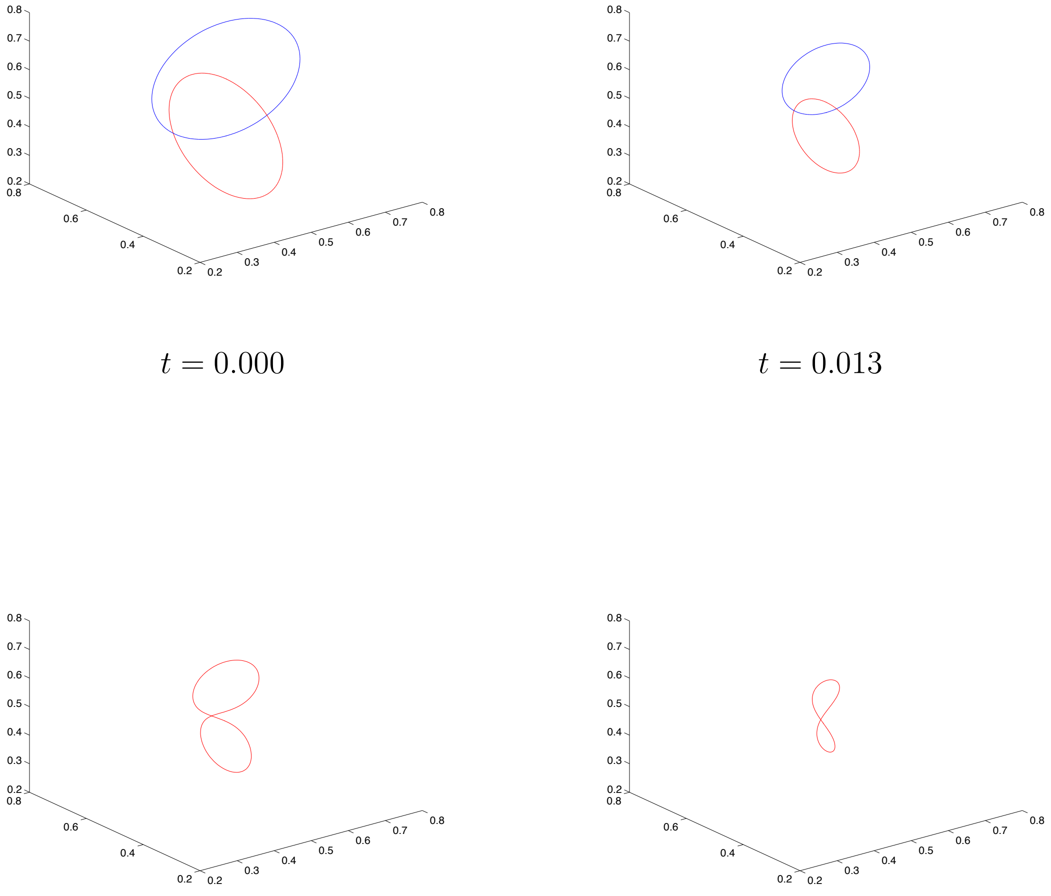
Figure 9: Initialization of χ for connected rings. Similar to the case of a single ring, the nearest point **O** to the closest filament is determined for each grid point **P**. The value of $\theta(\mathbf{P})$ is then given by the angle between **OP** and the outward normal at **O** (denoted $\hat{n}_O$). Letting **Q** be the nearest point on the distal ring and $\hat{n}_Q$ be the outward normal at **Q**, a consistent initialization is obtained by setting $\chi(\mathbf{P}) = \exp(i\theta(\mathbf{P}) + i\alpha(\mathbf{P}))$ where $\alpha(\mathbf{P})$ is the angle between $\hat{n}_O$ and $\hat{n}_Q$.

Using this initialization, the location of the filament was approximated over a time $t = 0.02$ using a time step size of $\Delta t = 0.00002$ and a mesh spacing of $\Delta x = 1/256$. As shown in Fig. 8, the ALGORITHM **CDGM** gives a very good agreement with the exact solution. Notice, in particular, that the method automatically selects the correct topological change and that (unlike level set formulations) the filaments do not develop interiors.

