Motion of Curves in Three Spatial Dimensions Using a Level Set Approach

Paul Burchard *
Li-Tien Cheng * †
Barry Merriman *
Stanley Osher *

February 5, 2001

Abstract

The level set method [13] was originally designed for problems dealing with codimension one objects, where it has been extremely successful, especially when topological changes in the interface, i.e., merging and breaking, occur. Attempts have been made to modify it to handle objects of higher codimension, such as vortex filaments, while preserving the merging and breaking property. We present numerical simulations of a level set based method for moving curves in $\mathbf{R}^3$, the model problem for higher codimension, that allows for topological changes. A vector valued level set function is used with the zero level set representing the curve. Our results show that this method can handle many types of curves moving under all types of geometrically based flows while automatically enforcing merging and breaking.

1 Introduction

Moving higher codimensional objects is of interest in many areas of mathematics and the physical world. We consider here the model problem of moving a curve in $\mathbf{R}^3$, a codimension two object, according to geometric quantities of the curve. This type of motion is called geometrically based

*Department of Mathematics, University of California Los Angeles, Los Angeles, California 90095. Research supported in part by ONR N00014-97-0027, NSF DMS 9706827, and ARO DAAG 55-98-1-0323.

†Research supported in part by an NDSEG Fellowship.

motion and, simply put, is the motion of a curve based on its configuration. Curves in $\mathbf{R}^3$ themselves, under a geometrically based motion, can be used to model various phenomena in the physical world. Vortex filaments such as smoke rings can be represented in this way, as can the vortex lines in superfluid helium [10]. Such flows of curves have also been used for active contours in image processing[11]. In pure mathematics, curve shortening, the motion of a curve by its curvature, has been and is continuing to be studied[2] and motion in the binormal direction has a link to the Schrodinger equation. Finally, extension of the level set method as a tool to handle objects of higher codimension is of great interest. A list of further problems dealing with codimension two objects can be found in [8].

In addition, we are further interested in cases of motions of curves that exhibit topological changes, i.e., merging and breaking. For curves in $\mathbf{R}^2$, this concept is simple and readily observed in two phase flow [17]. For curves in $\mathbf{R}^3$, merging and breaking should behave as in the case of smoke rings. This follows curve shortening principles as in the Ginzburg-Landau model (see [15],[4],[9]), meaning when two segments of the curve touch, the curve breaks and reconnects in the acute angle directions (see Figure 1). Pictures of merging and breaking can also be found in [10] and [15].

Attempts to extend the level set method for use on curves in $\mathbf{R}^3$ have been studied by De Giorgi[6] and Ambrosio and Soner[3]. They were interested in the theoretical aspects of curvature motion but also outlined algorithms to capture the flow. In the algorithms, a single level set function, usually the squared distance to the curve, was used to represent curves in $\mathbf{R}^3$. This was done in the standard level set way, with the curve being represented by the zero level set of the level set function. Note in this formulation, the zero level set is also the set of points achieving the minimum value. One problem, numerically, with this method is in accurately determining the location of the curve. The main problem, however, for the topic we consider here is that in simulations, we see the handling of topological changes does not carry over. A phenomenon called "thickening" occurs, where the zero level set develops a non-empty interior, when curves try to merge (see Figure 2 and [5]). The formulation, however, was successful in the theoretical study of curvature flow.

The study of curves in $\mathbf{R}^3$ has also been attempted from other directions, for example, using front tracking[7]. This is where the curve is parametrized and numerically represented by discrete points, each of which is then evolved under the flow. Following these points thus gives the curve at all time. The main problem with this approach lies in finding and enforcing merging or breaking when it occurs. This has proven to be difficult for curves in $\mathbf{R}^2$

and is equally difficult, if not more so, for curves in $\mathbf{R}^3$. On the other hand, another approach, diffusion generated motion by Ruuth, Merriman, Xin, and Osher[15], can correctly deal with topological changes but so far is limited to curvature flow. The results for curvature flow, however, are good and we have compared them with our results whenever possible.

2 Level Set Representation of the Curve

The representation we adopt makes use of two level set functions to model a curve in $\mathbf{R}^3$, an approach Ambrosio and Soner[3] suggested first but did not pursue because the theoretical aspects were too difficult. In this formulation, a curve is represented by the intersection between the zero level sets of two level set functions, ϕ and ψ, i.e., where $\phi = \psi = 0$. Here, ϕ and ψ can be considered as the two components of a vector valued level set function whose zeros give the curve in $\mathbf{R}^3$. Thus, we can consider objects of arbitrary codimension by using a vector valued level set function with the correct number of components. For example, the zeros of an m component vector valued function over $\mathbf{R}^n$ can be used to represent a codimension m object in dimension n. Note, the set of points satisfying $\phi = C_1$ and $\psi = C_2$, where C_1 and C_2 are constants, are also curves in $\mathbf{R}^3$. Finally, as is standard in level set methods, we regularize $|\nabla\phi|$ and $|\nabla\psi|$ to avoid division by zero at degeneracies.

Under this representation, moving a curve by a certain type of motion is accomplished by evolving the functions ϕ and ψ in $\mathbf{R}^3$, keeping in mind that the intersection of their zero level sets gives the curve. Usually, the curves gotten from the intersections of other level sets of ϕ and ψ will move under the same type of motion. This, of course, is not needed. The only requirement for movement of the other curves is not to interfere with the desired curve.

2.1 Geometric Quantities

In order to move a curve by a geometrically based motion, we need to be able to derive all relevant geometric quantities of the curve in terms of our representation, i.e., in terms of ϕ and ψ. Important quantities include tangent vectors, curvature times normal vectors, normal vectors, binormal vectors, and torsion times normal vectors.

To find the tangent vectors T, we notice that $\nabla\psi \times \nabla\phi$, taken on the curve, is tangent to the curve. So the tangent vectors are just a normalization of this,

$$T = \frac{\nabla\psi \times \nabla\phi}{|\nabla\psi \times \nabla\phi|}.$$

Note if we replace ϕ with $-\phi$, the direction of the tangent vectors will be reversed.

For the curvature times normal, κN, of the curve, we use the definition that it is the change in the tangent vector along the curve,

$$\kappa N = \frac{dT}{ds}.$$

Using directional derivatives, this becomes

$$\kappa N = \nabla T \cdot T = \begin{pmatrix} \nabla T_1 \cdot T \\ \nabla T_2 \cdot T \\ \nabla T_3 \cdot T \end{pmatrix},$$

where T_1, T_2, and T_3 are the components of T. We may then use the expression for T to write this in terms of ϕ and ψ.

For the normal vectors, N, of the curve, we use the definition that it is the normalization of κN,

$$N = \frac{\kappa N}{|\kappa N|}.$$

The binormal vectors B are then obtained using the definition

$$B = T \times N.$$

Note the direction of the binormal vectors are reversed if we replace ϕ by $-\phi$. Finally, the torsion times normal vectors can be derived using the definition

$$\tau N = -\nabla B \cdot T.$$

All these geometric quantities can be written in terms of ϕ and ψ by using the corresponding expression for T. Also, note the above geometric quantities derived at an arbitrary point in $\mathbf{R}^3$ are quantities for the curve $\{\phi = C_1, \psi = C_2\}$ that passes through that point. For more on the definitions of the geometric quantities introduced above, see [2].

3 The Evolution Equation

Moving the curve in $\mathbf{R}^3$ using our representation requires moving the level set functions ϕ and ψ. We will first investigate the motion of a curve under a given vector field v in $\mathbf{R}^3$. From standard level set ideas, we know that the partial differential equation

$$\phi_t + v \cdot \nabla\phi = 0,$$

moves the level sets of ϕ according to v. Similarly,

$$\psi_t + v \cdot \nabla\psi = 0,$$

moves the level sets of ψ according to v. Therefore, the system of partial differential equations

$$\begin{cases} \phi_t + v \cdot \nabla\phi &= 0 \\ \psi_t + v \cdot \nabla\psi &= 0, \end{cases}$$

moves the intersections of the level sets of ϕ and ψ, especially the zero level sets, according to v. This can be derived in a more precise fashion.

Let $\gamma(s,t)$ denote the intersection between two level set surfaces of ϕ and ψ. So $\phi(\gamma(s,t),t) = C_1$ and $\psi(\gamma(s,t),t) = C_2$. Taking a derivative with respect to t gives

$$\begin{cases} \nabla\phi(\gamma(s,t),t) \cdot \gamma_t(s,t) + \phi_t(\gamma(s,t),t) &= 0 \\ \nabla\psi(\gamma(s,t),t) \cdot \gamma_t(s,t) + \psi_t(\gamma(s,t),t) &= 0. \end{cases}$$

Since the curve is moving under the vector field v, this means $\gamma_t(s,t) = v$. Therefore, since C_1 and C_2 are arbitrary, we get back the system of equations above, valid in all of $\mathbf{R}^3$.

3.1 More General Motions

By allowing v depend on ϕ and ψ and their derivatives, we can write down the evolution equations for any type of motion. For example, setting $v = \kappa N$ in the evolution equations above gives curvature motion in the normal direction. Similarly, $v = N$ gives motion in the normal direction at unit speed, $v = B$ gives motion in the binormal direction at unit speed, $v = \tau N$ gives torsion motion in the normal direction, and $T \times \kappa N$ gives a motion related to that of vortex lines in superfluid helium. We now study some of these motions more carefully and present numerical discretizations and results.

