Threshold Dynamics for High Order Geometric Motions

Selim Esedoḡlu* Steven J. Ruuth† Richard Tsai‡

September 20, 2007

Abstract

In this paper, a class of algorithms for the high order geometric motion of planar curves is developed. The algorithms alternate two simple steps—a convolution and a thresholding step—to evolve planar curves according to combinations of Willmore flow, surface diffusion flow and curvature motion. A distinguishing feature of the methods is that they posses much better stability than typical explicit algorithms. Formal expansions and numerical examples are provided for a variety of high order flows to validate the methods and illustrate their behaviors.

1 Introduction

Higher order geometric motions of curves and surfaces arise in a variety of applications. They constitute a notoriously difficult class of computational problems. These motions typically correspond to the steepest descent for energies that are defined over curves and surfaces. For example, one of the algorithms proposed in this paper computes gradient descent for the energy

$$\int_C \frac{1}{2}\kappa^2 + \alpha\, ds \tag{1}$$

where C is a closed curve in the plane, κ is its curvature, and ds is the length element. Energy (1) is sometimes referred to as Euler's elastica energy. The resulting motion moves every point on the curve C by normal a velocity given by

$$v_n = -\kappa_{ss} - \frac{1}{2}\kappa^3 + \alpha\kappa$$

*Department of Mathematics, University of Michigan. Ann Arbor, MI 48109. **Email:** esedoglu@umich.edu. Research supported in part by NSF grant DMS-0605714.

†Corresponding author. Department of Mathematics, Simon Fraser University. Burnaby, British Columbia, V5A 1S6, Canada. **Email:** sruuth@sfu.ca. Research supported in part by a grant from NSERC Canada.

‡Department of Mathematics, University of Texas at Austin, TX 78712. **Email:** ytsai@math.utexas.edu. Research supported in part by the National Science Foundation under agreement No. DMS-0513394 and a Sloan Foundation Fellowship.

In Section 6.1 an algorithm is presented for a more general version of this motion. In Section 6.2, we propose an algorithm for *motion by surface diffusion* of a curve, which is given by the normal velocity

$$v_n = -\kappa_{ss}.$$

Computer vision and digital image processing are two application areas that would greatly benefit from improvements in computational methods for higher order geometric motion of curves and surfaces. Indeed, many important problems of image processing and computer vision involve minimizing curvature dependent functionals over curves and surfaces. For instance, the active contours (or "snakes") model [18] of Kass, Witkin, and Terzopoulos originally calls for minimizing an energy that includes the square of the curvature integrated along the curve. Another important example is the segmentation with depth involving disocclusion [20, 21] model of Nitzberg, Mumford, and Shiota [24, 10], where curvature dependent functionals are to be minimized in order to extract information about the three dimensional arrangement of objects making up a scene from a single two dimensional image of the scene. Yet another related problem of the field that leads to higher order geometric flows is the image inpainting problem of Bertalmio et. al., where the goal is to repair damaged regions in an image by connecting level lines of intensity using smooth curves [3].

There are a couple of techniques that are being applied and investigated for the type of high order geometric motions considered in this paper. Explicit, semi-implicit, and implicit surface algorithms have been introduced to track the evolution of the curves and surfaces, see for example, [1, 8, 9, 4, 12, 7, 6]. Due to the high order derivatives involved in the motions, stability conditions impose a severe efficiency drawback for explicit discetization methods. On the other hand, the nonlinear nature of the motions makes it difficult to develop implicit schemes; although there is significant progress in this direction [30], questions remain about whether usable implicit schemes perform quite as efficiently as what is required in applications such as image processing and vision.

In this paper, we will consider algorithms for computing a number of these geometric motions, and illustrate them on some of the mentioned applications. Our approach is based on extending to fourth order flows the *threshold dynamics* (or *diffusion generated motion*) idea of Merriman, Bence, and Osher [22, 23] proposed originally for motion by mean curvature. The algorithms have excellent stability properties. We believe this makes them particularly promising for rapid solution of problems in which steady states are sought; see [11] for an example in segmentation with depth problem of image processing.

2 Previous Work

In this section we recall the Merriman, Bence, Osher threshold dynamics for motion by mean curvature, and describe some recent work by Grzibovskis and Heintz on extending it to Willmore flow.

In [22, 23], Merriman, Bence, and Osher introduced a very interesting algorithm for approximating the motion of an interface by mean curvature motion. If we consider the interface to be the boundary

of a region $\Sigma \subset \mathbb{R}^N$, their idea is to alternate the following two steps: First, convolve the characteristic function of Σ with a Gaussian kernel. Then, threshold the resulting function at $\frac{1}{2}$. More precisely, their algorithm can be described as follows: Set $\Sigma^0 = \Sigma$. Alternate the following steps for $k = 0, 1, 2, \ldots$ in order to generate the sets Σ^k so that $\partial\Sigma^k$ approximate the motion of $\partial\Sigma^0$ by mean curvature at the discrete times $t_k = k \cdot \Delta t$:

1. Form the convolution:
$$K(x) := (\mathbf{1}_{\Sigma^k} * G_{\Delta t})(x)$$
where $G_{\Delta t}(x)$ is the fundamental solution of the heat equation in N space dimensions.

2. Set:
$$\Sigma^{k+1} = \left\{ x \in \mathbb{R}^N : K(x) \geq \frac{1}{2} \right\}$$

A major benefit of the algorithm is that both of the steps involved can be implemented very efficiently: The convolution can be done, for example, via the fast Fourier transform, and the second step – the thresholding – is trivial. A potential drawback of this algorithm is that the accuracy on a fixed grid is poor since there is no subcell resolution during the thresholding step. This limitation can be overcome using Fast Fourier transforms on grids adapted to the evolving interface [26]. The original MBO algorithm that we described above has been rigorously shown to converge to motion by mean curvature of $\partial\Sigma$ in [14, 2]. Several generalizations of the basic algorithm are given in [19, 25, 16, 27, 28, 29] and some applications to image segmentation are given in [17, 13].

In [15], Grzibovskis and Heintz propose a generalization of the MBO algorithm to Willmore flow for two dimensional surfaces in three space, which is a fourth order evolution of the interface obtained as gradient descent for the Willmore functional. In Section 4 we will give a particularly simple derivation of the Willmore flow algorithm in two dimensions in the special case where the convolution kernel is a Gaussian; we believe that numerically this is a particularly relevant choice for the kernel since the convolutional result is smooth and may be treated to high accuracy using the techniques in [26]. We concentrate on two dimensions since this is the particularly relevant setting for many applications, especially in image processing. In subsequent sections, this expansion will be utilized extensively to obtain algorithms for more general geometric motions. Derivation of the Willmore flow algorithm in 3D and with more general kernels can be found in [15]. That paper [15] also presents a selection of interesting 3D computations.

3 Preliminaries

The Willmore functional in two dimensions is given by

$$E(C) = \frac{1}{2} \int_C \kappa^2 \, ds \tag{2}$$

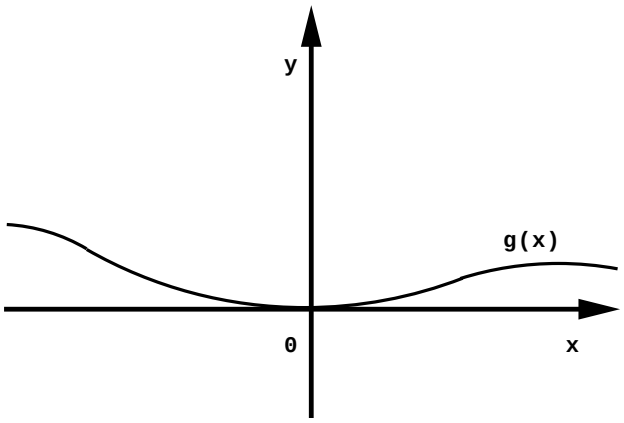

Figure 1: The initial interface.

where κ is the curvature of the curve C, and ds is the length element. Gradient descent for (2) moves the curve C with normal velocity given by

$$W = -\kappa_{ss} - \frac{1}{2}\kappa^3. \tag{3}$$

where s is the arclength parameter for the curve. See [31] for the derivation.