5.4 A Large System

In our final example, we simulate a large system of randomly generated filaments.

To initialize the system, a random phase angle $\theta \in [0, 2\pi)$ was assigned to blocks of size $0.2 \times 0.2 \times 0.2$. Using this initialization (i.e., with $\chi = \exp(i\theta)$),



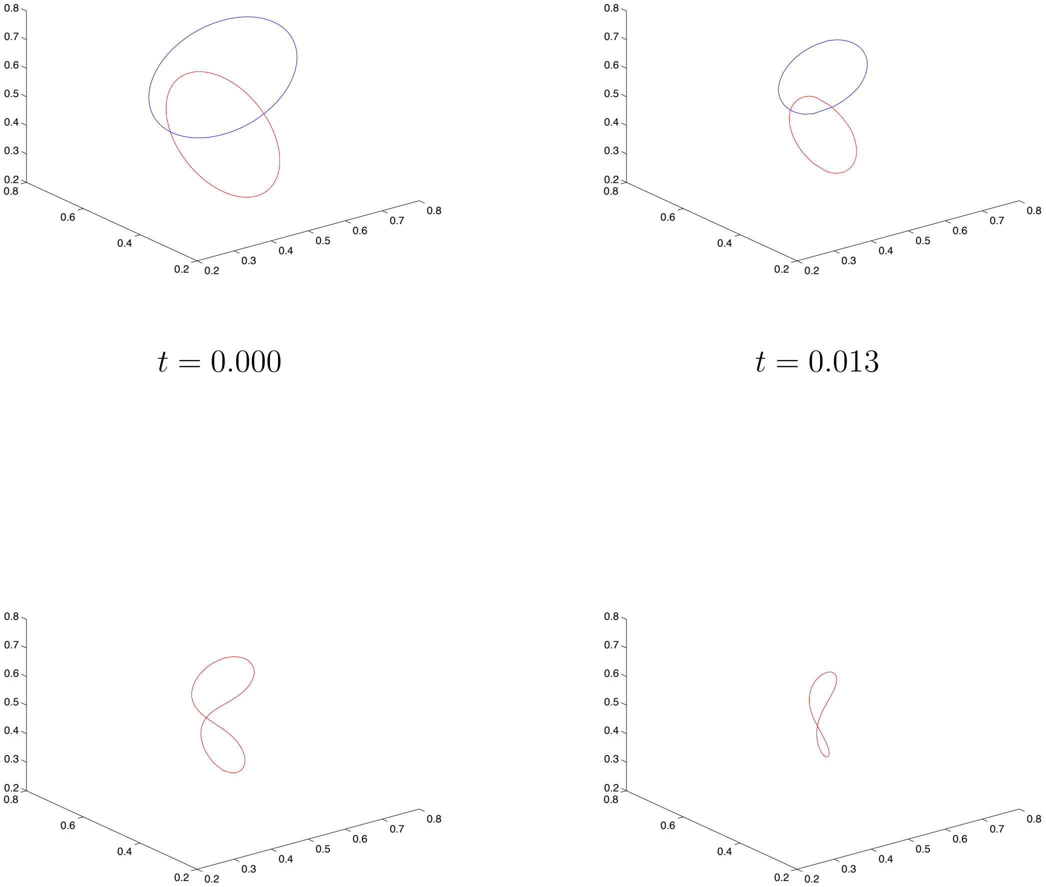
$t = 0.000$

$t = 0.013$

$t = 0.016$

$t = 0.020$

Figure 10: Two connected rings moving by curvature motion (exact solution).



$t = 0.000$

$t = 0.013$

$t = 0.016$

$t = 0.020$

Figure 11: The diffusion-generated motion of two (initially) connected rings.

the motion of the filaments was approximated over a time $t = 0.02$ using a time step size of $\Delta t = 0.0001$ and a mesh spacing of $\Delta x = 1/128$. As shown in Fig. 12, the ALGORITHM **CDGM** produces a smoothing and shortening of filaments as time progresses. Note also that topological shape changes are automatically captured and that the filaments do not develop interiors.