4 Curvature Motion

The evolution equations for curvature motion take the form

$$\begin{cases} \phi_t + \kappa N \cdot \nabla\phi &= 0 \\ \psi_t + \kappa N \cdot \nabla\psi &= 0, \end{cases}$$

where κN is as defined above in all of $\mathbf{R}^3$. This, in fact, can also be derived from modified gradient descent minimizing the length of the curve. For codimension one and the standard level set method, this means the introduction of the term $|\nabla\phi|$ which still preserves energy minimization (see, e.g., [12]). Notice first that the length of the curve represented by the intersection of the zero level sets of ϕ and ψ can be written as

$$L(\phi,\psi) = \int_{\mathbf{R}^3} \delta(\phi)\delta(\psi)|P_{\nabla\psi}\nabla\phi||\nabla\psi|dx,$$

where δ is the one dimensional delta function and P_v is the orthogonal projection matrix that projects vectors onto the plane with normal vector v,

$$P_v = I - \frac{v\otimes v}{|v|^2}.$$

In $\mathbf{R}^3$, we have $|P_v w| = \dfrac{|v\times w|}{|v|}$ and so we can also write the length as

$$L(\phi,\psi) = \int_{\mathbf{R}^3} \delta(\phi)\delta(\psi)|\nabla\phi\times\nabla\psi|dx.$$

Proposition 1 *The Euler-Lagrange equations for this energy are*

$$\begin{aligned} 0 &= -\nabla\cdot\left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}|\nabla\psi|\right)\delta(\psi)\delta(\phi) \\ 0 &= -\nabla\cdot\left(\frac{P_{\nabla\phi}\nabla\psi}{|P_{\nabla\phi}\nabla\psi|}|\nabla\phi|\right)\delta(\phi)\delta(\psi). \end{aligned}$$

Proof. See Appendix.

This can be rewritten as

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \delta(\phi)\delta(\psi) & 0 \\ 0 & \delta(\phi)\delta(\psi) \end{pmatrix} \begin{pmatrix} -\nabla\cdot\left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}|\nabla\psi|\right) \\ -\nabla\cdot\left(\frac{P_{\nabla\phi}\nabla\psi}{|P_{\nabla\phi}\nabla\psi|}|\nabla\phi|\right) \end{pmatrix}.$$

Following standard level set practice[19], we try to replace the matrix of delta functions, viewed as smoothed out delta functions, with a positive definite matrix that will give, on the right hand side of the Euler-Lagrange equations,

$$\begin{pmatrix} \kappa N \cdot \nabla\phi \\ \kappa N \cdot \nabla\psi \end{pmatrix}.$$

Then modified gradient descent minimizing the length of the curve will be equivalent to curvature motion. For this, we have

Proposition 2 *The replacement matrix that gives equivalence is the symmetric positive definite matrix given by*

$$\begin{pmatrix} \frac{|\nabla\phi|}{|P_{\nabla\phi}\nabla\psi|} & \frac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\psi}\nabla\phi||\nabla\psi|} \\ \frac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\phi}\nabla\psi||\nabla\phi|} & \frac{|\nabla\psi|}{|P_{\nabla\psi}\nabla\phi|} \end{pmatrix}.$$

Proof. See Appendix.

Note because this replacement matrix is symmetric positive definite, the length is still being minimized, i.e., $\frac{d}{dt}L(\phi,\psi) \leq 0$. This means curvature motion, under our representation, follows a curve shortening process.

4.1 Numerical Considerations

For all our numerical discretizations, we will lay down a uniform grid over $\mathbf{R}^3$ and use finite difference schemes. Finite differencing simplifies high order numerical discretizations of the partial differential equations and the uniform grid simplifies finite difference scheme construction and implementation. We thus discretize the curvature evolution equation, which is parabolic, by using second order central differencing on all spatial derivatives (see, e.g., [19]). The case $T = 0$ is regularized to remove singularities in the curvature expression. This is usually accomplished by introducing small positive constants into the denominators to prevent them from becoming zero, a standard level set practice (see [12]). The results we get using this particular regularization seems to agree with the curve shortening results of [15]. For the time discretization, we use Total Variation Diminishing Runge-Kutta (TVD-RK) of third order (see [16]). The associated Courant-Friedrichs-Lewy (CFL) condition says that the time step Δt needs to be less than a constant times Δx^2, where Δx denotes the spatial step size. In our simulations, we usually take the constant to be 0.5.

This representation allows topological changes to occur, as shown by our numerical simulations. The time of merging or breaking does not have to be computed and there is no need for switches to enforce the topological changes. All of this is automatically handled by the representation. The evolution equation is simply solved until the desired time and the resulting ϕ and ψ gives the curve, even when merging or breaking has taken place previously. Also the curve location does not have to be computed until the curve is to be plotted.

Plotting itself is carried out by using interpolation schemes. Each cube in the grid is broken up into six tetrahedra, inside of which ϕ and ψ can be approximated by hyperplanes. The intersection of the zero level sets of the two hyperplanes can then be computed, giving a small segment of line inside each tetrahedron. The union of all these segments gives an approximation of the curve. Higher order interpolation schemes can also be used without adversely affecting the speed of the algorithm since curve location is only needed at the end of the run.

Numerical simulations show that the method is second order accurate. One such result is shown in Table 1. This test compared the numerical evolution of a double helix in $\mathbf{R}^3$ with the exact solution, as detailed in [15]. The errors computed are in the maximum norm.

We consider the motion of a single helix in Figure 3. The helix straightens out, as it should, as time progresses. Evolution of two slightly translated helices is presented in Figure 4. The helices move independent of each other, each one straightening itself out, until they touch. Merging then occurs, with the resulting curves continuing to flow by curvature and shrink. Another example with two helices is shown in Figure 5. The two strands again touch and merging and breaking occurs. The resulting curve then continues to flow by curvature. In Figure 6, we consider the motion of linked rings. At first each ring will shrink its radius r by a speed of $\frac{1}{r}$. Eventually the rings touch and merging and breaking occurs. The resulting curve then continues flowing by curvature. This result can be compared to that in [15]. Initialization for the level set functions for these rings are derived by the same strategy as in [15]. Finally, Figures 7 and 8 show the evolution of other curves. The initial level set functions of Figure 8 are a cube and an ellipsoid. From these examples, we see that our method can handle curvature motion, even in the presence of merging and breaking of the curve.

5 Normal and Binormal Motion

The evolution equation for motion in the normal direction at unit speed is

$$\begin{cases} \phi_t + N \cdot \nabla\phi &= 0 \\ \psi_t + N \cdot \nabla\psi &= 0. \end{cases}$$

We discretize the time derivative using third order TVD-RK and all spatial derivatives using second order central differencing. Singularities occurring at $|T| = 0$ and $|\kappa N| = 0$ are regularized. Note geometrically, N is not defined when $|\kappa N| = 0$. Also the use here of central differencing in space is for convenience. High order schemes such as ENO and WENO schemes can be used in general by viewing the evolution equations as transport equations.

We consider a potato chip shaped curve as our initial curve in Figure 9. Normal motion in our simulations causes a sharp kink to develop in the curve after a certain time. This is because parts of the curve have rammed together and merging is enforced. In a standard tracking algorithm that allows curves to pass through each other, a swallow tail would appear instead. The level set functions for the initial curve are a sphere and an ellipsoid.

The evolution equation for motion in the binormal direction at unit speed is

$$\begin{cases} \phi_t + B \cdot \nabla\phi &= 0 \\ \psi_t + B \cdot \nabla\psi &= 0. \end{cases}$$

Once again, we discretize the time derivative using third order TVD-RK and all the spatial derivatives using second order central differencing. The singularities occurring at $|T| = 0$ and $|\kappa N| = 0$ are regularized. Note ENO and WENO schemes can also be used here for the spatial discretization.

We consider the simple case of a circle moving under binormal motion in Figure 10. The circle is translated, which is the correct solution. In Figure 11, we look at the evolution of two helices. Both slightly rotate and in opposite directions.

The cases of normal and binormal motion are not as nice as the case of curvature motion and not all initial curves evolve nicely. More work needs to be done on the discretization of the equations, especially the regularization of singularities. Flows we have not studied here include motion by torsion times normal and motion under the velocity field $T \times \kappa N$. The latter motion is related to vortex lines moving in superfluid helium.

5.1 Combinations

We can combine the motions we have studied above to form other types of motions. For example, taking the velocity field $v = N + \epsilon\kappa N$ gives motion in the normal direction with some curvature flow.

We look at the evolution of the potato chip shaped curve in Figure 12. ϵ is taken to be 0.1 and the result does not have a sharp kink anymore due to regularization by the curvature term.

6 Remarks

Some difficulties that are left include theoretical justification, which we will not investigate here, creating an optimal local method, and initializing ϕ and ψ to create a given curve.

6.1 Local Level Set Method

We would like to solve our evolution equations only in a small neighborhood of the curve instead of over all of $\mathbf{R}^3$. This would give optimal efficiency in both speed and memory usage. Solving in all of $\mathbf{R}^3$, however, is sometimes needed, for example, when the curves coming from the intersections of other level sets of ϕ and ψ play a role. It can also be adequate, for example, if the problem we are considering requires other equations to be solved in all of $\mathbf{R}^3$. But for the type of problems we have discussed here, a more local method is needed. Such an algorithm for curves in $\mathbf{R}^2$ or surfaces in $\mathbf{R}^3$ has been created[14] (see also [1]) but a few things need to be added when considering curves in $\mathbf{R}^3$. The main idea involves only doing computations in a tube around the curve with radius a constant times Δx. Reinitialization needs to be performed occasionally to keep errors at the boundary of the tube from influencing the curve. For curves in $\mathbf{R}^2$, this is accomplished by replacing the level set function with the signed distance function to the curve at each time step. Thus at each grid point, the level set function is given a value depending only on its zero level and not the other level sets.