Consider a smooth interface which is given by the graph of the function $g(x)$, initially passing through the origin, where it is tangent to the x-axis, as in Figure 1, so that $g(0) = 0$ and $g'(0) = 0$. We'll be working with this curve given by the graph of $g(x)$ and the region of the plane that lies below this curve in the xy-plane, namely $\{(x,y) : y < g(x)\}$; our interface – the curve – is the boundary of this region. At the origin we have

$$g_{ss} = g_{xx} \text{ and } \kappa_{ss} = g_{xxxx} - 3\kappa^3$$

which implies that

$$W = -g_{xxxx} + \frac{5}{2}\kappa^3 \tag{4}$$

there. We shall see that this expression will arise in our truncation error analysis.

4 Truncation Error Analysis: Willmore Flow

This section derives a convolution-thresholding algorithm for Willmore flow in two dimensions. The idea is to consider the convolution of the characteristic function of the interior of a smooth curve (the region $\{(x,y) : y < g(x)\}$) with a suitable kernel. To achieve good computational efficiency we consider Gaussian kernels, but other radially symmetric kernels are also possible (cf. [15]).

First, define the function $F(x,y,t)$ to be the convolution of the characteristic function of the region below our curve with the fundamental solution of the heat equation at time t:

$$F(x,y,t) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \frac{1}{4\pi t} e^{-\frac{(x-\xi)^2+(y-\eta)^2}{4t}} \mathbf{1}_{\{(\xi,\eta):g(\xi)>\eta\}}(\xi,\eta)\, d\xi\, d\eta$$

As in [19], the value of $F(x,y,t)$ along the y-axis is given by

$$\begin{aligned}
F(0,y,t) =& \frac{1}{4\pi t}\int_{-\infty}^{\infty}\int_{-\infty}^{0} e^{-\frac{\xi^2+(y-\eta)^2}{4t}}\, d\eta\, d\xi + \frac{1}{4\pi t}\int_{-\infty}^{\infty}\int_{0}^{g(\xi)} e^{-\frac{\xi^2+(y-\eta)^2}{4t}}\, d\eta\, d\xi \\
=& \frac{1}{2\sqrt{\pi t}}\int_{-\infty}^{0} e^{-\frac{(y-\eta)^2}{4t}}\, d\eta + \frac{1}{4\pi t}\int_{-\infty}^{\infty}\int_{0}^{g(\xi)} e^{-\frac{\xi^2+(y-\eta)^2}{4t}}\, d\eta\, d\xi \\
=& \frac{1}{2\sqrt{\pi t}}\int_{0}^{\infty} e^{-\frac{\gamma^2}{4t}}\, d\gamma - \frac{1}{2\sqrt{\pi t}}\int_{0}^{y} e^{-\frac{\gamma^2}{4t}}\, d\gamma + \frac{1}{4\pi t}\int_{-\infty}^{\infty}\int_{0}^{g(\xi)} e^{-\frac{\xi^2+(y-\eta)^2}{4t}}\, d\eta\, d\xi \\
=& \frac{1}{2} - \frac{1}{2\sqrt{t\pi}}\int_{0}^{y} e^{-\gamma^2/4t}\, d\gamma + \frac{1}{4t\pi}\int_{-\infty}^{\infty} e^{-\xi^2/4t}\int_{0}^{g(\xi)} e^{-(y-\eta)^2/4t}\, d\eta\, d\xi
\end{aligned}$$

Expanding two of the exponentials around 0, and replacing $g(\xi)$ by the first few terms of its Taylor series at 0, and noting that we'll be looking at $F(0,y,t)$ at $y = O(t^2)$ gives the expansion:

$$\begin{aligned}
F(0,y,t) &= \frac{1}{2} - \frac{1}{2\sqrt{t\pi}}\int_{0}^{y}(1-\frac{\gamma^2}{4t})\, d\gamma + \\
&\quad \frac{1}{4t\pi}\int_{-\infty}^{\infty} e^{-\xi^2/4t}\int_{0}^{\frac{1}{2}g^{(2)}(0)\xi^2+\frac{1}{3!}g^{(3)}(0)\xi^3+\frac{1}{4!}g^{(4)}(0)\xi^4}\left[1-\frac{(y-\eta)^2}{4t}\right] d\eta\, d\xi + O(t^{5/2}), \\
&= \frac{1}{2} - \frac{1}{2\sqrt{t\pi}}y + \\
&\quad \frac{1}{4t\pi}\int_{-\infty}^{\infty} e^{-\xi^2/4t}\left[\eta + \frac{1}{12t}(y-\eta)^3\right]_{\eta=0}^{\eta=\frac{1}{2}g^{(2)}(0)\xi^2+\frac{1}{3!}g^{(3)}(0)\xi^3+\frac{1}{4!}g^{(4)}(0)\xi^4} d\xi + O(t^{5/2}), \\
&= \frac{1}{2} - \frac{1}{2\sqrt{t\pi}}y + \\
&\quad \frac{1}{4t\pi}\int_{-\infty}^{\infty} e^{-\xi^2/4t}\left[\eta + \frac{1}{12t}(-3y^2\eta + 3y\eta^2 - \eta^3)\right]_{\eta=0}^{\eta=\frac{1}{2}g^{(2)}\xi^2+\frac{1}{3!}g^{(3)}\xi^3+\frac{1}{4!}g^{(4)}\xi^4} d\xi + O(t^{5/2})
\end{aligned}$$

where we write $g^{(n)}$ for $g^{(n)}(0)$. Noting that terms which are odd powers of ξ cancel in the integral, and remembering that we are looking for $y = O(t^2)$, a large number of terms can either be eliminated or recognized to be $O(t^{\frac{5}{2}})$. One ends up with:

$$F(0,y,t) = \frac{1}{2} - \frac{1}{2\sqrt{\pi t}}y + \frac{1}{4\pi t}\int_{-\infty}^{\infty} e^{-\xi^2/4t}\left[\frac{1}{2}g^{(2)}\xi^2 + \frac{1}{4!}g^{(4)}\xi^4 - \frac{1}{96t}\left(g^{(2)}\right)^3\xi^6\right] d\xi + O(t^{\frac{5}{2}})$$

Using the integrals

$$\int_{-\infty}^{\infty} \xi^2 e^{-\xi^2}\, d\xi = \frac{1}{2}\sqrt{\pi},\ \int_{-\infty}^{\infty} \xi^4 e^{-\xi^2}\, d\xi = \frac{3}{4}\sqrt{\pi},\ \text{and}\ \int_{\infty}^{\infty} \xi^6 e^{-\xi^2}\, d\xi = \frac{15}{8}\sqrt{\pi}$$

and changing variables appropriately yields

$$F(0,y,t) = \frac{1}{2} - \frac{1}{2\sqrt{\pi}} y t^{-1/2} + \frac{1}{2\sqrt{\pi}} g^{(2)} t^{1/2} + \frac{1}{4\sqrt{\pi}} g^{(4)} t^{3/2} - \frac{5}{8\sqrt{\pi}} \left(g^{(2)}\right)^3 t^{3/2} + O(t^{5/2})$$