6 Generalizations and Variations

Here we discuss possible generalizations, variations and alternative formulations for diffusion generated motion of filaments. Of particular interest is generalizing it to obtain "motion by vector mean curvature" for objects of any dimension and codimension.

For clarity, let us briefly recall the meaning of vector mean curvature. For a k dimensional object Γ inside of R^d, the mean curvature has an obvious meaning as a scalar—i.e. at any point on the object it is the sum of the principal curvatures. However, it also has a natural direction associated with it. To describe this in concrete terms, recall that passing through any point $\vec{p} \in \Gamma$ are geodesic curves $\vec{\sigma_i}, i = 1, \ldots, k$, such that

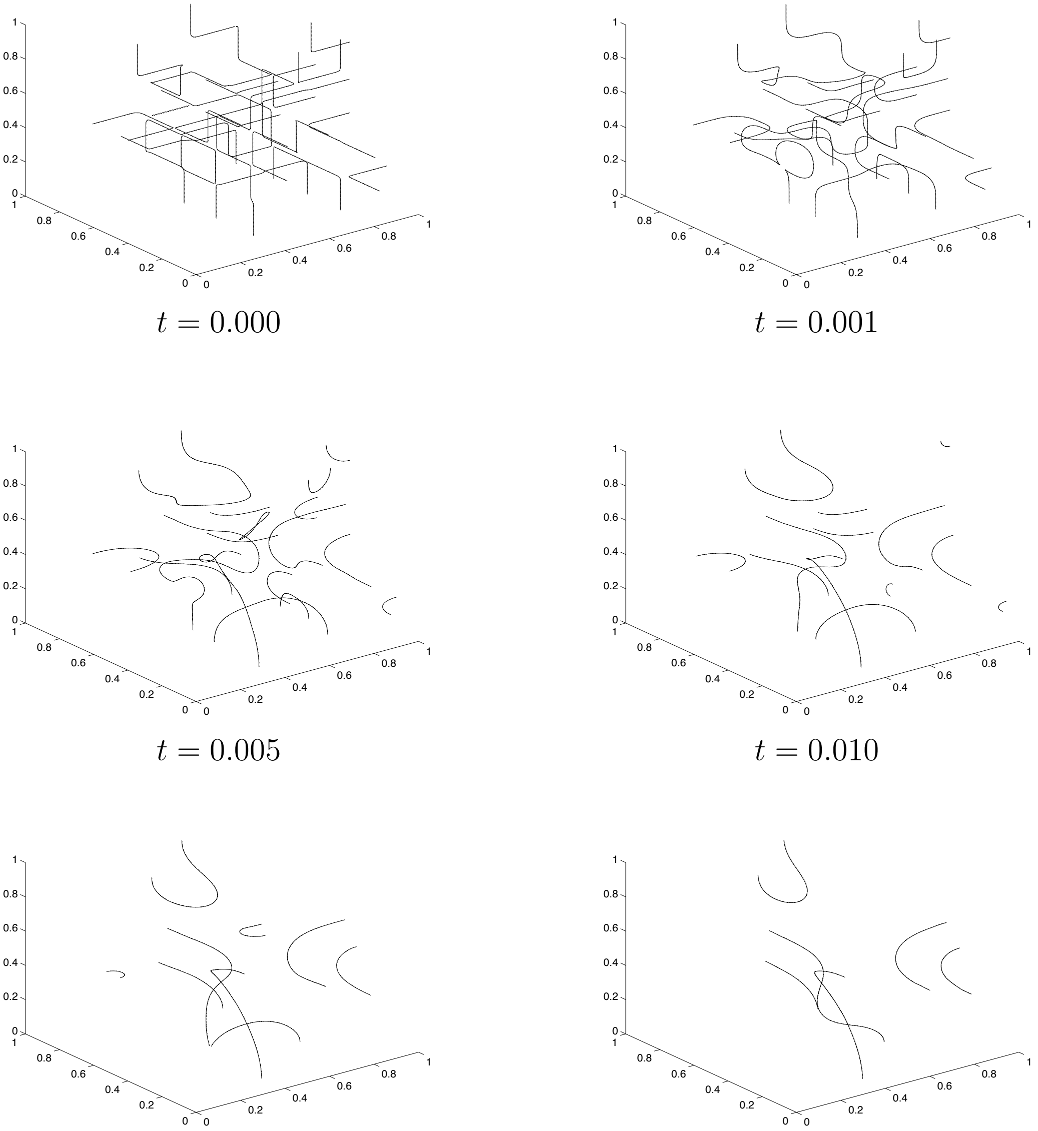
$$(\vec{\sigma_i})_{ss} = \kappa_i \hat{n}_i$$