Our first step in optimizing our algorithm is to cut down one of the dimensions by localizing around the zero level set of one of the level set functions, say ψ. The motion of ψ will exactly follow the procedure stated in [14] to achieve localization. The motion of ϕ will also follow some of the same techniques. Instead of the standard signed distance reinitialization, ϕ

can be reinitialized by first solving

$$\begin{cases} \tilde{\phi}_t + \text{sign}(\phi)(|P_{\nabla\psi}\nabla\tilde{\phi}| - 1) = 0 \\ \tilde{\phi}(t=0) = \phi, \end{cases}$$

to its steady state solution, $\tilde{\phi}_{\text{inf}}$, and then solving

$$\begin{cases} \hat{\phi}_t + \text{sign}(\phi)\dfrac{\nabla\psi}{|\nabla\psi|} \cdot \nabla\hat{\phi} = 0 \\ \hat{\phi}(t=0) = \tilde{\phi}_{\text{inf}}, \end{cases}$$

to its steady state solution, finally replacing ϕ by this function. This can be done every few time steps. The first equation makes ϕ into signed distance away from its zero level set, with distance measured on the level set surfaces of ψ. The second equation makes the level set surfaces of ϕ perpendicular to the zero level set surface of ψ. Both do not change the location of the intersection of the zero level sets but makes the other level sets well behaved during the flow. All this together allows us to treat the problem like a two dimensional one, centered about the zero level set of ψ, thus effectively reducing both the speed and memory usage of the algorithm.

A completely localized, and thus optimal, algorithm for curves in $\mathbf{R}^3$, however, has not yet been completed. Certain problems may arise from such an algorithm, for example, a twist of the level set functions about the curve may introduce spurious curves during topological changes. These kinds of abnormalities do not occur if ϕ and ψ are globally defined. Also the reinitialization process in standard local level set methods needs to be further studied and adapted.

6.2 Initialization

Another issue is how to choose the initial level set functions to create a desired curve. In some occasions, the initial functions are given, for example, as Clebsch variables (see [18]). Usually, however, they need to be constructed by hand. The difficulty in this lies in forming the functions ϕ and ψ in all of $\mathbf{R}^3$. Forming these functions local to the curve is very easy but sometimes these local constructions cannot be extended to $\mathbf{R}^3$, causing problems during topological changes. However, as Figure 6 shows, creating initial level set functions for complicated curves is not impossible. Another problem that may arise is that some constructions may hamper merging for certain motions. This also needs to be studied further.

7 Conclusion

We have analyzed a level set based method for representing and moving higher codimensional objects, especially curves in $\mathbf{R}^3$. As numerical results show, the representation automatically handles mergings and breakings of the curve. Evolution equations on the level set functions can then be used to move the curve under a variety of geometrically based flows. An underlying uniform grid allows for easy high order finite difference scheme constructions. All this sets up a foundation to deal with higher codimensional objects, especially when merging and breaking can occur.

8 Acknowledgments

The authors would like to thank Steve Ruuth for his help on the initialization of level set functions for some of the curves presented here as well as for his time and input on the subject of curves in $\mathbf{R}^3$ and diffusion generated motion.

9 Proofs of Propositions

These "proofs" are really only formal derivations. For example, we have not defined the precise spaces of functions we are using. Nevertheless, we present them as a formal guide to why the method seems to work well.

Proof of Proposition 1.

We will derive the Euler-Lagrange equations for the energy

$$L(\phi,\psi) = \int_{\mathbf{R}^n} |P_{\nabla\psi}\nabla\phi||\nabla\psi|\delta(\phi)\delta(\psi)dx,$$

where ϕ and ψ are real valued functions over $\mathbf{R}^n$.

Note

$$|P_{\nabla\psi}\nabla\phi|^2|\nabla\psi|^2 = |\nabla\phi|^2|\nabla\psi|^2 - (\nabla\phi\cdot\nabla\psi)^2 = |P_{\nabla\phi}\nabla\psi|^2|\nabla\phi|^2,$$

and, therefore,

$$|P_{\nabla\psi}\nabla\phi||\nabla\psi| = \sqrt{|\nabla\phi|^2|\nabla\psi|^2 - (\nabla\phi\cdot\nabla\psi)^2} = |P_{\nabla\phi}\nabla\psi||\nabla\phi|.$$

So

$$\begin{aligned}
&\frac{d}{ds}\bigg|_{s=0} (|P_{\nabla\psi+s\nabla\nu}(\nabla\phi + s\nabla\eta)||\nabla\psi + s\nabla\nu|\delta(\phi + s\eta)\delta(\psi + s\nu)) = \\
&= \frac{d}{ds}\bigg|_{s=0} (|P_{\nabla\psi}(\nabla\phi + s\nabla\eta)||\nabla\psi|\delta(\phi + s\eta)\delta(\psi)) + \\
&\quad \frac{d}{ds}\bigg|_{s=0} (|P_{\nabla\psi+s\nabla\nu}\nabla\phi||\nabla\psi + s\nabla\nu|\delta(\phi)\delta(\psi + s\nu)) \\
&= \frac{d}{ds}\bigg|_{s=0} (|P_{\nabla\psi}(\nabla\phi + s\nabla\eta)||\nabla\psi|\delta(\phi + s\eta)\delta(\psi)) + \\
&\quad \frac{d}{ds}\bigg|_{s=0} (|P_{\nabla\phi}(\nabla\psi + s\nabla\nu)||\nabla\phi|\delta(\psi + s\nu)\delta(\phi)).
\end{aligned}$$

Therefore,

$$\begin{aligned}
&\frac{d}{ds}\bigg|_{s=0} L(\phi + s\eta, \psi + s\nu) = \\
&\frac{d}{ds}\bigg|_{s=0} \textstyle\int_{\mathbf{R}^n} |P_{\nabla\psi}(\nabla\phi + s\nabla\eta)||\nabla\psi|\delta(\phi + s\eta)\delta(\psi)dx + \\
&\frac{d}{ds}\bigg|_{s=0} \textstyle\int_{\mathbf{R}^n} |P_{\nabla\phi}(\nabla\psi + s\nabla\nu)||\nabla\phi|\delta(\psi + s\nu)\delta(\phi)dx.
\end{aligned}$$

We now state and use

Lemma 1 *The Euler-Lagrange equation of the energy*

$$E(\phi) = \int_{\mathbf{R}^3} \delta(\phi)\delta(\psi)|P_{\nabla\psi}\nabla\phi||\nabla\psi|dx,$$

is

$$0 = -\nabla \cdot \left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}|\nabla\psi| \right) \delta(\psi)\delta(\phi).$$

So, using this and its version with ϕ replaced by ψ, we get that the Euler-Lagrange equations are

$$\begin{aligned}
0 &= -\nabla \cdot \left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}|\nabla\psi| \right) \delta(\psi)\delta(\phi) \\
0 &= -\nabla \cdot \left(\frac{P_{\nabla\phi}\nabla\psi}{|P_{\nabla\phi}\nabla\psi|}|\nabla\phi| \right) \delta(\phi)\delta(\psi).
\end{aligned}$$

Proof of Lemma 1.

We will prove the more general statement that the Euler-Lagrange equation of the energy

$$E(\phi) = \int_{\mathbf{R}^3} \beta \left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|} \right) \delta(\psi)\delta(\phi)|\nabla\psi \times \nabla\phi|dx.$$

is

$$0 = -\nabla \cdot (P_{\nabla\psi} \nabla \beta (P_{\nabla\psi} \nabla\phi) |\nabla\psi|) \delta(\psi) \delta(\phi),$$

where β is a real valued homogeneous of degree one function over $\mathbf{R}^3$. Note $\beta(p) = |p|$, $p \in \mathbf{R}^3$, then finishes the proof.

We have, since β is a homogeneous of degree one function,

$$\begin{aligned} E(\phi) &= \int_{\mathbf{R}^3} \beta\left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}\right) |P_{\nabla\psi}\nabla\phi||\nabla\psi|\delta(\psi)\delta(\phi)dx \\ &= \int_{\mathbf{R}^3} \beta(P_{\nabla\psi}\nabla\phi)\delta(\phi)|\nabla\psi|\delta(\psi). \end{aligned}$$

So,

$$\begin{aligned} \left.\frac{d}{ds}\right|_{s=0} E(\phi + s\eta) &= \int_{\mathbf{R}^3} (D\beta(P_{\nabla\psi}\nabla\phi) \cdot P_{\nabla\psi}\nabla\eta)\delta(\phi)|\nabla\psi|\delta(\psi)dx + \\ &\quad + \int_{\mathbf{R}^3} \beta(P_{\nabla\psi}\nabla\phi)\delta'(\phi)|\nabla\psi|\delta(\psi)\eta dx \\ &= I + J, \end{aligned}$$

where I is the first integral on the right hand side and J is the second. Now,

$$\begin{aligned} I &= \int_{\mathbf{R}^3} \left(D\beta(P_{\nabla\psi}\nabla\phi) \cdot \left(\nabla\eta - \frac{\nabla\psi \cdot \nabla\eta}{|\nabla\psi|^2}\nabla\psi\right)\right) \delta(\phi)|\nabla\psi|\delta(\psi)dx \\ &= -\int_{\mathbf{R}^3} \nabla \cdot (D\beta(P_{\nabla\psi}\nabla\phi)\delta(\phi)|\nabla\psi|\delta(\psi))\eta dx + \\ &\quad + \int_{\mathbf{R}^3} \nabla \cdot \left((D\beta(P_{\nabla\psi}\nabla\phi) \cdot \nabla\psi)\nabla\psi\delta(\phi)\frac{\delta(\psi)}{|\nabla\psi|}\right) \eta dx \\ &= -\int_{\mathbf{R}^3} \nabla \cdot (D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|\delta(\psi))\delta(\phi)\eta dx + \\ &\quad - \int_{\mathbf{R}^3} (D\beta(P_{\nabla\psi}\nabla\phi) \cdot \nabla\phi)\delta'(\phi)|\nabla\psi|\delta(\psi)\eta dx + \\ &\quad + \int_{\mathbf{R}^3} \nabla \cdot \left((D\beta(P_{\nabla\psi}\nabla\phi) \cdot \nabla\psi)\nabla\psi\frac{\delta(\psi)}{|\nabla\psi|}\right) \delta(\phi)\eta dx + \\ &\quad + \int_{\mathbf{R}^3} (D\beta(P_{\nabla\psi}\nabla\phi) \cdot \nabla\psi)(\nabla\psi \cdot \nabla\phi)\delta'(\phi)\frac{\delta(\psi)}{|\nabla\psi|}\eta dx \\ &= -\int_{\mathbf{R}^3} \nabla \cdot \Big(D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|\delta(\psi) - \\ &\qquad (D\beta(P_{\nabla\psi}\nabla\phi) \cdot \nabla\psi)\nabla\psi\frac{\delta(\psi)}{|\nabla\psi|}\Big) \delta(\phi)\eta dx + \end{aligned}$$

$$
\begin{aligned}
& - \int_{\mathbf{R}^3} \Big((D\beta(P_{\nabla\psi}\nabla\phi)\cdot\nabla\phi)|\nabla\psi| - \\
& \qquad (D\beta(P_{\nabla\psi}\nabla\phi)\cdot\nabla\psi)\frac{\nabla\psi\cdot\nabla\phi}{|\nabla\psi|}\Big)\,\delta'(\phi)\delta(\psi)\eta dx \\
= \; & - \int_{\mathbf{R}^3} \nabla\cdot(P_{\nabla\psi}D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|\delta(\psi))\delta(\phi)\eta dx + \\
& - \int_{\mathbf{R}^3} (P_{\nabla\psi}\nabla\phi\cdot D\beta(P_{\nabla\psi}\nabla\phi))\delta'(\phi)|\nabla\psi|\delta(\psi)\eta dx.
\end{aligned}
$$

