ALGORITHMS FOR THE EVOLUTION OF SURFACES

MATTHEW STONE

Abstract. We develop algorithms for simulating various curvature dependent motions of interfaces. These motions are described by a class of partial differential equations for which a notion of weak solution (the "viscosity" solution) has been developed that is unique and persists beyond topological changes that may occur during the evolution. A comparison principle that holds for these evolutions is the essential tool in the theory. It is also known that to establish the convergence (to the viscosity solution) of a consistent approximation scheme for such an evolution, it is sufficient to show that it respects the comparison principle (in which case it is called "monotone").

A familiar example that comes up in many applications is motion by mean curvature. We try to extend recent consistent and monotone numerical schemes for motion by mean curvature to evolution laws that have a more general dependence on the principal curvatures.

1. Introduction

In [1], Esedoglu, Ruuth, and Tsai (ERT) developed an algorithm for approximating the motion by certain functions of mean curvature of an interface. They adapt a similar method used by Merriman, Bence, and Osher (MBO) in [2], which alternated two steps: Convolution and simple thresholding. The MBO algorithm proceeded as follows:

Let $\Sigma \subset \mathbb{R}^N$ be a region whose border, denoted $\partial\Sigma$, is to be evolved via motion by mean curvature. For any time step $\delta t > 0$, the algorithm gives approximations, $\{\partial\Sigma_n\}$ to motion by mean curvature, where $\partial\Sigma_n$ is an approximation of the curve at time $t = n \cdot \delta t$. These approximations are generated inductively by the following, with $\Sigma_0 = \Sigma$:

(1) *Convolution*: Form $u : \mathbb{R}^N \to \mathbb{R}$ as
$$u(\mathbf{x}) = (G_t * \mathbf{1}_{\Sigma_n})(\mathbf{x})$$
where $G_t(\mathbf{x})$ is the N-dimensional Gaussian kernel
$$G_t(\mathbf{x}) = \frac{1}{(4\pi t)^{N/2}} e^{-\frac{|\mathbf{x}|^2}{4t}}$$

(2) *Thresholding*: Compute the next approximation with
$$\Sigma_{n+1} = \left\{ \mathbf{x} : u(\mathbf{x}) \geq \frac{1}{2} \right\}$$

This algorithm, while computationally inexpensive, has its flaws. Most prominently, unless grid size is refined along with the time step, the approximate motion generated by the algorithm will get "stuck" [1][2].

Thus Esedoglu, Ruuth, and Tsai replace the thresholding step with another efficient procedure: constructing the signed distance function to the interface, and using formulas for approximations of the mean curvature in terms of convolutions with the Gaussian. Due to the Lipschitz continuity of signed distance functions, this eliminated the inaccuracies found in the MBO algorithm.

This paper explores similar methods to the ERT algorithm, investigating functions not only of mean curvature, but of its components, the principal curvatures. Due to certain problems encountered with maintaining the monotonicity of the algorithm, our final goal is to approximate motion by $f((\kappa_1)_+, (\kappa_2)_-)$, where f is any Lipschitz function that is increasing in each coordinate, $\kappa_1 > \kappa_2$ are the principal curvatures. The function $(x)_+$ is defined as 0 for $x \leq 0$ and x otherwise. Similarly, $(x)_-$ is defined as 0 for $x \geq 0$ and x otherwise.

1.1. **Things that need to be done.** Prove theorems.

2. Expansions for the distance function

In this section, we first write down the Taylor expansion of the signed distance function $d_\Sigma(\mathbf{x})$ along a unital direction $\vec{v}$ in the neighborhood of a point $p \in \partial\Sigma$ on the smooth boundary $\partial\Sigma$ of a set Σ. We work in $\mathbb{R}^3$ where we write $\mathbf{x} = (x, y, z)$. This expansion allows us to obtain a Taylor expansion for the convolution of $d_\Sigma(x, y, z)$ with the Gaussian kernel $G_{t,\vec{v}}(x, y, z)$, which is defined as:

$$G_{t,\vec{v}}(\mathbf{x}) = \frac{1}{(4\pi t)^{N/2}} e^{-\frac{(\mathbf{x}\cdot\vec{v})^2}{4t}}.$$

This will allow us to express the expansion coefficients in terms of the second fundamental form, and thus, using the convolution for every direction $\vec{v}$, will allow us to express the principle curvatures of $\partial\Sigma$.