In terms of the curvature $\kappa = g^{(2)}$ and the Willmore flow speed W (see equation (4)) at $x = 0$ for the curve given by the graph of $g(x)$, the last equation reads:

$$F(0,y,t) = \frac{1}{2} - \frac{1}{2\sqrt{\pi}} y t^{-1/2} + \frac{1}{2\sqrt{\pi}} \kappa t^{1/2} - \frac{1}{4\sqrt{\pi}} W t^{3/2} + O(t^{5/2}) \tag{5}$$

An algorithm for Willmore flow can be obtained if we eliminate the curvature term in the expression above by evaluating $F(0,y,t)$ at two different values of t and forming the appropriate linear combination:

$$t_2^{1/2} F(0,y,t_1) = \frac{t_2^{1/2}}{2} - \frac{1}{2\sqrt{\pi}} \left(\frac{t_2}{t_1}\right)^{1/2} y + \frac{1}{2\sqrt{\pi}} (t_1 t_2)^{1/2} \kappa - \frac{1}{4\sqrt{\pi}} t_1^{3/2} t_2^{1/2} W + O(t_1^3)$$

$$t_1^{1/2} F(0,y,t_2) = \frac{t_1^{1/2}}{2} - \frac{1}{2\sqrt{\pi}} \left(\frac{t_1}{t_2}\right)^{1/2} y + \frac{1}{2\sqrt{\pi}} (t_1 t_2)^{1/2} \kappa - \frac{1}{4\sqrt{\pi}} t_2^{3/2} t_1^{1/2} W + O(t_2^3)$$

Subtracting, one gets:

$$\begin{aligned} \sqrt{t_2} F(0,y,t_1) &- \sqrt{t_1} F(0,y,t_2) - \sqrt{t_2}/2 + \sqrt{t_1}/2 \\ &= \frac{1}{2\sqrt{\pi}} \left(\sqrt{\frac{t_1}{t_2}} - \sqrt{\frac{t_2}{t_1}}\right) y - \frac{1}{4\sqrt{\pi}} \left(\sqrt{t_2} t_1^{3/2} - \sqrt{t_1} t_2^{3/2}\right) W + O(t_1^3) + O(t_2^3) \end{aligned} \tag{6}$$

Choose a time step-size $\Delta t > 0$ and a positive factor $\theta \neq 1$, and set $t_1 = \sqrt{2\Delta t}/\theta$ and $t_2 = \theta\sqrt{2\Delta t}$. With these choices, the formula above reads:

$$\begin{aligned} \sqrt{\theta}(2\Delta t)^{\frac{1}{4}} F(0,y,\sqrt{2\Delta t}/\theta) - \frac{(2\Delta t)^{\frac{1}{4}}}{\sqrt{\theta}} F(0,y,\theta\sqrt{2\Delta t}) - \frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta} - 1/\sqrt{\theta}) \\ = \frac{1}{2\sqrt{\pi}} \left(\frac{1}{\theta} - \theta\right) y - \frac{2\Delta t}{4\sqrt{\pi}} \left(\frac{1}{\theta} - \theta\right) W + O(\Delta t^{\frac{3}{2}}) \end{aligned}$$

Hence, if the expression

$$A(x,y) := \sqrt{\theta}(2\Delta t)^{\frac{1}{4}} F(x,y,\sqrt{2\Delta t}/\theta) - \frac{(2\Delta t)^{\frac{1}{4}}}{\sqrt{\theta}} F(x,y,\theta\sqrt{2\Delta t}) \tag{7}$$

is formed and thresholded at $\frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta}-1/\sqrt{\theta})$, the resulting curve would intersect the y-axis at a y value such that:

$$\frac{1}{2\sqrt{\pi}}\left(\frac{1}{\theta}-\theta\right)y-\frac{2\Delta t}{4\sqrt{\pi}}\left(\frac{1}{\theta}-\theta\right)W+O(\Delta t^{\frac{3}{2}})=0.$$

The above value of the threshold was chosen so that constant terms of order $o(\Delta t^{\frac{1}{4}})$ that appear in the linear combination of the kernels drop out. Simplifying, we see that the solution is:

$$y=\Delta t W+O(\Delta t^{\frac{3}{2}})$$

First of all, note that $y=O(\Delta t)=O(t^2)$, as required by our expansion. Second, we see that the curve moves with the following normal speed:

$$v_n=W$$

during this time step of size Δt, with error $O(\sqrt{\Delta t})$.

5 Implementation

Based on the discussion of the last section, the resulting algorithm for Willmore flow can be described as follows: Let $\Sigma\subset\mathbb{R}^2$ denote the set whose boundary is to be moved via Willmore flow. Set $\Sigma^0:=\Sigma$. Alternate the following steps for $k=0,1,2,\ldots$ in order to generate the approximations Σ^k to the flow at the discrete times $t_k=k\cdot\Delta t$:

1. Form the convolution:

$$A(x,y):=\mathbf{1}_{\Sigma^k}(x,y)*(2\Delta t)^{\frac{1}{4}}\left(\sqrt{\theta}G_{\sqrt{2\Delta t}/\theta}(x,y)-\frac{1}{\sqrt{\theta}}G_{\theta\sqrt{2\Delta t}}(x,y)\right)$$

   where $G_t(x,y)$ is the fundamental solution of the heat equation:

$$G_t(x,y)=\frac{1}{4\pi t}e^{-\frac{x^2+y^2}{4t}}$$

2. Set:

$$\Sigma^{k+1}=\left\{(x,y)\in\mathbb{R}^2 : A(x,y)\geq\frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta}-1/\sqrt{\theta})\right\}$$

Convolution with a Gaussian kernel eliminates the high frequencies that are present in the initial characteristic function. Since this is a linear convolution, the different modes do not interact and there is *never*

a need to approximate the high frequency modes when using a Gaussian kernel. This idea was exploited in [26] to give an efficient spectral discretization of the MBO method, and in [15] to give an efficient implementation of their algorithm for Willmore flow. However, analogues of the algorithm using other radially symmetric kernels can also be obtained. The original algorithm of Grzibovskis and Heintz consists of two linear convolutions and a thresholding based on a difference. We have preferred to (equivalently) construct the kernel corresponding to the difference and convolve this with the characteristic function for the region, since this way only one convolution is required using the combined kernel.

Various values of the parameter θ can be chosen, but notice that $\theta > 1$ is required for the consistency of the resulting algorithm. This implies that the combined kernel used in the convolution step of the algorithm will be negative sufficiently far away from the origin. The negativity discourages mergers (topological changes) from taking place as two interfaces approach one another. Interestingly, this behavior agrees with the underlying Willmore flow of curves since the Willmore functional (2) would be expected to become infinite at such transitions.

6 More General Motions

The expansion obtained in Section 4 can be utilized to easily derive threshold based algorithms for more general velocities than that of Willmore flow.

6.1 Willmore with lower order terms

In this section, we will consider interface motions with normal velocities of the form:

$$v_n = c_0(x,y,t) + c_1(x,y,t)\kappa + c_2(t)W$$

where the coefficients involved may be non-constant both in space and in time. Since without the higher order Willmore term, motions of this type have been previously considered using threshold type schemes, we concentrate on the case when c_2 does not vanish, i.e. $c_2(t) > 0$ for all t. But then, since one can always rescale time, we may as well consider only velocities of the form:

$$v_n = c_0(x,y,t) + c_1(x,y,t)\kappa + W \tag{8}$$

One application area where velocities of type (8) arise is computer vision. More specifically, in the segmentation with depth model of Nitzberg, Mumford, and Shiota [24], as well as in digital image inpainting models inspired by that work [3, 5], it is necessary to minimize an energy of the form $\int_C \alpha + (1/2)\beta\kappa^2\, d\sigma$ over curves C, which leads to the normal velocity $v = \alpha\kappa + \beta W$ for a curve evolving via gradient descent for that energy.