Note,

$$
\nabla\cdot(P_{\nabla\psi}D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|\delta(\psi)) = \nabla\cdot(P_{\nabla\psi}D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|)\delta(\psi),
$$

since $P_{\nabla\psi}D\beta(P_{\nabla\psi}\nabla\phi)\cdot\nabla\psi = 0$.

Also for $p \in \mathbf{R}^3$, we have

$$
\beta((1+\epsilon)p) = (1+\epsilon)\beta(p).
$$

Thus using a suggestion by an anonymous referee, we take a derivative with respect to ϵ on both sides and evaluate at $\epsilon = 0$. This leads to the equality,

$$
p\cdot D\beta(p) = \beta(p).
$$

Note especially that $p = P_{\nabla\psi}\nabla\phi$ gives

$$
P_{\nabla\psi}\nabla\phi\cdot D\beta(P_{\nabla\psi}\nabla\phi) = \beta(P_{\nabla\psi}\nabla\phi).
$$

So altogether, we get

$$
\left.\frac{d}{ds}\right|_{s=0} E(\phi+s\eta) = -\int_{\mathbf{R}^3} \nabla\cdot(P_{\nabla\psi}D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|)\delta(\psi)\delta(\phi)\eta dx,
$$

and so $\left.\frac{d}{ds}\right|_{s=0} E(\phi+s\eta) = 0$ for arbitrary η leads to the Euler-Lagrange equation

$$
-\nabla\cdot(P_{\nabla\psi}D\beta(P_{\nabla\psi}\nabla\phi)|\nabla\psi|)\delta(\psi)\delta(\phi) = 0.
$$

Proof of Proposition 2.

We will show that we can get $\kappa N\cdot\nabla\phi$ and $\kappa N\cdot\nabla\psi$ from replacing the matrix of delta functions in the right hand side of the Euler-Lagrange equations.

We first state the following,

Lemma 2 $P_{\nabla\psi}\kappa N \cdot \nabla\phi = -\nabla \cdot \left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}|\nabla\psi| \right) \frac{|P_{\nabla\psi}\nabla\phi|}{|\nabla\psi|}$.

We see from this that

$$\begin{aligned} \kappa N \cdot P_{\nabla\psi}\nabla\phi &= -\nabla \cdot \left(\frac{P_{\nabla\psi}\nabla\phi}{|P_{\nabla\psi}\nabla\phi|}||\nabla\psi| \right) \frac{|P_{\nabla\psi}\nabla\phi|}{|\nabla\psi|} \\ \kappa N \cdot P_{\nabla\phi}\nabla\psi &= -\nabla \cdot \left(\frac{P_{\nabla\phi}\nabla\psi}{|P_{\nabla\phi}\nabla\psi|}||\nabla\phi| \right) \frac{|P_{\nabla\phi}\nabla\psi|}{|\nabla\phi|}. \end{aligned}$$

So we want to look for functions f and g that may depend on ϕ, ψ, and their derivatives such that

$$\begin{pmatrix} f_1 & g_1 \\ g_2 & f_2 \end{pmatrix} \begin{pmatrix} \kappa N \cdot P_{\nabla\psi}\nabla\phi \\ \kappa N \cdot P_{\nabla\phi}\nabla\psi \end{pmatrix} = \begin{pmatrix} \kappa N \cdot \nabla\phi \\ \kappa N \cdot \nabla\psi \end{pmatrix}.$$

This is equivalent to

$$\begin{aligned} \kappa N \cdot (f_1 P_{\nabla\psi}\nabla\phi + g_1 P_{\nabla\phi}\nabla\psi) &= \kappa N \cdot \nabla\phi \\ \kappa N \cdot (g_2 P_{\nabla\psi}\nabla\phi + f_2 P_{\nabla\phi}\nabla\psi) &= \kappa N \cdot \nabla\psi. \end{aligned}$$

Therefore, we are looking for a decomposition of $\nabla\phi$ in terms of $P_{\nabla\psi}\nabla\phi$ and $P_{\nabla\phi}\nabla\psi$ and similarly for $\nabla\psi$. First note that this is possible since

$$\nabla\phi \times P_{\nabla\phi}\nabla\psi = \nabla\phi \times \nabla\psi = P_{\nabla\psi}\nabla\phi \times \nabla\psi,$$

so we have that $\nabla\phi, \nabla\psi, P_{\nabla\psi}\nabla\phi, P_{\nabla\phi}\nabla\psi$ all lie on the same plane. Also $P_{\nabla\psi}\nabla\phi$ and $P_{\nabla\phi}\nabla\psi$ span the plane if we disregard the degenerate case where $\nabla\phi$ and $\nabla\psi$ are parallel.

Taking a dot product of $\nabla\phi$ with the equations

$$\begin{aligned} \nabla\phi &= f_1 P_{\nabla\psi}\nabla\phi + g_1 P_{\nabla\phi}\nabla\psi \\ \nabla\psi &= g_2 P_{\nabla\psi}\nabla\phi + f_2 P_{\nabla\phi}\nabla\psi, \end{aligned}$$

gives

$$\begin{aligned} f_1 &= \frac{|\nabla\phi|^2}{|P_{\nabla\psi}\nabla\phi|^2} \\ g_2 &= \frac{\nabla\phi \cdot \nabla\psi}{|P_{\nabla\psi}\nabla\phi|^2}. \end{aligned}$$

Also taking a dot product of $\nabla\psi$ with the same equations gives

$$\begin{aligned} f_2 &= \frac{|\nabla\psi|^2}{|P_{\nabla\phi}\nabla\psi|^2} \\ g_1 &= \frac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\phi}\nabla\psi|^2}. \end{aligned}$$

Note that $f_1 = f_2$.

Therefore, replacing

$$\begin{pmatrix} \delta(\phi)\delta(\psi) & 0 \\ 0 & \delta(\phi)\delta(\psi) \end{pmatrix},$$

in the Euler-Lagrange equations by

$$\begin{pmatrix} f_1\dfrac{|P_{\nabla\psi}\nabla\phi|}{|\nabla\psi|} & g_1\dfrac{|P_{\nabla\phi}\nabla\psi|}{|\nabla\phi|} \\ g_2\dfrac{|P_{\nabla\psi}\nabla\phi|}{|\nabla\psi|} & f_2\dfrac{|P_{\nabla\phi}\nabla\psi|}{|\nabla\phi|} \end{pmatrix},$$

will give the curvature flow evolution equation. So the replacement matrix can be written as

$$\begin{pmatrix} \dfrac{|\nabla\phi|}{|P_{\nabla\phi}\nabla\psi|} & \dfrac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\psi}\nabla\phi||\nabla\psi|} \\ \dfrac{\nabla\phi\cdot\nabla\psi}{|P_{\nabla\phi}\nabla\psi||\nabla\phi|} & \dfrac{|\nabla\psi|}{|P_{\nabla\psi}\nabla\phi|} \end{pmatrix}.$$

In standard level set practice in general, this replacement can be used, unchanged, for all other Euler-Lagrange equations of curve flow in $\mathbf{R}^3$.

Note the determinant of our replacement matrix is

$$\frac{|\nabla\phi|^2|\nabla\psi|^2 - (\nabla\phi\cdot\nabla\psi)^2}{|P_{\nabla\psi}\nabla\phi|^2|\nabla\psi|^2},$$

which is equal to 1. Also the first entry is positive and the matrix is symmetric, so it is symmetric positive definite.

Proof of Lemma 2.

The main property we will show is

$$\nabla\cdot(T\times\nabla\psi) = \kappa N\cdot(\nabla\psi\times T),$$

where

$$T = \frac{\nabla\psi\times\nabla\phi}{|\nabla\psi\times\nabla\phi|}.$$

Using this to expand the right hand side of the evolution equation in the second method, we get

$$\begin{aligned}
-P_{\nabla\psi}\kappa N\cdot\nabla\phi &= -P_{\nabla\psi}\nabla\phi\cdot\kappa N \\
&= -\kappa N\cdot\left(\frac{|\nabla\psi|^2\nabla\phi-(\nabla\phi\cdot\nabla\psi)\nabla\psi}{|\nabla\psi|^2}\right) \\
&= -\kappa N\cdot\left(\frac{(\nabla\psi\times\nabla\phi)\times\nabla\psi}{|\nabla\psi|^2}\right) \\
&= -\kappa N\cdot\left(\frac{(\nabla\psi\times\nabla\phi)\times\nabla\psi}{|\nabla\psi\times\nabla\phi|}\right)\frac{|\nabla\psi\times\nabla\phi|}{|\nabla\psi|^2} \\
&= \kappa N\cdot(\nabla\psi\times T)\frac{|\nabla\psi\times\nabla\phi|}{|\nabla\psi|^2} \\
&= \nabla\cdot(T\times\nabla\psi)\frac{|\nabla\psi\times\nabla\phi|}{|\nabla\psi|^2} \\
&= \nabla\cdot\left(\frac{(\nabla\psi\times\nabla\phi)\times\nabla\psi}{|\nabla\psi\times\nabla\phi|}\right)\frac{|\nabla\psi\times\nabla\phi|}{|\nabla\psi|^2},
\end{aligned}$$

which finishes the proof.