where s denotes arc length, and $\kappa_i, i = 1, \ldots, k$ are the principle curvatures at $\vec{p}$. Here $\hat{n}_i$ lie in the $d - k$ dimensional normal space to Γ at $\vec{p}$. The vector mean curvature $\kappa\hat{n}$ is simply the the sum of these principle curvatures vectors,

$$\kappa\hat{n} = \sum_{i=1}^{i=k} \kappa_i \hat{n}_i .$$

Alternatively, we can directly interpret "motion by mean curvature" for Γ to be the motion produced by gradient descent minimization of its (k-dimensional) "surface area". The resulting normal velocity field induced on Γ by this flow is the vector mean curvature, by definition. Thus the vector mean curvature is simply the variational derivative of the surface area of Γ (with respect to Γ).

Diffusion Generated Motion for Codimension 2 Filaments The diffusion generated motion algorithm for filaments in R^3 has an immediate ex-



$t = 0.000$

$t = 0.001$

$t = 0.005$

$t = 0.010$

$t = 0.015$

$t = 0.020$

Figure 12: The diffusion-generated motion of a system of filaments. Here, zero flux boundary conditions are assumed.

tension to moving objects of dimension $d-2$ inside of R^d, i.e. arbitrary codimension 2 "filaments". The evolution ALGORITHM **CDGM** remains unchanged, and the initialization procedures described in Section 3.3 also have direct extensions to this setting. The only difference is that now $\chi : R^d \to R^2$ for some $d \geq 3$, and the linear diffusion step is carried out on R^d.

We conjecture that this extension of diffusion generated motion produces motion by vector mean curvature of codimension 2 filaments in any number of dimensions. Formal analysis similar to those presented in Section 4 should be possible, by suitably extending the usual Frenet coordinate system to the general codimension 2 case. Heuristic support for this conjecture is also contained in the work of Lin [14], which shows that the singular strong reaction limit of the complex Ginzburg-Landau reaction diffusion equations on R^d produce codimension 2 filamentary structures moving by vector mean curvature.

Diffusion Generated Motion for Arbitrary Codimension Filaments The diffusion generated motion algorithm for filaments in R^3 further has a natural extension to moving objects of arbitrary dimension k inside of R^d, i.e. arbitrary codimension $d-k$. We simply take $\chi = (\chi_1, \ldots, \chi_{d-k})$ to be a $d-k$-vector valued function on R^d that represent a k dimensional "filament" Γ as its "center of winding". More specifically, this means that for any $d-k-1$ dimensional sphere S that loops around Γ in R^d, the restricted mapping $\chi : S \to R^{(d-k)}$ has nonzero degree m about the origin in $R^{(d-k)}$. The evolution ALGORITHM **CDGM** remains formally unchanged, consisting in general of linear diffusion on R^d, followed by renormalization of χ to a unit vector. The initialization methods discussed in Section 3.3 can also be generalized to this setting. For example, the shape based method described there extends quite directly: if we have Γ represented as the intersection of the zero level surfaces of $d-k$ smooth functions $\phi_i : R^d \to R, i = 1, \ldots, d-k$, then the mapping

$$\chi = (H(\phi_1), \ldots, H(\phi_{d-k})),$$

where $H(x) = x/|x|$ is the sign function, suitably "winds around" Γ with degree 1.

Note that this general form of the filament algorithm also reduces to the original diffusion generated motion algorithm [16] in the case of codimension 1. Thus it constitutes a generalization of the original algorithm to arbitrary codimension, as we desired. It is interesting to see that the familiar representation of a surface as a $\chi = 0$ level set used there is actually the trivial case of using the topological degree of a vector field. This provides insight into how to extend level set methods to objects of arbitrary codimension, which we will follow up in subsequent work.