As in the previous section, the idea is to take the correct linear combination between two different Gaussians to form the convolution kernel (but, again, the results can be extended to combinations of other non-negative, radially symmetric kernels). In other words, using the same notation as in Section 4, we will once again consider combinations of the form

$$\alpha F(x,y,\sqrt{2\Delta t}/\theta)+\beta F(x,y,\theta\sqrt{2\Delta t})+\gamma$$

First, we eliminate the $O(\Delta t^{-\frac{1}{4}})$ term that involves y in expansion (5):

$$\theta^{-\frac{1}{2}}\left(F(0,y,\sqrt{2\Delta t}/\theta)-\frac{1}{2}\right)-\theta^{\frac{1}{2}}\left(F(0,y,\theta\sqrt{2\Delta t})-\frac{1}{2}\right)=\frac{\kappa}{2\sqrt{\pi}}(1/\theta-\theta)(2\Delta t)^{\frac{1}{4}}+O(\Delta t^{\frac{3}{4}})$$

Hence, if we set

$$B(x,z):=\theta^{-\frac{1}{2}}\left(F(x,y,\sqrt{2\Delta t}/\theta)-\frac{1}{2}\right)-\theta^{\frac{1}{2}}\left(F(x,y,\theta\sqrt{2\Delta t})-\frac{1}{2}\right) \tag{9}$$

we can then form the combination:

$$\begin{aligned}A(x,y)-\Delta t(2\Delta t)^{-\frac{1}{4}}c_1(x,y,t)B(x,y)-\frac{\Delta t}{2\sqrt{\pi}}(1/\theta-\theta)c_0(x,y,t)-\frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta}-1/\sqrt{\theta})\\ =\frac{1}{2\sqrt{\pi}}(1/\theta-\theta)\left\{y-\Delta t\left[W+c_1(x,y,t)\kappa+c_0(x,y,t)\right]\right\}+O(\Delta t^{\frac{3}{2}})\end{aligned}$$

This last formula gives us a thresholding algorithm for normal velocities of the form (8), with error $O(\sqrt{\Delta t})$. It consists in the following: Let $\Sigma\subset\mathbb{R}^2$ denote the set whose boundary is to be moved via the geometric motion described by (8). Set $\Sigma^0:=\Sigma$. Alternate the following steps for $k=0,1,2,\ldots$ in order to generate the approximations Σ^k to the flow at the discrete times $t=k\cdot\Delta t$:

1. Form the combination:

$$C(x,y):=A(x,y)-\Delta t(2\Delta t)^{-\frac{1}{4}}c_1(x,y,t)B(x,y)-\frac{\Delta t}{2\sqrt{\pi}}(1/\theta-\theta)c_0(x,y,t)$$

   where $A(x,y)$ is as in (7) and $B(x,y)$ is as in (9).

2. Set:

$$\Sigma^{k+1}=\left\{(x,y)\in\mathbb{R}^2:C(x,y)\geq\frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta}-1/\sqrt{\theta})\right\}$$

Note that by utilizing the expansion of Section 4 and the approach of this and the previous section, one can devise more elaborate linear combinations of kernels in order to kill off leading order error terms (i.e. Richardson extrapolation) and thus obtain thresholding algorithms with higher order accuracy.

6.2 Surface Diffusion

If we break away from simple linear combinations of convolutions, we can also obtain thresholding algorithms for other important fourth order flows. In this section, we will extend our results to motion by surface Laplacian of curvature for a curve, which is an important geometric motion that arises in a number of applications. The normal velocity takes the form

$$v_n = -\kappa_{ss},$$

where s is the arclength and κ is the curvature of the curve.

In Section 4 we presented a threshold algorithm for Willmore flow. Our approach to surface diffusion is to utilize the expansion in that section, in particular the term $A(x,y)$ defined in (7), and to modify the Willmore flow algorithm so that surface diffusion flow is approximated instead. The normal velocity for Willmore flow is given by equation (3). It differs from surface diffusion velocity by the term $\frac{1}{2}\kappa^3$. We can use the term $B(x,y)$ (equation (9) from Section 6.1 to compensate for the difference. Indeed, we have:

$$(2\Delta t)^{\frac{1}{4}} B^3(0,y) = \frac{1}{4\pi^{\frac{3}{2}}}(1/\theta - \theta)^3 \kappa^3(\Delta t) + O(\Delta t^{\frac{3}{2}})$$

Hence, we can form the combination:

$$A(0,y) + \frac{\pi}{(1/\theta-\theta)^2}(2\Delta t)^{\frac{1}{4}} B^3(0,y) - \frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta} - 1/\sqrt{\theta}) = \frac{1}{2\sqrt{\pi}}(1/\theta - \theta)\{y - \Delta t \cdot \kappa_{ss}\} + O(\Delta t^{\frac{3}{2}})$$

This leads to the following algorithm for the motion of a curve by surface diffusion: Let $\Sigma \subset \mathbb{R}^2$ denote the set whose boundary is to be evolved according to surface diffusion. Set $\Sigma^0 := \Sigma$. Alternate the following steps for $k = 0, 1, 2, \ldots$ in order to generate the approximations Σ^k to the flow at the discrete times $t_k = k \cdot \Delta t$:

1. Form the combination:

$$D(x,y) := A(0,y) + \frac{\pi}{(1/\theta-\theta)^2}(2\Delta t)^{\frac{1}{4}} B^3(0,y)$$

   where $A(x,y)$ is as in (7) and $B(x,y)$ is as in (9).

2. Set:

$$\Sigma^{k+1} = \left\{ (x,y) \in \mathbb{R}^2 : D(x,y) \geq \frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta} - 1/\sqrt{\theta}) \right\}$$

This algorithm differs from the standard convolution generated motions in that the first step – i.e. the convolution step – involves a nonlinear expression. Nevertheless, both steps of the algorithm maintain the efficiency of their counterparts in the standard MBO scheme, since the nonlinear operation involved in the convolution step is pointwise. We anticipate that designing such an algorithm for surface diffusion in 3D may be more involved because of the appearance of a Gaussian curvature term in Willmore velocity.

7 Numerical Experiments for Willmore Flow

We now provide some experiments to verify the numerical convergence of our Willmore flow algorithm. Following [26], FFTs on unequally spaced grids are used to carry out the spatial discretization. These adaptive methods lead to results which are essentially free of spatial discretization error, allowing us to better focus on the behavior of the underlying semi-discrete method. Typical values for the adaptive spatial discretization were 256×256 basis functions and a finest grid cell size of 2^{-14}. For our time step choices, this gives kernel sizes that are much larger than the finest grid cell size.

7.1 Expanding Circle

For our first example, consider the evolution of a circle according to Willmore flow. By symmetry, the curve will remain a circle and will evolve according to the ordinary differential equation

$$\dot{R} = \frac{1}{2}\left(\frac{1}{R}\right)^3.$$

From this differential equation, it is straightforward to determine the radius at any time t:

$$R(t) = \sqrt[4]{R(0)^4 + 2t}.$$

Our tests set the initial radius equal to 0.2 and evolve for a time $T = 0.0025$. In this and subsequent examples, the time step-size was chosen sufficiently fine to give a clear indication of the convergence rate. This leads to the absolute errors and numerically observed convergence rates given in Table 1 (all measured in terms of area). The expected convergence rate of $O(\sqrt{\Delta t})$ is clearly observed.