The proof of the main property we used is as follows. Let e_1, e_2, e_3 be orthonormal. Then

$$\nabla\cdot X = \langle e_1(X), e_1\rangle + \langle e_2(X), e_2\rangle + \langle e_3(X), e_3\rangle,$$

where $e_1(f_1, f_2, f_3) = (e_1(f_1), e_2(f_2), e_3(f_3))$. This is since for $\tilde{e}_1, \tilde{e}_2, \tilde{e}_3$ orthonormal, $\tilde{e}_i = \sum_{j=1}^3 a_i^j e_j$, so

$$\begin{aligned}
\sum_{i=1}^3\langle\tilde{e}_i(X), \tilde{e}_i\rangle &= \sum_{i=1}^3\langle\sum_{j=1}^3 a_i^j e_j(X), \sum_{k=1}^3 a_i^k e_k\rangle \\
&= \sum_{i=1}^3\sum_{j=1}^3\sum_{k=1}^3 a_i^j a_i^k\langle e_j(X), e_k\rangle \\
&= \sum_{j=1}^3\sum_{k=1}^3\delta_{jk}\langle e_j(X), e_k\rangle \\
&= \sum_{j=1}^3\langle e_j(X), e_j\rangle.
\end{aligned}$$

Therefore, if $\tilde{e}_1 = (1,0,0), \tilde{e}_2 = (0,1,0), \tilde{e}_3 = (0,0,1)$, we get

$$\sum_{i=1}^3\langle\tilde{e}_i(X), \tilde{e}_i\rangle = \nabla\cdot X = \sum_{j=1}^3\langle e_j(X), e_j\rangle.$$

Now, let $e_1 = \dfrac{\nabla\psi \times \nabla\phi}{|\nabla\psi \times \nabla\phi|}, e_2 = \dfrac{e_1 \times \nabla\psi}{|e_1 \times \nabla\psi|}, e_3 = e_1 \times e_2$. Note e_1, e_2, e_3 is orthonormal and the quantity we wish to investigate is $\nabla \cdot (e_1 \times \nabla\psi)$. Now

$$\nabla \cdot (e_1 \times \nabla\psi) = \langle e_1(e_1 \times \nabla\psi), e_1\rangle + \langle e_2(e_1 \times \nabla\psi), e_2\rangle + \langle e_3(e_1 \times \nabla\psi), e_3\rangle.$$

But

$$e_1(e_1 \times \nabla\psi) = e_1(e_1) \times \nabla\psi + e_1 \times (e_1(\nabla\psi)),$$

so

$$\begin{aligned}
\langle e_1(e_1 \times \nabla\psi), e_1\rangle &= \langle e_1(e_1) \times \nabla\psi, e_1\rangle \\
&= \langle \kappa N \times \nabla\psi, e_1\rangle \\
&= \det(\kappa N, \nabla\psi, e_1) \\
&= \kappa N \cdot (\nabla\psi \times e_1).
\end{aligned}$$

Also,

$$\langle e_2(e_1 \times \nabla\psi), e_2\rangle = \langle e_2(e_1) \times \nabla\psi, e_2\rangle + \langle e_1 \times (e_2(\nabla\psi)), e_2\rangle.$$

Now e_1 is a unit vector field implies $e_2(e_1) \perp e_1$, which means $e_2(e_1)$ is a linear combination of $\nabla\psi$ and e_2. Therefore,

$$\langle e_2(e_1) \times \nabla\psi, e_2\rangle = 0,$$

and so

$$\langle e_2(e_1 \times \nabla\psi), e_2\rangle = \langle e_1 \times (e_2(\nabla\psi)), e_2\rangle.$$

Finally,

$$\langle e_3(e_1 \times \nabla\psi), e_3\rangle = \langle e_3(e_1) \times \nabla\psi, e_3\rangle + \langle e_1 \times (e_3(\nabla\psi)), e_3\rangle.$$

Now e_1 is a unit vector field implies $e_3(e_1) \perp e_1$, which means $e_3(e_1)$ is a linear combination of $\nabla\psi$ and e_2. Since

$$\langle e_2 \times \nabla\psi, e_3\rangle = -\langle e_2 \times e_3, \nabla\psi\rangle = -\langle e_1, \nabla\psi\rangle = 0,$$

therefore,

$$\langle e_3(e_1 \times \nabla\psi), e_3\rangle = \langle e_1 \times (e_3(\nabla\psi)), e_3\rangle.$$

So altogether, we have

$$\nabla \cdot (e_1 \times \nabla\psi) = \kappa N \cdot (\nabla\psi \times e_1) + \langle e_1 \times (e_2(\nabla\psi)), e_2\rangle + \langle e_1 \times (e_3(\nabla\psi)), e_3\rangle.$$

But

$$\begin{aligned}\langle e_1 \times (e_2(\nabla\psi)), e_2\rangle &= -\langle e_1 \times e_2, e_2(\nabla\psi)\rangle \\ &= -\langle e_3, e_2(\nabla\psi)\rangle,\end{aligned}$$

and

$$\begin{aligned}\langle e_1 \times (e_3(\nabla\psi)), e_3\rangle &= \langle e_3 \times e_1, e_3(\nabla\psi)\rangle \\ &= \langle e_2, e_3(\nabla\psi)\rangle.\end{aligned}$$

Now

$$\langle e_2, e_3(\nabla\psi)\rangle = \sum_{i=1}^{3}\sum_{j=1}^{3} a_i b_j \frac{\partial^2\psi}{\partial x_i \partial x_j} = \sum_{i=1}^{3}\sum_{j=1}^{3} b_i a_j \frac{\partial^2\psi}{\partial x_i \partial x_j} = \langle e_3, e_2(\nabla\psi)\rangle.$$

Therefore,

$$\nabla \cdot (e_1 \times \nabla\psi) = \kappa N \cdot (\nabla\psi \times e_1),$$

which is what we want.

References

[1] D. Adalsteinsson and J.A. Sethian *A Fast Level Set Method for Propagating Interfaces.* J. Comput. Phys., 118:269–277, 1995.

[2] S.J. Altschuler *Singularities of the curve shrinking flow for space curves.* J. Differential Geom., 34(2):491–514, 1991.

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

[4] L. Ambrosio and H.M. Soner. *A Measure Theoretic Approach to Higher Codimension Mean Curvature Flows.* Ann. Scuola Normale Sup. Pisa, Cl. Sci. 25(4):27–49, 1997.

[5] G. Bellettini and M. Novaga. *An Example of Three Dimensional Fattening for Linked Space Curves Evolving by Curvature.* Comm. Partial Differential Equations, 23:1475–1492, 1998.

[6] E. De Giorgi. *Barriers, boundaries, motion of manifolds.* Lectures held in Pavia, Italy, 1994.

[7] T.Y. Hou, J.S. Lowengrub, and M.J. Shelley. *Removing the Stiffness from Interfacial Flows with Surface Tension.* J. Comput. Phys., 114:312, 1994.

[8] S.D. Howison, J.D. Morgan, and J.R. Ockendon. *A Class of Codimension-Two Free Boundary Problems.* SIAM Rev., 39(2):221–253, 1997.

[9] R.L. Jerrard. *Fully Nonlinear Phase Field Equations and Generalized Mean Curvature Motion.* Comm. Partial Differential Equations, 20(1–2):233–265, 1995.

[10] J. Koplik and H. Levine *Vortex Reconnection in Superfluid Helium.* Physical Review Letters, 71(9):1375-1378, 1993.

[11] L.M. Lorigo, O. Faugeras, W.E.L. Grimson, R. Keriven, R. Kikinis, and C.F. Westin. *Co-Dimension 2 Geodesic Active Contours for MRA Segmentation.* Int'l Conf. Information Processing in Medical Imaging, June/July 1999.

[12] S. Osher and R. Fedkiw. *Level Set Methods: A Survey and Some Recent Results.* UCLA CAM Report, 00-08, 2000, to appear in J. Comput. Phys.

[13] S. Osher and J.A. Sethian. *Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations.* J. Comput. Phys., 79(1):12–49, 1988.

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

[15] S. Ruuth, B. Merriman, J. Xin, and S. Osher. *Diffusion-Generated Motion by Mean Curvature for Filaments.* UCLA CAM Report, 98-47, 1998.

[16] C.W. Shu and S. Osher *Efficient implementation of essentially nonoscillatory shock-capturing schemes.* J. Comput. Phys., 77(2):439–471, 1988.

[17] M. Sussman, P. Smerka, and S. Osher. *A level set method for computing solutions to incompressible two-phase flow.* J. Comput. Phys., 114:146–159, 1994.

[18] V.E. Zakharov, S.L. Musher, and A.M. Rubenchik. *Hamiltonian Approach to the Description of Non-Linear Plasma Phenomena.* Review Section of Physics Letters, 129(5):285–366, 1985.

[19] H.K. Zhao, T.F. Chan, B. Merriman, and S. Osher. *A variational level set approach to multiphase motion.* J. Comput. Phys., 127:179–195, 1996.

grid size	error	order
$32 \times 32 \times 32$	0.00459276	
$64 \times 64 \times 64$	0.00140586	1.7079
$128 \times 128 \times 128$	0.000356941	1.9777

Table 1: Order of accuracy analysis for a double helix moving under curvature flow. Results show the method is second order accurate under the maximum norm.