2.1. **Expansion for a smooth interface.** First let us recall a few well known properties of the signed distance function. The following will be taken directly from [1].

For the following, d_s will denote $\frac{\partial d_\Sigma}{\partial s}$, d_{xs} will denote $\frac{\partial^2 d_\Sigma}{\partial x \partial s}$, and so on.
The taylor expansion at $s = 0$ is given by:

(2.1)

$$\begin{aligned} d(x,y,z,s) = d((x,y,z)+s\vec{v}) = d(x,y,z) + d_s(x,y,z,0)s + \frac{1}{2}d_{ss}(x,y,z,0)s^2 + d_{xs}(x,y,z,0)xs \\ + d_{ys}(x,y,z,0)ys + d_{zs}(x,y,z,0)zs + \frac{1}{6}d_{sss}(x,y,z,0)s^3 + \frac{1}{2}d_{ssx}(x,y,z,0)s^2x \\ + \frac{1}{2}d_{ssy}(x,y,z,0)s^2y + \frac{1}{2}d_{ssz}(x,y,z,0)s^2z + d_{sxy}(x,y,z,0)sxy + d_{sxz}(x,y,z,0)sxz \\ + d_{syz}(x,y,z,0)syz + \text{higher order terms} \end{aligned}$$

We can substitute the expansion into the convolution integral

$$\int_{\mathbb{R}} G_{t,\vec{v}}(x,y,z,\sigma)d(x,y,z,s-\sigma)d\sigma$$

to get a Taylor expansion for the convolution $(G_{t,\vec{v}} * d)(x,y,z,s)$ at $(x,y,z,s) = (0,0,0,0)$. The terms we need are:

$$(c * G_{t,\vec{v}})(x,y,z,0) = c \quad (s * G_{t,\vec{v}})(x,y,z,0) = 0$$

$$(s^2 * G_{t,\vec{v}})(x,y,z,0) = 2t \quad (s^3 * G_{t,\vec{v}})(x,y,z,0) = 0$$

Using these, we arrive at the following expansion:

Proposition 2.1. *Convolution of the signed distance function d with the Gaussian kernel $G_{t,\vec{v}}$ has the following expansion*

(2.2)

$$(d * G_{t,\vec{v}})(x,y,z,0) = d(x,y,z) + \frac{1}{2}d_{ss}(x,y,z,0)\cdot 2t + \frac{1}{2}d_{ssx}(x,y,z,0)\cdot 2tx + \frac{1}{2}d_{ssy}(x,y,z,0)\cdot 2ty + \frac{1}{2}d_{ssz}(x,y,z,0)\cdot 2tz + \text{higher order terms}$$

Provided that each of x,y,z are $O(t)$, we get

$$(d * G_{t,\vec{v}})(x,y,z,0) = d(x,y,z) + d_{ss}(x,y,z,0)t + O(t^2) \tag{2.3}$$

Thus we find

$$G_{t,\vec{v}} * d - d = d_{ss}(x,y,z,0)t + O(t^2)$$

By its definition,

$$d_{ss}(x,y,z,0) = \left\langle (D^2 d)\vec{v}, \vec{v} \right\rangle.$$

Finally giving

$$(G_{t,\vec{v}} * d - d) = \left\langle (D^2 d)\vec{v}, \vec{v} \right\rangle t + O(t^2)$$

For any signed distance function d, the eigenvalues of $D^2 d$ are given by the principle curvatures with eigenvectors along the principle directions, plus an eigenvalue of 0 with corresponding eigenvector of the normal to the surface.
Thus $\max_{|\vec{v}|=1} \left\langle (D^2 d)\vec{v}, \vec{v} \right\rangle$, that is, the greatest eigenvalue, is either κ_1 or 0, depending which is larger. That is, $\max_{|\vec{v}|=1} \left\langle (D^2 d)\vec{v}, \vec{v} \right\rangle$.
Similarly, $\min_{|\vec{v}|=1} \left\langle (D^2 d)\vec{v}, \vec{v} \right\rangle = (\kappa_2)_-$.
This leads to the conclusions that:

$$\max_{|\vec{v}|=1}(G_{t,\vec{v}} * d - d) = (\kappa_1)_+ t + O(t^2) \tag{2.4}$$

$$\min_{|\vec{v}|=1}(G_{t,\vec{v}} * d - d) = (\kappa_2)_- t + O(t^2) \tag{2.5}$$