Δt	*Error*	*Conv. Rate*
$T/256$	0.01918	-
$T/512$	0.01277	0.587
$T/1024$	0.00864	0.562
$T/2048$	0.00595	0.540
$T/4096$	0.00413	0.527

Table 1. Absolute errors in the final area of the expanding circle.

Remark: This test set $\theta = (1.001)^2$. Ignoring the numerical cancellation issues discussed in Section 7.2, values of θ close to 1 are expected to produce better accuracy since such values correspond to kernels with smaller support. Our next example examines this choice more closely.

7.2 Evolving Polynomial

Our second example approximates the initial velocity at the origin for the curve

$$y = \frac{1}{2}d_2x^2 + \frac{1}{3!}d_3x^3 + \frac{1}{4!}d_4x^4.$$

Since the curve is initially a polynomial, it is easily shown that the initial velocity at the origin is

$$v_n = \frac{5}{2}d_2 - d_4.$$

For simplicity our experiments set $d_2 = d_3 = d_4 = 1$, although qualitatively similar results were observed for other choices as well.

Our first task is to see how θ affects the accuracy of the algorithm, so we fix $\Delta t = T/400$ and vary θ. This produces the relative errors displayed in Table 2. These results illustrate our earlier conjecture: kernels with smaller support (corresponding to θ-values close to 1) have smaller errors, provided roundoff errors do not dominate.

θ	*Error in velocity*
$(2)^2$	24.6
$(1.125)^2$	0.148
$(1.01)^2$	0.074
$(1.001)^2$	0.074
$(1.0001)^2$	0.074
$(1.0000001)^2$	0.074
$(1.00000000000001)^2$	5.933

Table 2. Relative errors in initial velocity for various θ and $\Delta t = T/400$.

Directing our attention now to the numerical convergence of our algorithm, we take $\theta = (1.001)^2$ and vary Δt. This gives the relative errors and numerically observed convergence rates displayed in Table 3 (all measured in terms of velocity at the origin).

Δt	*Error in velocity*	*Conv. Rate*
$T/400$	0.0735	-
$T/800$	0.0258	1.51
$T/1600$	0.0187	0.46
$T/3200$	0.0135	0.47
$T/6400$	0.0097	0.48
$T/12800$	0.0068	0.50

Table 3. Relative errors in initial velocity for the evolving polynomial.

Similar to the case of the expanding circle, the expected convergence rate of $O(\sqrt{\Delta t})$ is clearly observed.

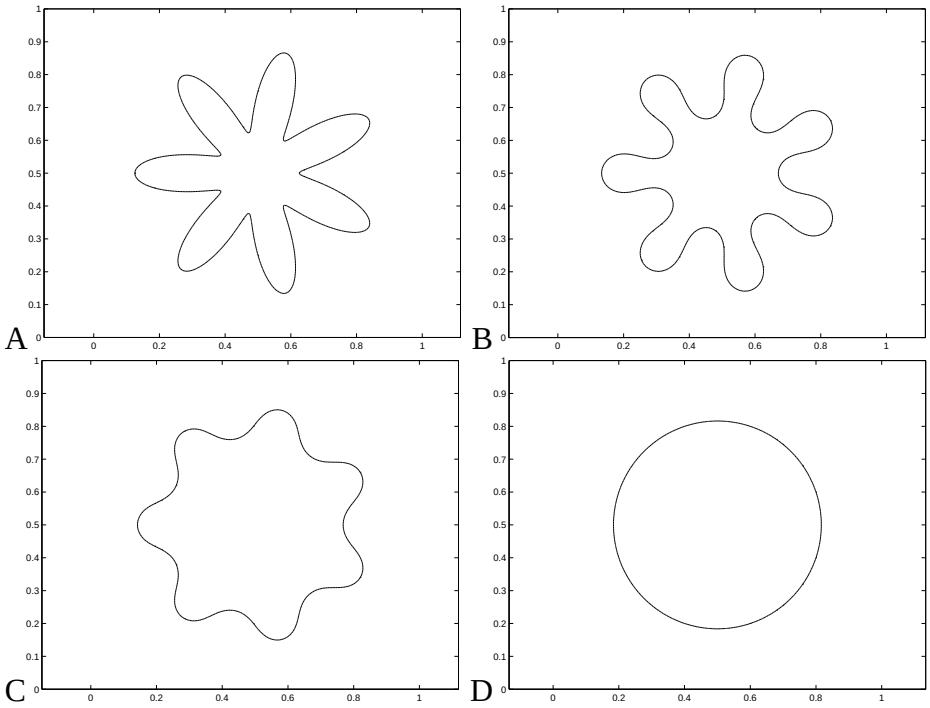

Figure 2: A flower-shaped curve evolving under Willmore flow. A: Initial curve ($t = 0$). B: Computed solution at $t = 0.001/256$. C: Computed solution at $t = 0.010/256$. D: Final contour is very closely a circle ($t = 0.030/256$). The time step-size was $\Delta t = 0.001/(50 \times 256)$.

7.3 Evolution of a Flower-Shaped Region

A further example of Willmore flow, this time with a more interesting, flower-shaped curve as initial data, is given in Figure 2. Figure 3 shows computations with the same initial data but with much larger choices of time step-size. As can be seen, the method is free of numerical instabilities even when the time step is so large that the first plotted stage of the previous calculation is reached in a single step. This unconditional stability property was observed in all the motions we investigated.

8 Numerical Experiments for Generalized Motions

This section focuses on the application of the algorithm to more general flows. Similar to the previous section, all spatial discretizations are carried out using the methods described in [26].

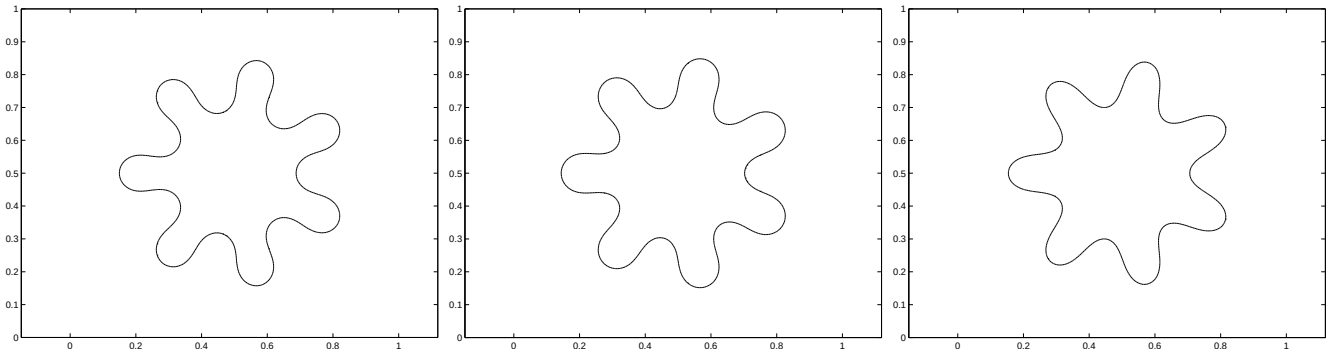

Figure 3: Illustration of the stability properties of the proposed Willmore flow algorithm: Arbitrarily large time steps can be taken, with accuracy considerations being the only constraint. The curves, from left to right, represent the computed solution at a fixed time, starting from the same initial data as in the experiment of Figure 2, using 50, 5, and 1 time steps.