We conjecture that this generalized diffusion generated motion algorithm produces motion by vector mean curvature for objects of arbitrary codimension. It seems that the formal analyses presented in Section 4 could be generalized in the direction required to study this motion, although we have not done so. The key ingredient for this would be writing the Laplace operator in coordinates suitably adapted to the local geometry of Γ, so as to bring out its explicit dependence on the vector mean curvature. Nor have we carried out numerical investigations of this conjecture, due to the high cost of doing even the simplest new case (which requires diffusion of a 2-vector field on R^4 to move a 2-D surface in R^4).

We are not aware of any analogous result (rigorous or formal) for the Ginzburg-Landau equation, although it seems like a natural conjecture based on what has been shown for the codimension 1 and 2 cases [4, 14]. Specifically, we would expect that the Ginzburg-Landau evolution equation applied to a $d-k$-vector valued field on R^d that winds around a k dimensional filament Γ to capture vector motion by mean curvature of Γ in the strong reaction limit.

A Geometric Filament Representation In the approach presented here, we applied diffusion to an implicit *topological* representation of the filament. Specifically, we used a 2-vector valued field whose center of winding in R^2 was the image of the filament. Note that such 2-vector χ values are not vectors in R^3, and so they have no natural geometric relation to the filament, such as "pointing away from it". In contrast, we could represent the filament by a field of 3-vectors in R^3 that point radially away from it. The model for such a radial vector field is the unit vector field $n = -\nabla d$, where $d(\vec{x})$ is the distance from $\vec{x}$ to the filament. The diffusion and re-normalization processes in ALGORITHM **CDGM** both apply to such a radial vector field, and this provides a somewhat different approach to diffusion generated filament motion, in which we can at all times identify the filament geometrically as "the set the vector field points away from". Note this geometric variant of diffusion generated motion also makes sense for objects of any codimension, since the unit vector field pointing away from the object, $n = -\nabla d$, is always defined.

For the initial timestep, our previous heuristic and asymptotic analysis from Section 4 suggests diffusion of this "radial vector field" will displace the associated filament in proportion to its vector mean curvature. However, it is not clear that the renormalization step will produce a new vector field that points away from the displaced filament in the same radially symmetric fashion as $-\nabla d$, and thus its not clear that the next step of diffusion will produce further motion by mean curvature. Determining the limiting motion law (if any) from this geometric approach is an interesting topic for future research.

Note the geometric approach is no longer directly analogous to the complex Ginzburg-Landau PDE model for filaments, although it does suggest an analogous formulation in that setting. Specifically, if the Ginzburg-Landau evolution equation were applied to a 3-vector field on R^3 that pointed radially away from a filament, presumably the strong reaction asymptotic limit would be a unit vector field pointing radially away from that filament moving by vector mean curvature. We are not aware of any numerical or theoretical studies of this possibility, however.

There is another PDE analog in the work of Ambrosio and Soner [1]. They present an approach to mean curvature motion based on a level set evolution equation applied to the squared distance to the object (of any codimension), $d(\vec{x})^2$. As a part of their investigation, they show that "near the object"

$$\vec{\eta}_t = \nabla^2 \vec{\eta}$$

where $\vec{\eta} = -\nabla d^2/2 = d\hat{n}$. Thus they show that mean curvature motion of the object corresponds to a diffusion of this geometric radial vector field. This suggests (but by no means proves) that if we apply diffusion to the unit vector field pointing away from an object, $\hat{n}$, we might also move the object so defined by vector mean curvature.