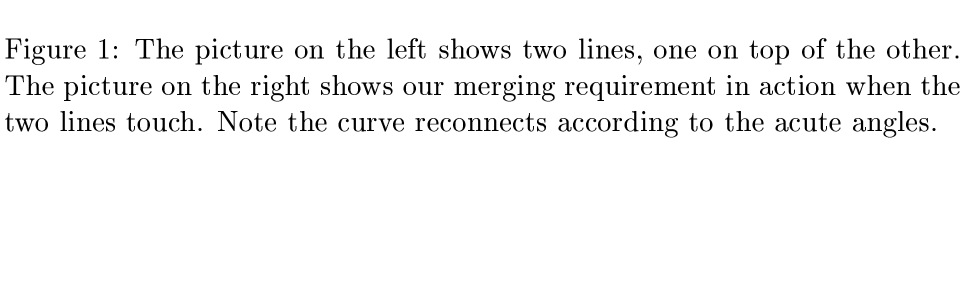

Figure 1: The picture on the left shows two lines, one on top of the other. The picture on the right shows our merging requirement in action when the two lines touch. Note the curve reconnects according to the acute angles.

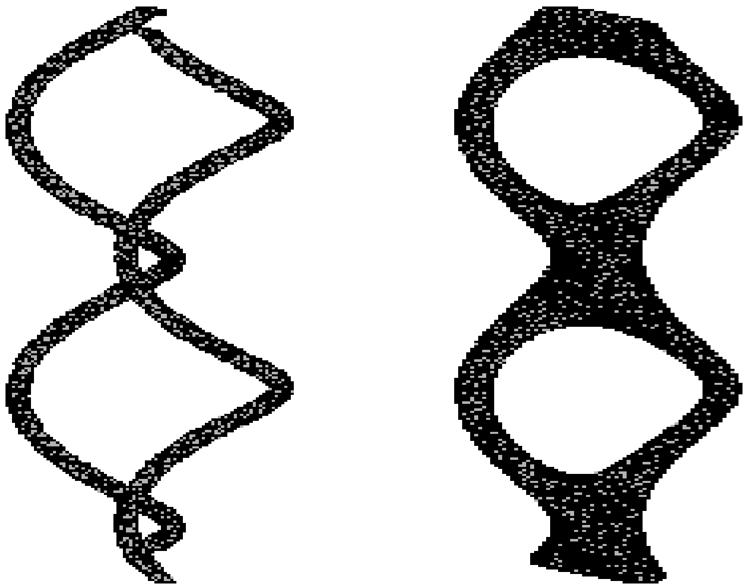

Figure 2: These pictures were generated using De Giorgi's method. The picture on the left shows two helices about to touch. The picture on the right shows thickening occurring when they do touch. The merging requirement is not satisfied here.

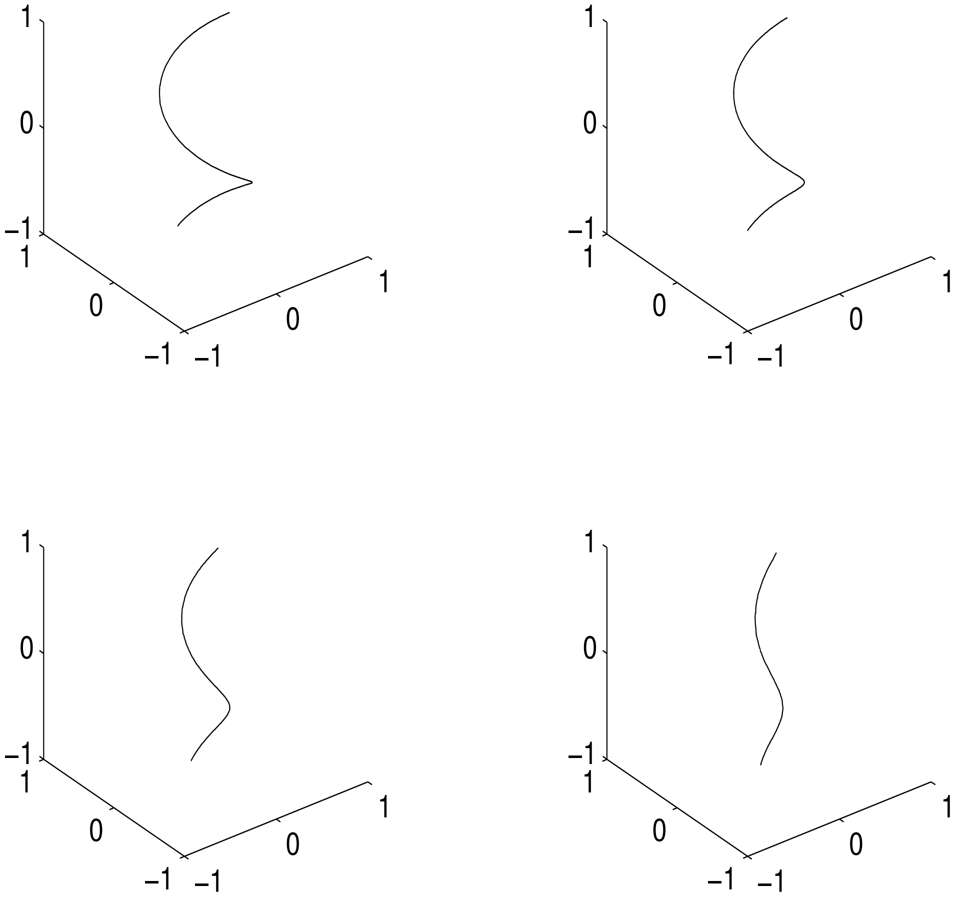

Figure 3: This shows a single helix evolving under curvature motion. The curve remains helical in form but the radius about the center axis shrinks, straightening out the curve.

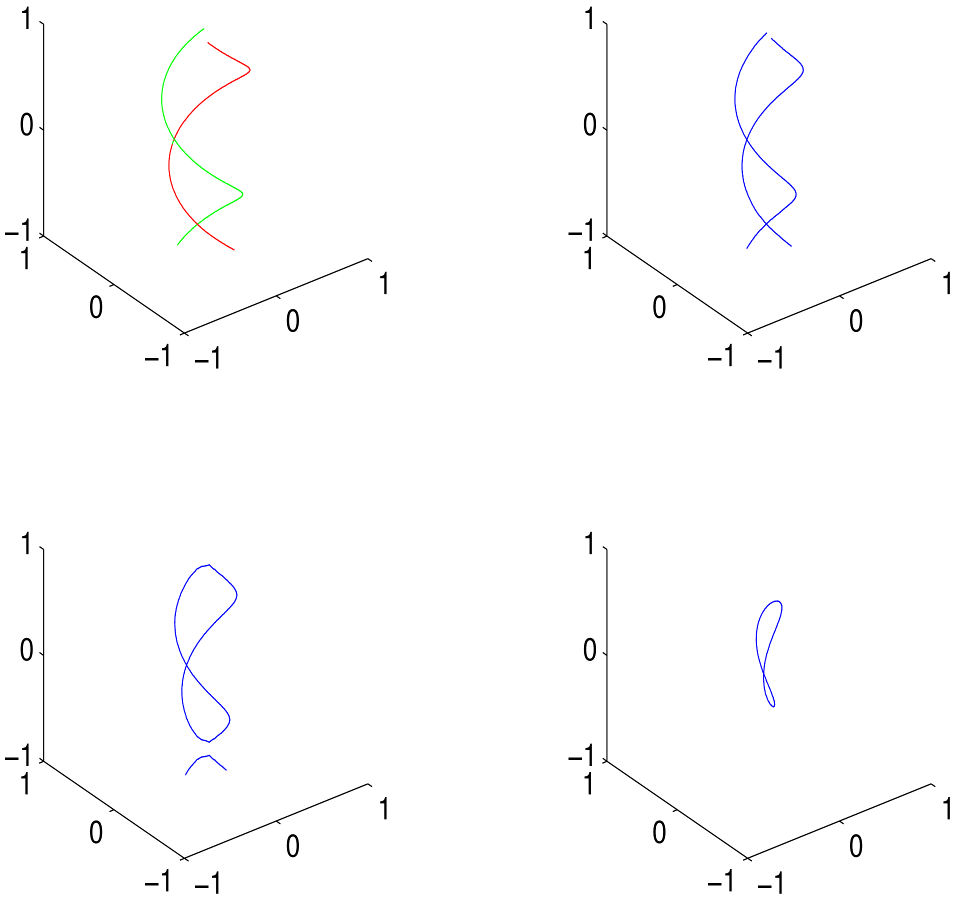

Figure 4: This shows two slightly translated helices evolving under curvature motion. The translation allows the helices to eventually touch and merge. The resulting curve then continues to evolve under curvature motion and shrink.