8.1 Combined Motion

To examine the numerical behavior of the general algorithm, we consider the motion law

$$v_n = -10 + \pi\kappa + 2W$$

and approximate the initial numerical velocity at the origin for the polynomial interface

$$y = \frac{1}{2}x^2 + \frac{1}{3!}x^3 + \frac{1}{4!}x^4. \tag{10}$$

Taking $\theta = (1.001)^2$ and varying Δt gives the relative errors and numerically observed convergence rates displayed in Table 4.

Δt	*Error in velocity*	*Conv. Rate*
T/800	0.0575	-
T/1600	0.0204	1.50
T/3200	0.0148	0.46
T/6400	0.0107	0.47
T/12800	0.0077	0.48
T/25600	0.0055	0.49

Table 4. Relative errors in initial velocity for the evolving polynomial, and a general motion.

The expected convergence rate of $O(\sqrt{\Delta t})$ is clearly observed.

8.2 Surface Diffusion Flow

Section 6.2 gives an extension of the basic algorithm which can be used to treat surface diffusion or intrinsic Laplacian of curvature flows. To understand the numerical behavior of this algorithm, we approximate the initial velocity at the origin for the polynomial interface (10). This is compared to the (analytically derived) exact result.

Taking $\theta = (1.001)^2$ and varying Δt gives the relative errors and numerical convergence rates displayed in Table 5. The expected convergence rate of approximately $O(\sqrt{\Delta t})$ is again observed.

Δt	*Error in velocity*	*Conv. Rate*
T/800	0.0165	-
T/1600	0.0120	0.46
T/3200	0.0087	0.46
T/6400	0.0063	0.47
T/12800	0.0045	0.49
T/25600	0.0031	0.51

Table 5. Relative errors in initial velocity for the evolving polynomial, and a general motion.

Another example for surface diffusion flow is provided in Figure 4. This test case evolves an initial ellipse until it becomes nearly circular. The results obtained are clearly stable in the sense that there are no spurious oscillations, even with very large time steps that far exceed the CFL condition of explicit schemes. We also find that the relative error in the area enclosed by the final curve is less than 1.5%. (The exact value of the final area is known from the fact that surface diffusion flow preserves the area enclosed by curves.) In fact, arbitrarily large time steps lead to regular evolutions and no observed instabilities.

For a further example of surface diffusion flow, we again consider the flower-shaped curve as initial data. See Figure 5 and compare to a similar computation in [30]. Figure 6 shows computations with the same initial data but with much larger choices of time step-size. Similar to the case of Willmore flow, the method is free of numerical instabilities even when the time step is so large that the first plotted stage of the previous calculation is reached in a single step.

8.3 Reconstructed Shape

In [11], it was shown that threshold dynamics can be applied to minimize the energy (1) and thereby reconstruct missing (or occluded) parts of shapes described by their characteristic functions. That paper evolved the shape by sequentially carrying out a step of Willmore flow, followed by a step of the Merriman, Bence, Osher algorithm for curvature motion.

Using the approach for general flows described in Section 6.1, these two steps can be combined into one. The corresponding simplified algorithm takes the following form: Let D be the image region, $\tilde{D}$ be a

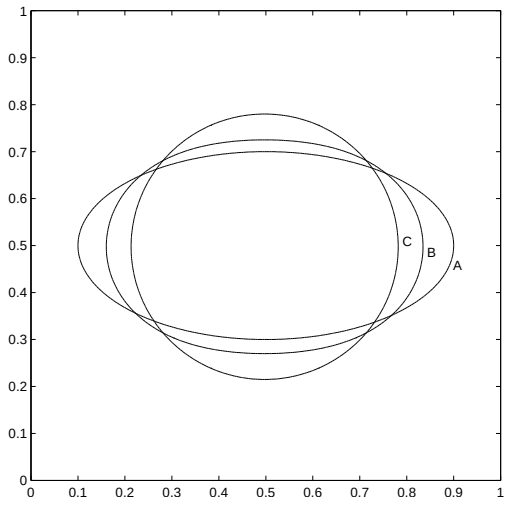

Figure 4: Ellipse evolving under surface diffusion flow. A: Initial ellipse ($t = 0$). B: Intermediate contour ($t = 0.0003125$). C: Final contour approximates a circle ($t = 0.0025$). The time step was taken to be $\Delta t = 0.0025/512$.

subdomain of D where image (i.e. shape) information is known to be missing and $\Sigma^0 \subset D$ denote the initial guess for the shape. By alternating the following steps, approximations $\Sigma^k, k = 0,1,2,\ldots$ to the shape can be generated until a steady solution is obtained:

1. Evolution step: Form the combination

$$C(x,y) := A(x,y) - \Delta t(2\Delta t)^{-\frac{1}{4}}\alpha B(x,y)$$

   where $A(x,y)$ is as in (7) and $B(x,y)$ is as in (9).

2. Fidelity step: Set

$$\Sigma^{k+1} = \left\{(x,y) \in \tilde{D} : C(x,y) \geq \frac{(2\Delta t)^{\frac{1}{4}}}{2}(\sqrt{\theta} - 1/\sqrt{\theta})\right\} \bigcup \left(\Sigma^0 \cap \left(D \setminus \tilde{D}\right)\right)$$

An example illustrating the algorithm appears in Figure 7. The algorithm effectively imposes Dirichlet boundary conditions on $\partial\tilde{D}$ by forcing the solution to coincide with the known image information outside of $\tilde{D}$ at the end of each time step. This is a convenient but not necessarily the most accurate treatment of boundaries; indeed, when Δt is taken to be larger, as shown in the third plot of Figure 7, kinks may be introduced into the solution at the boundary.

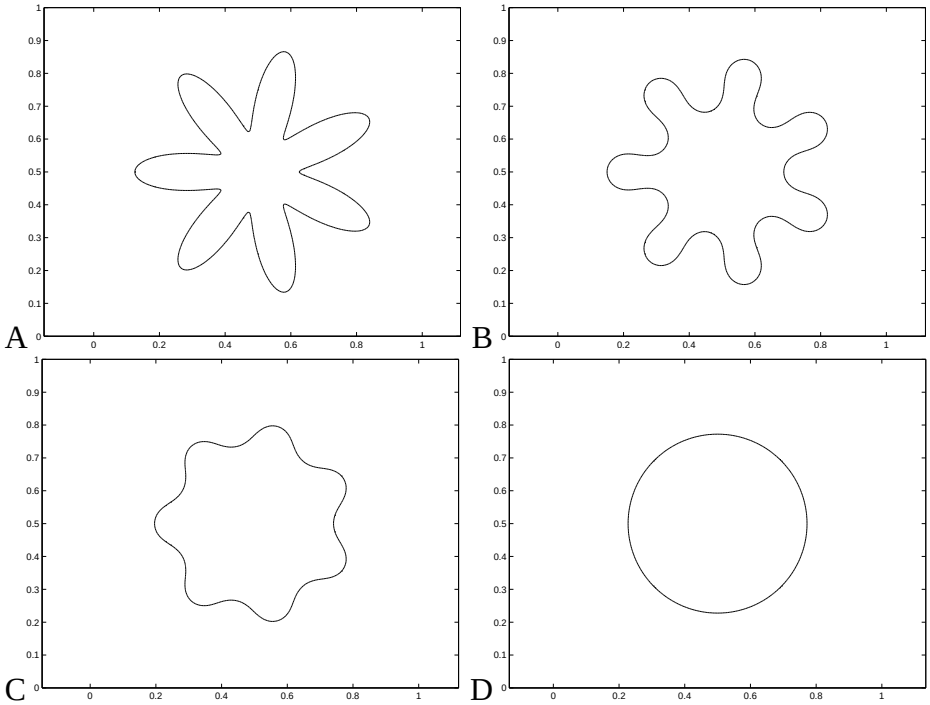

Figure 5: A flower-shaped curve evolving under surface diffusion flow. A: Initial curve ($t = 0$). B: Computed solution at $t = 0.001/256$. C: Computed solution at $t = 0.003/256$. D: Final contour is very closely a circle ($t = 0.04/256$). The time step-size was $\Delta t = 0.001/(50 \times 256)$. Compare to a similar computation in [30].

