On an approximation method for hyperbolic mean curvature flow

Elliott Ginder*
Research Institute for Electronic Science
Hokkaido University, Japan

Ayumu Katayama
Graduate School of Science
Hokkaido University, Japan

Karel Svadlenka
Department of Mathematics
Kyoto University, Japan

Models in modern engineering often include elements that pose challenges to numerical methods which should solve them. Difficult aspects can include, for example, singularities, free boundaries or nonlinear constraints. In this article, we present an approximation scheme for treating multiphase oscillatory interfacial motions. We also discuss the algorithm used for encoding and tracking the evolution of multiphase geometries.

1 Introduction

A frequently used model equation in applications is the *mean curvature flow*. This geometric evolution states that interfaces move in the direction of their normal with velocity v, which is proportional to their mean curvature κ:

$$v=\sigma\kappa.$$

Here, σ usually denotes the surface tension of the inteface.

This model has a variational structure, since for a smooth closed curve $\gamma:[a,b]\rightarrow \mathbb{R}^{2}$ it corresponds to the L^{2}-gradient flow of the interfacial surface energy:

$$E(\gamma)=\int_{a}^{b}\sigma|\gamma'(s)|\,ds.$$

A wide range of numerical methods for the computation of mean curvature flow and other interfacial motions are available. They can mainly be divided in two groups: methods explicitly tracking the interface (front-tracking) and methods dealing with the interface implicitly by expressing it as a level set of an auxiliary function. Although front-tracking methods are effective in various simulations [16] and are usually more straightforward than level-set methods, they are generally not able to deal with singularities and topological changes. Relatedly, these computational difficulties can correspond to a natural feature of the phenomena under investigation.

Recently, models including oscillatory versions of interface motions have been introduced and have gained much attention. One of the main research topics here is the *hyperbolic mean curvature flow* (HMCF, see [11]):

$$a=(1-v^{2})\kappa,$$

where a denotes the normal acceleration, v is the normal velocity, and κ is the mean curvature vector of the interface. This geometric evolution equation arises in relation to the motion of relativistic strings, where the speed of light is normalized to unity (see [3]). Considering the case where the velocity of the interface is small, relative to the speed of light, it is also interesting to investigate curvature dependent acceleration. In particular, the geometric evolutionary equation that we will consider is the case where the normal acceleration of the interface is proportional to its mean curvature:

$$a = \kappa. \tag{1}$$

Here we remark that the interface is also accompanied by a smooth initial velocity field (acting normal to the interface).

The outline of this manuscript is as follows. We begin by introducing our approximation method for (1), the HMBO. Then we formally describe an algorithm for detecting and encoding the precise location of multiphase geometries. We then present numerical results which utilize our methods, including an examination into the behavior of a multiphase volume preserving HMCF

2 The HMBO algorithm

Our approximation method for (1) is threshold dynamical and is formulated by using the solution to single vector-valued wave equation. In particular, choosing a small time step Δt, we find a function $\boldsymbol{u} : \Omega \to \mathbf{R}^{N-1}$ solving:

$$\begin{cases} \boldsymbol{u}_{tt} = c^2 \Delta \boldsymbol{u} & \text{in } (0, \Delta t) \times \Omega, \\ \frac{\partial \boldsymbol{u}}{\partial \nu} = 0 & \text{on } (0, \Delta t) \times \partial\Omega, \\ \boldsymbol{u}_t(0, x) = \boldsymbol{v}_0 & \text{in } \Omega, \\ \boldsymbol{u}(t = 0, x) = 2\boldsymbol{z}_\epsilon^0 - \boldsymbol{z}_\epsilon^{-\Delta t} & \text{in } \Omega, \end{cases} \tag{2}$$

where N denotes the number of phases, Ω is a smooth bounded domain in $\mathbf{R}^d$, $\boldsymbol{v}_0$ is an appropriate initial velocity, c^2 is a wave speed depending on the dimension d (see the remark at the end of this section). The initial condition is defined by the following signed-distance interpolated vector field:

$$\boldsymbol{z}_\epsilon^t(x) = \sum_{i=1}^{N} \boldsymbol{p}_i \chi_{\{d_i^t(x) > \epsilon/2\}} + \frac{1}{\epsilon}\left(\frac{\epsilon}{2} + d_i^t(x)\right) \boldsymbol{p}_i \chi_{\{-\epsilon/2 \le d_i^t(x) \le \epsilon/2\}}, \tag{3}$$

where $\boldsymbol{z}_\epsilon^{-\Delta t}(x)$ is constructed using the initial velocity along the interface. Here, $\epsilon > 0$ is an interpolation parameter and $d_i^t(x)$ denotes the signed distance function to the boundary of phase i at location x and time t, ∂P_i^t:

$$d_i^t(x) = \begin{cases} \inf_{y \in \partial P_k^t} \|x - y\| & \text{if } x \in P_k^t, \\ -\inf_{y \in \partial P_k^t} \|x - y\| & \text{otherwise.} \end{cases} \tag{4}$$

In the above, χ_E denotes the characteristic function of the set E and $\boldsymbol{p}_i$ is the i^{th} coordinate vector of a *regular simplex* in $\mathbf{R}^{N-1}$, $i = 1, ..., N$. We remark that, when $N = 2$, equation (2) is scalar.

At time Δt, in a process called *thresholding*, each phase region is evolved as follows:

$$P_i^{\Delta t} = \{x \in \Omega : \boldsymbol{u}(\Delta t, x) \cdot \boldsymbol{p}_i \ge \boldsymbol{u}(\Delta t, x) \cdot \boldsymbol{p}_k, \text{for all } k \in \{1, ..., N\}\}. \tag{5}$$

The vector field $\boldsymbol{z}_\epsilon^0$ is then reconstructed using the boundaries of these sets and the initial condition for the wave equation is updated. The procedure is then repeated and one can show that if $\boldsymbol{v}_0 = \boldsymbol{0}$ then the geometric evolution of the interface approximates (1) in the cases $d = 2$ and $d = 3$ (provided that one takes $c^2 = 2$ when $d = 2$, and $c = 1$ when $d = 3$).

3 Detection of multiphase geometries

In the numerical implementation of our methods, the domain is first triangulated and numerical solutions are obtained by means of finite element methods. In our computations, the $P1$ finite element assumption is utilized and, using the process described below, this allows one to determine the precise geometry of interfaces within elements. We also remark that, since the target geometric evolution equation (1) is hyperbolic, care must be taken when tracking the interface and constructing (3).

Encoding the geometry and tracking the evolution of multiphase regions can be accomplished by the following procedure. Since the details related to its actual numerical implementation are rather technical, our explanation is formal. The algorithm is as follows:

Input.

N: number of phases.

e: a tetrahedral element with vertices $\boldsymbol{x}_1, \boldsymbol{x}_2, \boldsymbol{x}_3, \boldsymbol{x}_4$ and edges $\ell_1^2, \ell_1^3, \ell_1^4, \ell_2^3, \ell_2^4, \ell_3^4$.

$\hat{\boldsymbol{u}}$: a smooth vector field taking values in $\mathbf{R}^N$, defined on e.

Output.

The multiphase geometry within e.

1. Construct a regular simplex in R^N with vertex coordinates $\boldsymbol{p}_1, \boldsymbol{p}_2, ..., \boldsymbol{p}_N$.

2. Construct the $P1$-Lagrange approximation to $\hat{\boldsymbol{u}}$:

$$\boldsymbol{u}(x,y,z) = \boldsymbol{\alpha} x + \boldsymbol{\beta} y + \boldsymbol{\gamma} x + \boldsymbol{\delta}$$

$$\begin{pmatrix} x_1 & y_1 & z_1 & 1 \\ x_2 & y_2 & z_2 & 1 \\ x_3 & y_3 & z_3 & 1 \\ x_4 & y_4 & z_4 & 1 \end{pmatrix} \begin{pmatrix} \alpha_i \\ \beta_i \\ \gamma_i \\ \delta_i \end{pmatrix} = \begin{pmatrix} \hat{u}_{1,i} \\ \hat{u}_{2,i} \\ \hat{u}_{3,i} \\ \hat{u}_{4,i} \end{pmatrix}$$

where $\hat{u}_{k,i}$ denotes the i^{th} component of $\hat{\boldsymbol{u}}$ and location $\boldsymbol{x}_i$, $(i = 1, 2, ..., N)$.