Using these two equations, for any Lipschitz function f,

(2.6)

$$f\left(\frac{1}{t}\left(\max_{|\vec{v}|=1}(G_{t,\vec{v}} * d - d)\right), \frac{1}{t}\left(\min_{|\vec{v}|=1}(G_{t,\vec{v}} * d - d)\right)\right) = f((\kappa_1)_+, (\kappa_2)_-) + O(t)$$

This leads to the following algorithm:

Algorithm 2.2. *Given the initial set Σ_0 through its signed distance function $d_0(\mathbf{x})$ and a time step $\delta t > 0$, generate the sets Σ_j via their signed distance functions $d_j(\mathbf{x})$ at subsequent discrete times $t = j(\delta t)$ by alternating the following steps:*

(1) *Form the function*

$$A(\mathbf{x}) := d_j + (\delta t) f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d - d)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d - d)\right)\right) \tag{2.7}$$

(2) *Construct distance function d_{j+1} by*

$$d_{j+1}(\mathbf{x}) = \mathbf{Redist}(A). \tag{2.8}$$

At the jth step of the algorithm, the set Σ_j can be recovered through the relation

$$\Sigma_j = \{\mathbf{x} : d_j(\mathbf{x}) > 0\}$$

So using Equation 2.6 gives the 0-level set of which has moved with the desired speed, proving the consistency of the algorithm.

Finally we must show the monotonicity of our algorithm, assuming f is Lipschitz and increasing:

Proposition 2.3. *If $M \geq L_f$, then algorithm 2.7 and 2.8 is monotone for any choice of time step size $\delta t > 0$.*

Proof. Let Σ_1, Σ_2 be two sets satisfying $\Sigma_1 \subset \Sigma_2$, and $d_1(\mathbf{x}), d_2(\mathbf{x})$ be signed distance functions to Σ_1, Σ_2 respectively. Firstly, we have

$$\forall \mathbf{x}, d_1(\mathbf{x}) \leq d_2(\mathbf{x})$$

Using the same notation as in the algorithm, let

$$A_1(\mathbf{x}) = d_1(\mathbf{x}) + (\delta t) f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_1 - d_1)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_1 - d_1)\right)\right)$$

$$A_2(\mathbf{x}) = d_2(\mathbf{x}) + (\delta t) f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2)\right)\right)$$

Then we calculate:

$$\begin{aligned} A_2 - A_1 = (d_2 - d_1) + (\delta t)\Bigg[& f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2)\right)\right) \\ & - f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_1 - d_1)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_1 - d_1)\right)\right)\Bigg] \end{aligned} \tag{2.9}$$

(2.10)

$$d_2 \geq d_1 \Rightarrow \forall \vec{v}, G_{M(\delta t),\vec{v}} * d_2 \geq G_{M(\delta t),\vec{v}} * d_1 \Rightarrow \max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2) \geq \max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_1 - d_1) \text{ and } \min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2) \geq \min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_1 - d_1)$$

Substituting 2.10 into 2.9, using the fact that f is increasing in both variables:

(2.11)

$$A_2 - A_1 \geq (d_2 - d_1) + (\delta t)\Bigg[f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_2)\right)\right) - f\left(\frac{1}{M(\delta t)}\left(\max_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_1)\right), \frac{1}{M(\delta t)}\left(\min_{|\vec{v}|=1}(G_{M(\delta t),\vec{v}} * d_2 - d_1)\right)\right)\Bigg]$$

Using the fact that f is Lipschitz with constant L_f and $M \geq L_f$:

$$A_2 - A_1 \geq (d_2 - d_1) - (\delta t)\frac{1}{M(\delta t)}\sqrt{2}L_f(d_2 - d_1) \geq 0 \tag{2.12}$$

□

Thus we have that our algorithm is both consistent and monotone, sufficient conditions to show the convergence of the algorithm to the desired motion as $\delta t \to 0$

References

[1] S. Esedoglu, S. Ruuth, and R. Tsai, *Diffusion generated motion using signed distance functions*, Journal of Computational Physics 229 (2010) 1017-1042.

[2] B. Merriman, J.K. Bence, and S. Osher, *Motion of multiple junctions: a level set approach*, Journal of Computational Physics 112 (2) (1994) 334-363.

Department of Mathematics, Yale University, New Haven, CT 57386

E-mail address: matthew.i.stone@yale.edu