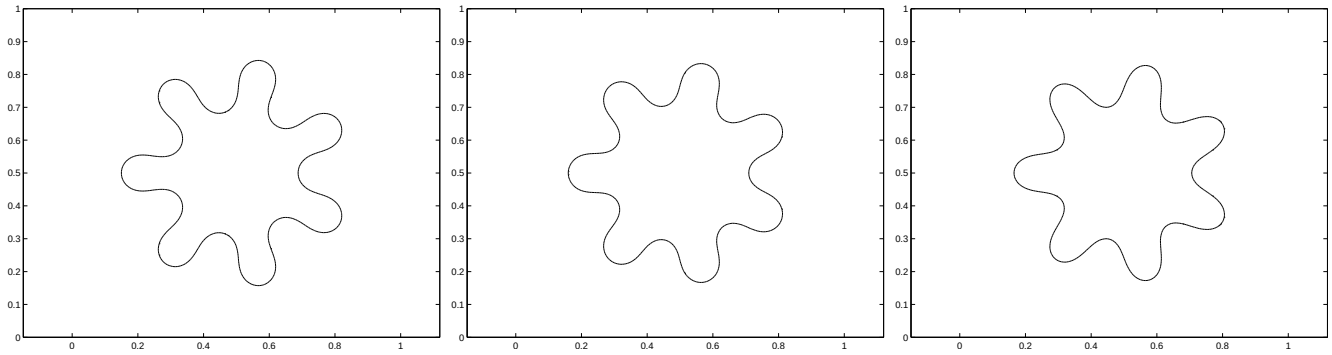

Figure 6: Illustration of the stability properties of the proposed surface diffusion algorithm: Arbitrarily large time steps can be taken, with accuracy considerations being the only constraint. The curves, from left to right, represent the computed solution at a fixed time, starting from the same initial data as in the experiment of Figure 5, using 50, 5, and 1 time steps.

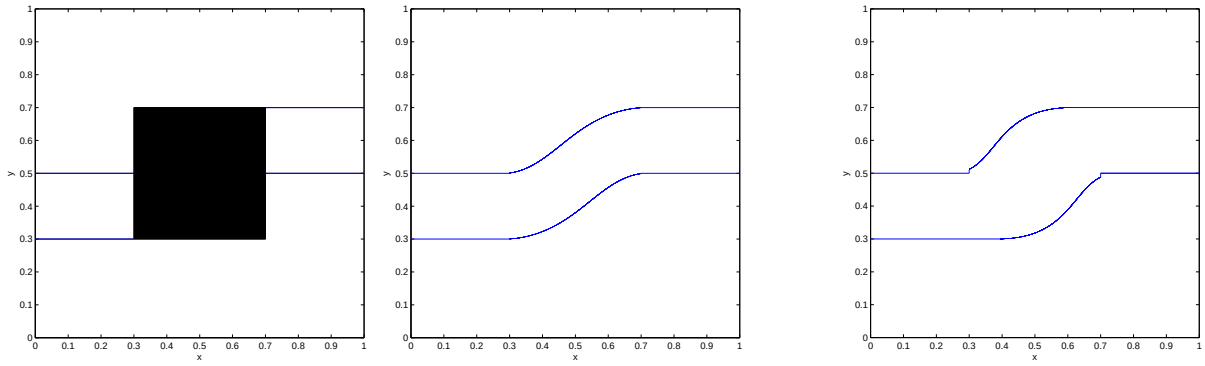

Figure 7: Image reconstruction example. The central square is the damaged region in the left image. Taking the initial region Σ^0 to be the union of the missing region and the two rectangular subregions and $\alpha = 1$ leads to the reconstructed bent bar shown in the second image from the left (time $t = 0.05$); this result was very carefully computed with time step-size $0.00001/1024$ using 128×128 basis functions. The rightmost was obtained by repeating the calculation with 1000 time steps and a time step-size of 0.0000005. Larger time steps did not produce oscillations in the missing region, $\tilde{D}$, but did lead to a further loss of accuracy.

8.4 Junctions

Similar to the original threshold dynamics of Merriman, Bence and Osher [22, 23], the algorithm for Willmore flow extends naturally to multiple junctions even though the asymptotic expansions require smoothness of the interfaces. This extension involves convolving the characteristic function for each region with the appropriate kernel, and thresholding according to the maximum point-wise value of the convolutions. More precisely, let $\Sigma_i \subset \mathbb{R}^2$ denote the set of the ith region ($2 \leq i \leq N$) whose boundary is to be moved via Willmore flow and set $\Sigma_i^0 := \Sigma_i$. We then alternate the following steps for $k = 0, 1, 2, \ldots$ in order to generate the approximations Σ_i^k to the flow at the discrete times $t_k = k \cdot \Delta t$:

1. Form the convolution:

$$A_i(x,y) := \mathbf{1}_{\Sigma_i^k}(x,y) * (2\Delta t)^{\frac{1}{4}} \left(\sqrt{\theta} G_{\sqrt{2\Delta t}/\theta}(x,y) - \frac{1}{\sqrt{\theta}} G_{\theta\sqrt{2\Delta t}}(x,y) \right), 1 \leq i \leq N$$

   where $G_t(x,y)$ is the fundamental solution of the heat equation:

$$G_t(x,y) = \frac{1}{4\pi t} e^{-\frac{x^2+y^2}{4t}}$$

2. Set:

$$\Sigma_i^{k+1} = \left\{ (x,y) \in \mathbb{R}^2 : A_i(x,y) \geq A_j(x,y), i \neq j \right\}, 1 \leq i \leq N$$

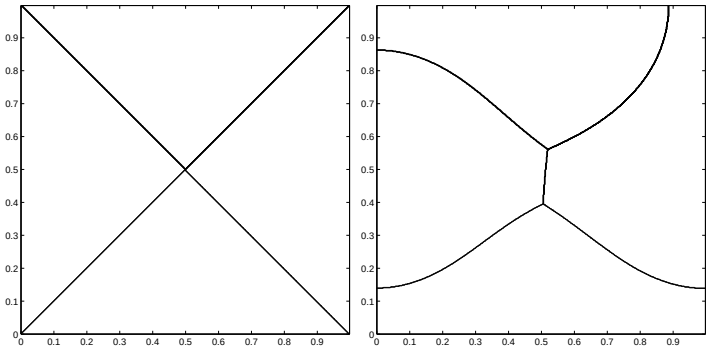

Figure 8: Evolution according to Willmore flow. Unstable junction at time $t = 0$ (left) breaks into 2 symmetric triple junctions by time $t = 0.01$ (right). The time step-size was taken to be $\Delta t = 0.01/10000$.

An illustration of junctions evolving by Willmore flow is given in Figure 8. The initial system is taken to be 4 regions meeting at a point. The symmetry of this configuration suggests that it is an unstable equilibrium: different nearby initial data would be expected to diverge. Our numerical algorithm perturbs this state into a lower energy configuration, and leads to the evolution shown. Note that our simulation makes use of homogeneous Neumann boundary conditions; this explains why curves meet the boundary at right angles in the final configuration. We are not aware of any analytical work concerning Willmore flow of junctions.