Other Velocity Laws The original diffusion generated motion by mean curvature of algorithm can be generalized to obtain to obtain a great variety of other motion laws [16, 10, 23, 20, 11]. The basic idea behind these generalizations is that diffusive evolution is just convolution with a Gaussian kernel, so we could instead carry out linear convolution with some other kernel to generate the basic motion. In addition, instead of locating the updated object as the set where $\chi = 0$, we could locate it as the set where $\chi = \lambda$, for some given λ in the range of χ. This same approach can be applied to the filament formulation. Specifically, we can define the Convolution Generated Motion of a filament as follows: let K be a convolution kernel, $K : R^3 \to R$, and let λ be a complex number, of norm $|\lambda| < 1$. Starting from a normalized $\chi : R^3 \to R^2$ representing a filament as its center of winding, we update by convolution

$$\bar{\chi}(\vec{x}) = K * \chi(\vec{x}) = \int_{R^3} K(\vec{x} - \vec{y})\chi(\vec{y})d\vec{y}$$

followed by renormalization about the value λ,

$$\chi = \frac{\bar{\chi} - \lambda}{|\bar{\chi} - \lambda|}$$

which represents the updated filament as its center of winding.

For the case of surface motion, this general approach has been used to generate constant normal motion, $v_n = 1$, constant plus curvature motion, $v_n = a + b\kappa$, anisotropic versions of this, $v_n = a(\hat{n}) + b(\hat{n})\kappa$, and even nonlocal motion laws such as volume preserving motion by mean curvature. It has also been used to generate pattern formation and dynamics that has traditionally been described by reaction-diffusion or cellular automata models [24].

We expect that similarly general motions can be achieved for filaments. In particular, for a filament the local velocity will have both normal and binormal components,

$$\vec{v} = v_n\hat{n} + v_b\hat{b},$$

and it would be quite interesting to know what family of such motions can be obtained with a particular kernel K and a fixed normalization threshold λ, similar to the rigorous characterization presented in [11] for surface motion.

7 Summary and Topics for Future Research

In this work, we have presented a diffusion-generated approach for the curvature motion of filaments that automatically captures topological mergers with no special algorithmic procedures. We have also provided formal arguments for the convergence for our proposed method and validated our results with numerical experiments that include topological change.

A variety of interesting computational aspects related to our algorithm are still unexplored. Note, in particular, that we utilized a pseudospectral spatial discretization for the numerical experiments presented in Section 5. Although simple, this approach is inefficient because it does not provide for subgrid resolution or local refinement. For the codimension-one case, however, efficient discretizations based on adaptive resolution with fast Fourier transforms have been developed [22]. We expect that these same methods can be applied to surfaces in arbitrary codimension by replacing the exact integration used in [22] with appropriate quadrature steps. These fast methods are the subject of ongoing research.

We presented heuristic and formal derivations of the diffusion generated motion law for filaments in Section 4. A rigorous proof of convergence—both for the special case of a filament in R^3 and for the case of filaments of arbitrary codimension described in Section 6—would be of great interest. Further, numerical experiments suggest that in the presence of filament mergers, diffusion generated motion gives the "optimal curve shortening" filament evolution. A proof of this observation is desirable as well.

We also outlined the generalizations to arbitrary convolution generated filament motion in Section 6. It would be interesting to give specific realizations (convolution kernel K and normalization threshold λ) for filament motions of interest, such as constant normal motion, constant binormal motion, motion by the vector torsion, length preserving motion by mean curvature, etc.. It would also be quite interesting to classify what filament velocity laws are attainable with a fixed kernel and threshold.

Finally, we noted that the method of alternately diffusing and normalizing can be motivated by a formal operator splitting of the Ginzburg-Landau equations. Yet phase-field models cannot always be reduced in this way—i.e. the associated diffusion generated motion does not always produce a convergent discrete time approximation to the $\epsilon \to 0$ singular limit of the PDEs. For example, we found that the filament motion derived using a complex diffusion coefficient in the ALGORITHM **CDGM** does not agree with the solution to the corresponding Ginzburg-Landau equation (which is a nonlinear Schroedinger equation). A particularly interesting and important meta-problem is to determine in general when a phase field model has the same singular limiting behavior as its diffusion generated motion analog.

8 Acknowledgments

We thank Fang-Hua Lin for helpful discussions on complex Ginzburg-Landau equations, and for the series of lectures he gave at UCLA which inspired the present work; J. Xin would like to thank the computational and applied math faculty and staff (CAM) at UCLA for the hospitality, and support during his visit.

References

[1] L. Ambrosio and H.M. Soner. Level set approach to mean curvature flow in arbitrary codimension. *J. of Differential Geometry*, 43(4):693–737, 1996.