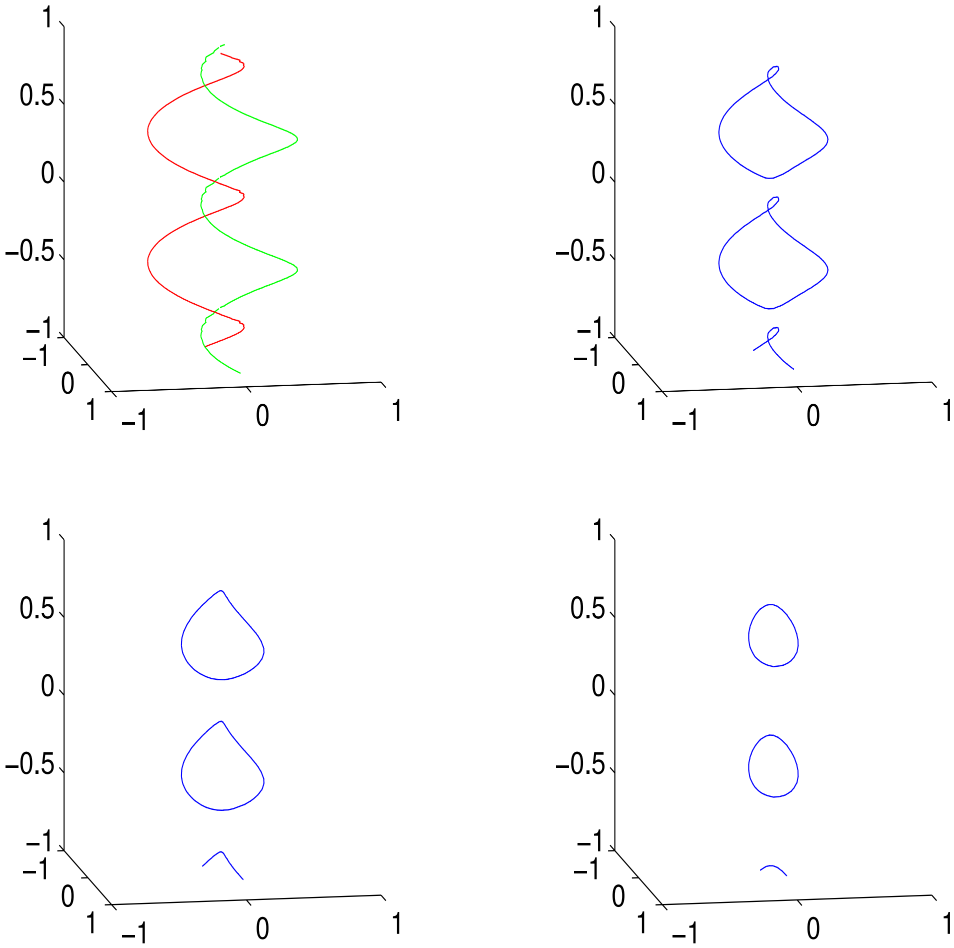

Figure 5: This is another picture of two helices evolving under curvature motion. The two touch and merge at a certain time. The resulting curve then continues to evolve under curvature motion.

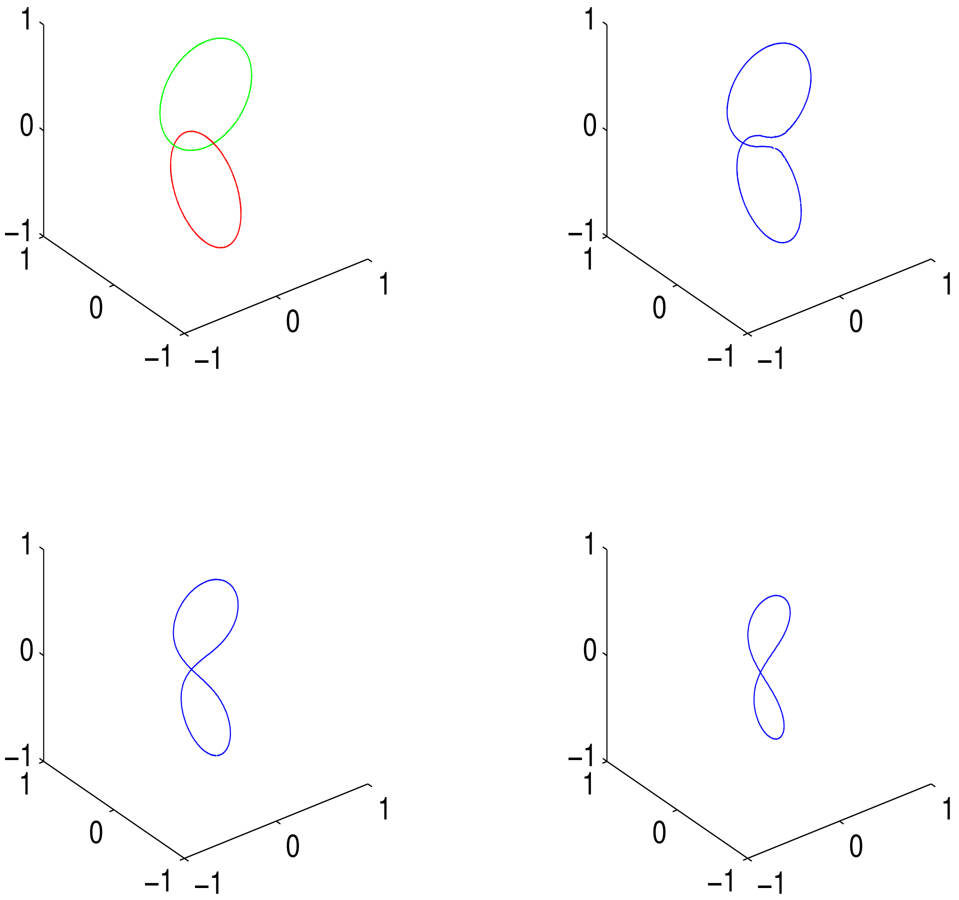

Figure 6: This shows two linked rings evolving under curvature motion. The two rings shrink independent of each other until they touch and merge. The resulting curve then continues to evolve under curvature motion.

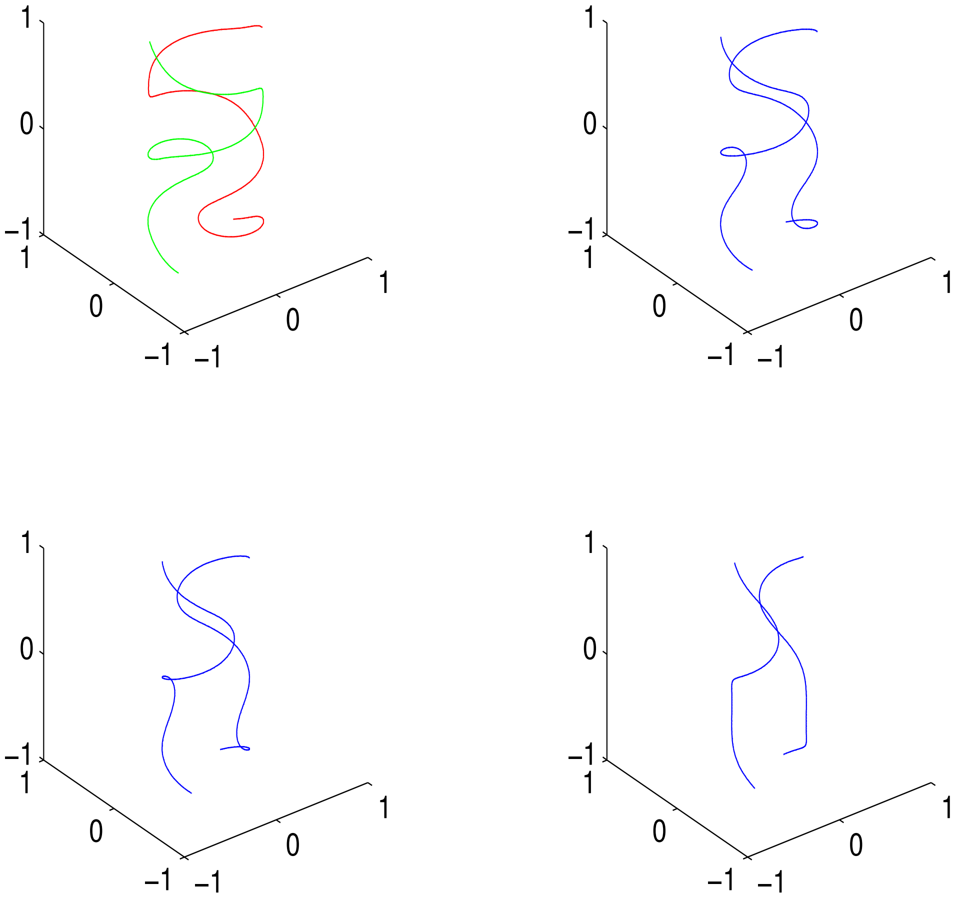

Figure 7: This shows two complicated curves evolving under curvature motion. The simulation stops before the curves touch.

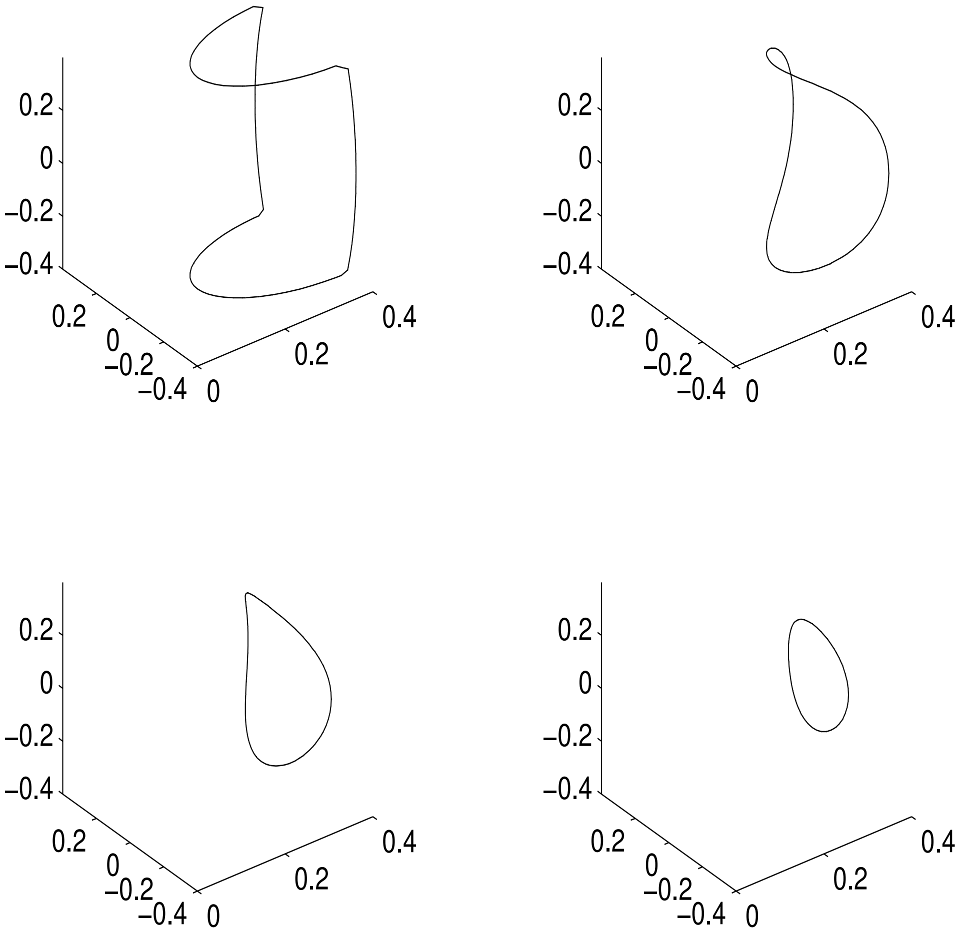

Figure 8: This shows a curve with kinks evolving under curvature motion. The kink is smoothed out almost immediately and then shrinks to a circle.