9 Challenges

An important issue is the width of the kernels used. Wide kernels lead to nonlocal effects in the resulting motion, by coupling the flow of disparate parts of the interface to each other. This source of error becomes particularly acute as the distance between two different parts of the interface becomes comparable to the width of the kernel, which happens for example when the interface is about to touch itself. In such a scenario, one can no longer think of the interface as locally the graph of a function, and our asymptotic expansions become inaccurate. In order to compute faithfully the motion at such difficult times, one is forced to take very narrow kernels, which in effect translates into taking very small time steps. However, even with small kernels we have found that this effect can prevent topological changes from taking place in both Willmore and surface diffusion flows. Energy considerations suggest topological changes of curves under Willmore flow would be discouraged; however, no such energetic impediment is apparent for surface diffusion flow.

10 Conclusions

In this paper threshold dynamics algorithms for the high-order geometric motion of curves are investigated. Our work continues earlier studies by Grzibovskis and Heintz [15] to give two dimensional Willmore flow plus lower order terms as well as methods for surface diffusion flow. Asymptotic expansions and numerical convergence studies are carried out throughout the paper to validate the methods.

Our methods have several characteristic properties. These include a very simple structure which consists of alternating convolution and thresholding steps. Our methods also have excellent stability properties, and in practice did not exhibit any apparent instabilities.

From a mathematical perspective, this article as well as earlier studies [15, 19, 25, 16, 27, 28, 29] hint at the diversity of motions possible under threshold dynamics and we feel that rigorous mathematical studies of such motions would be of great interest. We are also optimistic about the practical uses of such schemes and note that convolution and thresholding are simple operations that have the potential to be treated rapidly at the level of the graphics processing unit (GPU). Given the excellent stability properties exhibited by the algorithms, and the diversity of motion laws that are possible, we consider this class of algorithms to be interesting candidates for the treatment of certain geometric motion laws arising in applications such as image processing.

Acknowledgments: The authors thank Peter Smereka for suggesting the extension of our work to surface diffusion flow, which is presented in Section 8.2.

References

[1] E. BÄNSCH, P. MORIN, AND R. NOCHETTO, *Finite element methods for surface diffusion*, International meeting on free boundary problems, theory and applications, Trento, (2002).

[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 (1995), pp. 484–500.

[3] M. BERTALMIO, G. SAPIRO, V. CASELLES, AND C. BALLESTER, *Image inpainting*, in Siggraph 2000, Computer Graphics Proceedings, K. Akeley, ed., ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 2000, pp. 417–424.

[4] M. BURGER, *Numerical simulation of anisotropic surface diffusion with curvature-dependent energy*, J. Comput. Phys., 203 (2005), pp. 602–625.

[5] T. F. CHAN, S. H. KANG, AND J. SHEN, *Euler's elastica and curvature based inpaintings*, J. Appl. Math., 63 (2002), pp. 564–592.

[6] K. DECKELNICK AND G. DZIUK, *Error analysis of a finite element method for the Willmore flow of graphs*, Interfaces and Free Boundaries, 8 (2006), pp. 21–46.

[7] K. DECKELNICK, G. DZIUK, AND C. ELLIOTT, *Computation of geometric partial differential equations and mean curvature flow*, Acta Numerica, 14 (2005), pp. 139–232.

[8] E. DOGAN, P. MORIN, R. NOCHETTO, AND M. VERANI, *Finite element methods for shape optimization and applications*, Preprint, (2006).

[9] M. DROSKE AND M. RUMPF, *A level set formulation for Willmore flow*, Interfaces Free Bound., 6 (2004), pp. 361–378.

[10] S. ESEDOḠLU AND R. MARCH, *Segmentation with depth but without detecting junctions*, Journal of Mathematical Imaging and Vision, 18 (2003), pp. 7–15.

[11] S. ESEDOĞLU, S. RUUTH, AND R. TSAI, *Threshold dynamics for shape reconstruction and disocclusion*, in Proc. ICIP05, International Conference on Image Processing, Genova, Italy, September 2005, pp. II–502 – II–505.

[12] S. ESEDOḠLU AND J. SHEN, *Digital inpainting based on the Mumford-Shah-Euler image model*, European J. Appl. Math., 13 (2002), pp. 353–370.

[13] S. ESEDOḠLU AND Y. TSAI, *Threshold dynamics for the piecewise constant Mumford-Shah functional*, J. Comput. Phys., 211 (2006), pp. 367–384.

[14] L. EVANS, *Convergence of an algorithm for mean curvature motion*, Indiana University Mathematics Journal, 42 (1993), pp. 553–557.

[15] R. GRZIBOVSKIS AND A. HEINTZ, *A convolution-thresholding scheme for the Willmore flow*, preprint 34 (revised version), Chalmers Univ. of Tech, Goteborg, Sweden, 2003.

[16] H. ISHII, G. PIRES, AND P. SOUGANIDIS, *Threshold dynamics type schemes for propagating fronts*, TMU Mathematics Preprint Series, 4 (1996).

[17] B. JAWERTH AND P. LIN, *Shape recovery by diffusion generated motion*, J. of Visual Communication and Image Representation, 13 (2002), pp. 94–102.

[18] M. KASS, A. WITKIN, AND D. TERZOPOULOS, *Snakes: Active contour models*, International Journal of Computer Vision, 1 (1987), pp. 321–331.

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

[20] S. MASNOU, *Disocclusion: a variational approach using level lines*, IEEE Trans. Image Processing, 11 (2002), pp. 68–76.

[21] S. MASNOU AND J.-M. MOREL, *Level lines based disocclusion*, in 5th IEEE International Conference on Image Processing, 1998.

[22] B. MERRIMAN, J. BENCE, AND S. OSHER, *Diffusion generated motion by mean curvature*, in Computational Crystal Growers Workshop, J. Taylor, ed., American Mathematical Society, Providence, Rhode Island, 1992, pp. 73–83. Also available as UCLA CAM Report 92-18, April 1992.

[23] ——, *Motion of multiple junctions: a level set approach*, J. Comput. Phys., 112 (1994), pp. 334–363.

[24] M. NITZBERG, D. MUMFORD, AND T. SHIOTA, *Filtering, segmentation, and depth*, in Lecture Notes in Computer Science, vol. 662, Springer-Verlag, 1993.

[25] S. J. RUUTH, *A diffusion-generated approach to multiphase motion*, J. Comput. Phys., 145 (1998), pp. 166–192.

[26] S. J. RUUTH, *Efficient algorithms for diffusion-generated motion by mean curvature*, J. Comput. Phys., 144 (1998), pp. 603–625.

[27] S. J. RUUTH AND B. MERRIMAN, *Convolution-generated motion and generalized Huygens' principles for interface motion*, SIAM Journal on Applied Mathematics, 60 (2000), pp. 868–890.

[28] S. J. RUUTH, B. MERRIMAN, AND S. OSHER, *Convolution generated motion as a link between cellular automata and continuum pattern dynamics*, J. Comput. Phys., 151 (1999), pp. 836–861.

[29] S. J. RUUTH AND B. WETTON, *A simple scheme for volume-preserving motion by mean curvature*, J. Scientific Computation, 19 (2003), pp. 373–384.

[30] P. SMEREKA, *Semi-implicit level set methods for curvature flow and for motion by surface diffusion*, J. Sci. Comput., 19 (2003), pp. 439–456.

[31] T. J. WILLMORE, *Riemannian Geometry*, Clarendon Press, 1993.