[2] G. Barles and C. Georgelin. A simple proof of convergence for an approximation scheme for computing motions by mean curvature. *SIAM Journal on Numerical Analysis*, 32(2):484–500, 1995.

[3] G. Bellettini, M. Novaga, and M. Paolini. An example of three dimensional fattening for linked space curves evolving by curvature. *Comm. Partial Differential Equations*, In Press.

[4] L. Bronsard and R.V. Kohn. Motion by mean curvature as the singular limit of Ginzburg-Landau dynamics. *Journal of Differential Equations*, 90(2):211–237, 1991.

[5] A. Callegari and L. Ting. Motion of a curved vortex filament with decaying vortical core and axial velocity. *SIAM J. Appl. Math*, 35:148–175, 1978.

[6] S.J. Chapman and G. Richardson. Motion of vortices in type II superconductors. *SIAM J. Appl. Math.*, 55(5):1275–1296, 1995.

[7] L.C. Evans. Convergence of an algorithm for mean curvature motion. *Indiana University Mathematics Journal*, 42:553–557, 1993.

[8] L.C. Evans, H.M. Soner, and P.E. Souganidis. Phase transitions and generalized motion by mean curvature. *Communications on Pure and Applied Mathematics*, 45(9):1097–1123, 1992.

[9] P. Hartman. *Ordinary Differential Equations.* Wiley and Sons, 1973.

[10] H. Ishii. A generalization of the Bence, Merriman and Osher algorithm for motion by mean curvature. In A. Damlamian, J. Spruck, and A. Visintin, editors, *Curvature Flows and Related Topics*, pages 111–127. Gakkôtosho, Tokyo, 1995.

[11] H. Ishii, G.E. Pires, and P.E. Souganidis. Threshold dynamics type schemes for propagating fronts. *TMU Mathematics Preprint Series*, 4, 1996.

[12] J.P. Keener and J.J. Tyson. The dynamics of scroll waves in excitable media. *SIAM Review*, 34:1–39, 1992.

[13] Y. Kuramoto. *Chemical Oscillations, Waves, and Turbulence.* Springer-Verlag, 1980.

[14] F-H Lin. Complex Ginzburg-Landau equations and dynamics of vortices, filaments and codimension 2 submanifolds. *Comm. Pure Appl. Math*, 51:385–441, 1998.

[15] P. Mascarenhas. Diffusion generated motion by mean curvature. CAM Report 92-33, University of California, Dept. of Math, Los Angeles, 1992.

[16] B. Merriman, J. Bence, and S. Osher. Diffusion generated motion by mean curvature. In J.E. Taylor, editor, *Computational Crystal Growers Workshop*, pages 73–83. American Mathematical Society, Providence, Rhode Island, 1992. Also available as UCLA CAM Report 92-18, April 1992.

[17] B. Merriman, J. Bence, and S. Osher. Motion of multiple junctions: a level set approach. *J. Comput. Phys.*, 112(2):334–363, 1994.

[18] S. Osher and J.A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.*, 79:12–49, 1988.

[19] J. Rubinstein. Self-induced motion of line defects. *Quart. Appl. Math*, 49(1):1–9, 1991.

[20] S.J. Ruuth. *Efficient Algorithms for Diffusion-Generated Motion by Mean Curvature.* PhD thesis, University of British Columbia, Vancouver, Canada, 1996.

[21] S.J. Ruuth. A diffusion-generated approach to multiphase motion. *J. Comput. Phys.*, 145:166–192, 1998.

[22] S.J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *J. Comput. Phys.*, 144:603–625, 1998.

[23] S.J. Ruuth and B. Merriman. Convolution generated motion and generalized Huygens' principles for interface motion. CAM Report 98-04, University of California, Los Angeles, 1998.

[24] S.J. Ruuth, B. Merriman, and S. Osher. Convolution generated motion as a link between cellular automata and continuum pattern dynamics. CAM Report 98-43, University of California, Los Angeles, 1998.

[25] J.J. Tyson and J.P. Keener. Singular perturbation theory of traveling waves in excitable media. *Physica D*, 32:327–361, 1988.