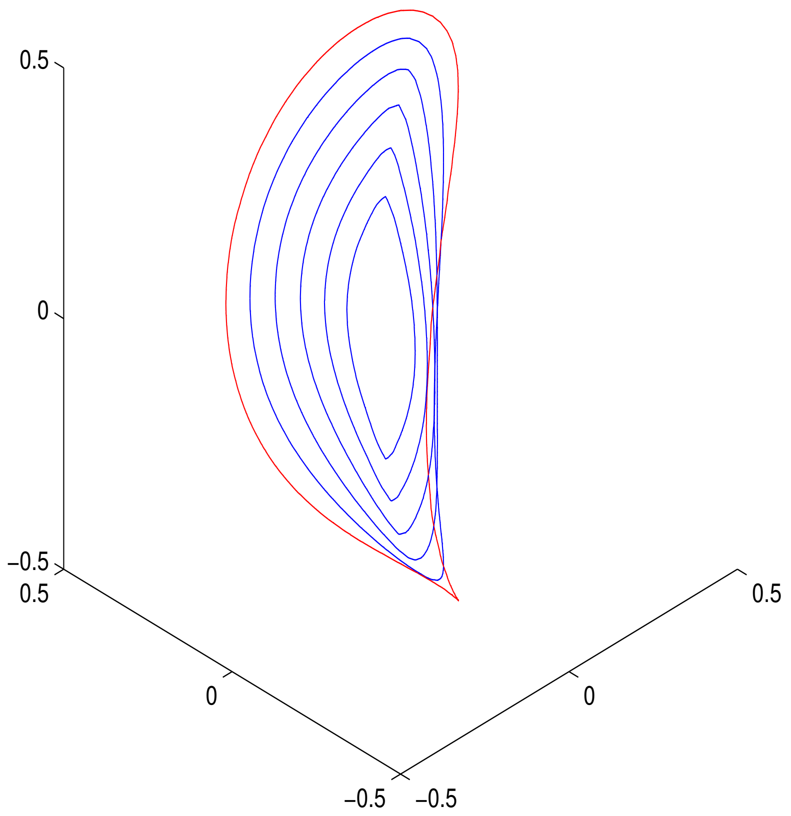

Figure 9: This shows the time evolution of a curve under constant flow in the normal direction. The curve is initially shaped like the boundary of a potato chip and shrinks thereafter. Note a kink forms in the curve at a certain time, an indication of merging.

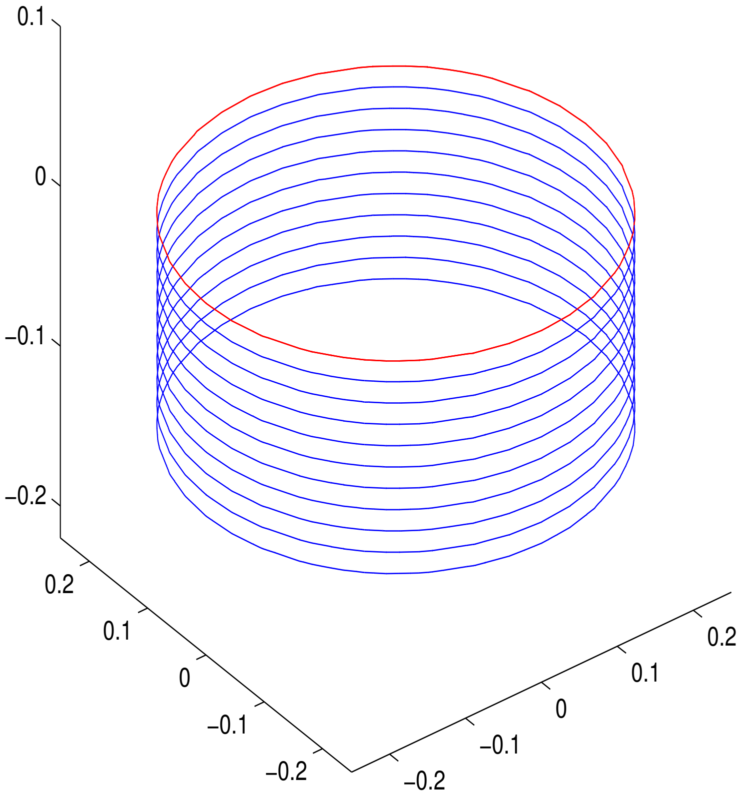

Figure 10: This shows the time evolution of a circle under constant flow in the binormal direction. Since the original curve lies on a plane, the evolution is simply translation in the normal direction of the plane. In this picture, the circle moves in the downward direction.

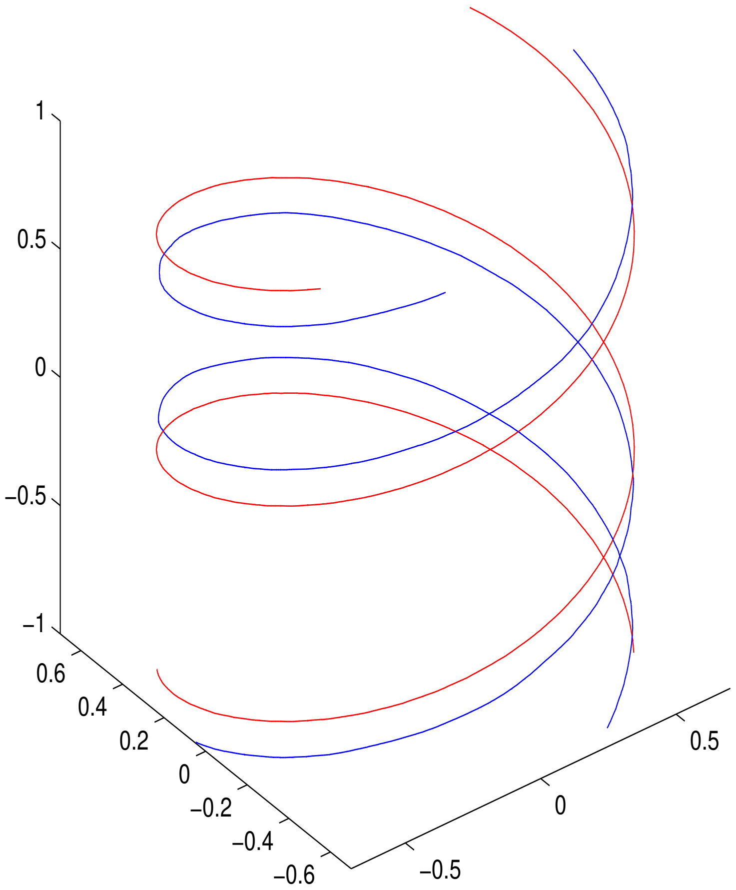

Figure 11: This shows the time evolution of two helices under constant flow in the binormal direction. Each helix rotates about its center axis but in opposite directions.

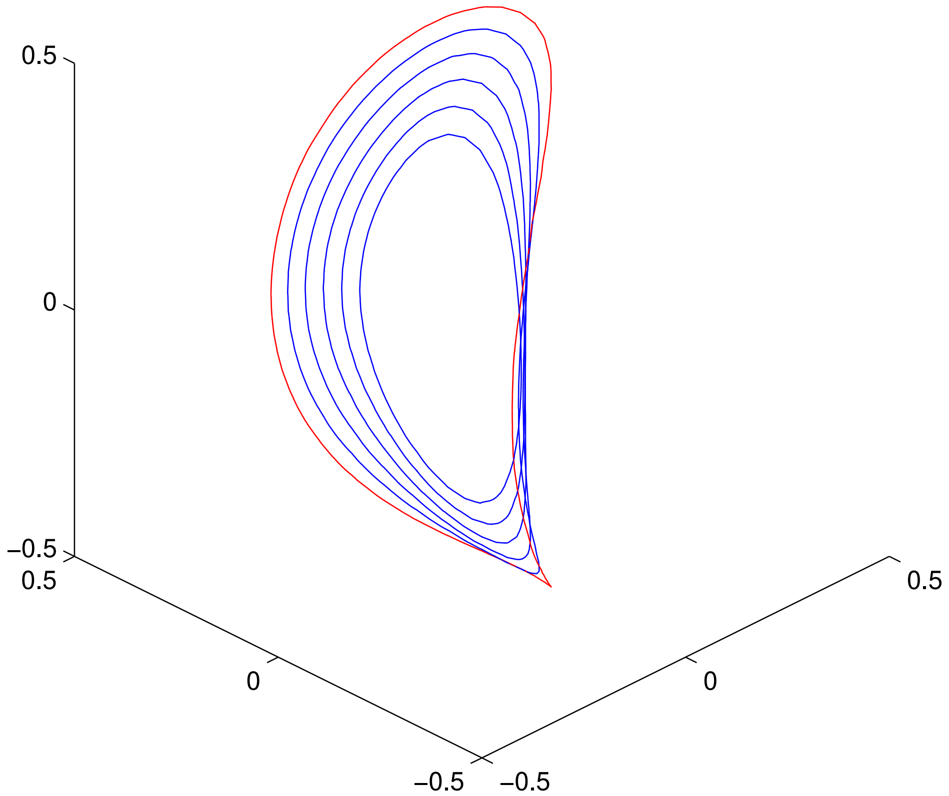

Figure 12: This shows the time evolution of an initial potato chip shaped curve by unit speed in the normal direction combined with 0.1 times curvature. Note a kink no longer forms due to adding the curvature term.