3. For all combinations of i and j (not counting order repetition), construct the set:

$$T = \bigcup T_{ij},$$

where each number in the union is a plane defined by

$$T_{ij} = \{\boldsymbol{x} \in e \mid \langle \boldsymbol{u}(\boldsymbol{x}), \boldsymbol{p}_i - \boldsymbol{p}_j \rangle = 0\}.$$

The collection of planes within T contains all candidate locations for phase changes, which completely describe the interfaces.

4. For each edge ℓ_m^n of the element e, detect the location of intersection (or lack thereof) with each T_{ij} and accumulate them into a set

$$\mathcal{C} = \bigcup_{m,n,i,j} I_{mn}^{ij}$$

where each member of the union is defined:

$$I_{mn}^{ij} = \{\boldsymbol{x} \in e \mid \boldsymbol{x} \in \{\ell_m^n \cap T_{ij}\}\}. \tag{6}$$

Note. The intersection may be empty, consist of a single point, or consist of an infinite number of points when ℓ_m^n lies in the plane described by T_{ij} (in such a case, take the endpoints of ℓ_m^n as $\boldsymbol{x}$.)

5. For each pair of elements in T, find their lines of intersection ℓ_{ij}^{kl}, and collect them in a set:

$$\mathcal{M} = \bigcup_{i,j,k,l} \ell_{ij}^{kl}.$$

Notes.

- When the planes are parallel and do not coincide, there is no intersection.
- When the planes are coincident, $T_{ij} = T_{kl}$.
- Otherwise, the intersection is a line in $\mathbf{R}^3$.

6. Determine the location of intersection of the lines in $\mathcal{M}$ and accumulate them into a set

$$\mathcal{P} = \bigcup_{a,b \in \mathcal{M}} v_a^b, \tag{7}$$

where

$$v_a^b = \{\boldsymbol{x} \in e \mid \boldsymbol{x} \in a \cap b \quad a, b \in \mathcal{M}\}.$$

Note. Lines in $\mathbf{R}^3$ almost never intersect, and so the intersections here need to be checked using appropriate floating point error measurements.

7. Form the union of $\mathcal{C}$ and $\mathcal{P}$, together with the set of element vertices and their corresponding phases into a set $\hat{\mathcal{P}}$.

8. Remove all points in $\hat{\mathcal{P}}$ that are outside the element (again call the set $\hat{\mathcal{P}}$).

9. Partition and filter $\hat{\mathcal{P}}$ into N subsets (some of which may be empty):

$$P_i = \{\boldsymbol{x} \in \mathbf{R}^3 \mid \langle \boldsymbol{u}(\boldsymbol{x}), \boldsymbol{p}_i \rangle \geq \langle \boldsymbol{u}(\boldsymbol{x}), \boldsymbol{p}_j \rangle \text{ for all } j\}. \tag{8}$$

The points in P_i (except possibly those corresponding to vertices of the element) correspond to locations on the boundary of phase i.

10. The points in each P_i define a convex polytope, so one can construct their convex hull to obtain the precise geometry of each phase.

Note. When displaying the geometry of the interfaces, element vertices should only be used when a phase change occurs at the location of the vertex.

4 Application to simulation of interfacial motions

Using the numerical counterpart of the algorithm for detecting multiphase geometries described above, we are able to approximate interfacial motions in two and three dimensions. We will examine multiphase curvature flow and HMCF in $\mathbf{R}^3$, and simulate a multiphase volume preserving HMCF in $\mathbf{R}^2$.

4.1 Curvature flow

Using a Delaunay triangulation, a uniform grid with node spacing 1/20 was used to partition the unit cube into a finite number of tetrahedra. The initial condition corresponds to the configuration of the three phases shown in the first image of Figure 1. The numerical results were obtained by means of the MBO algorithm [12].

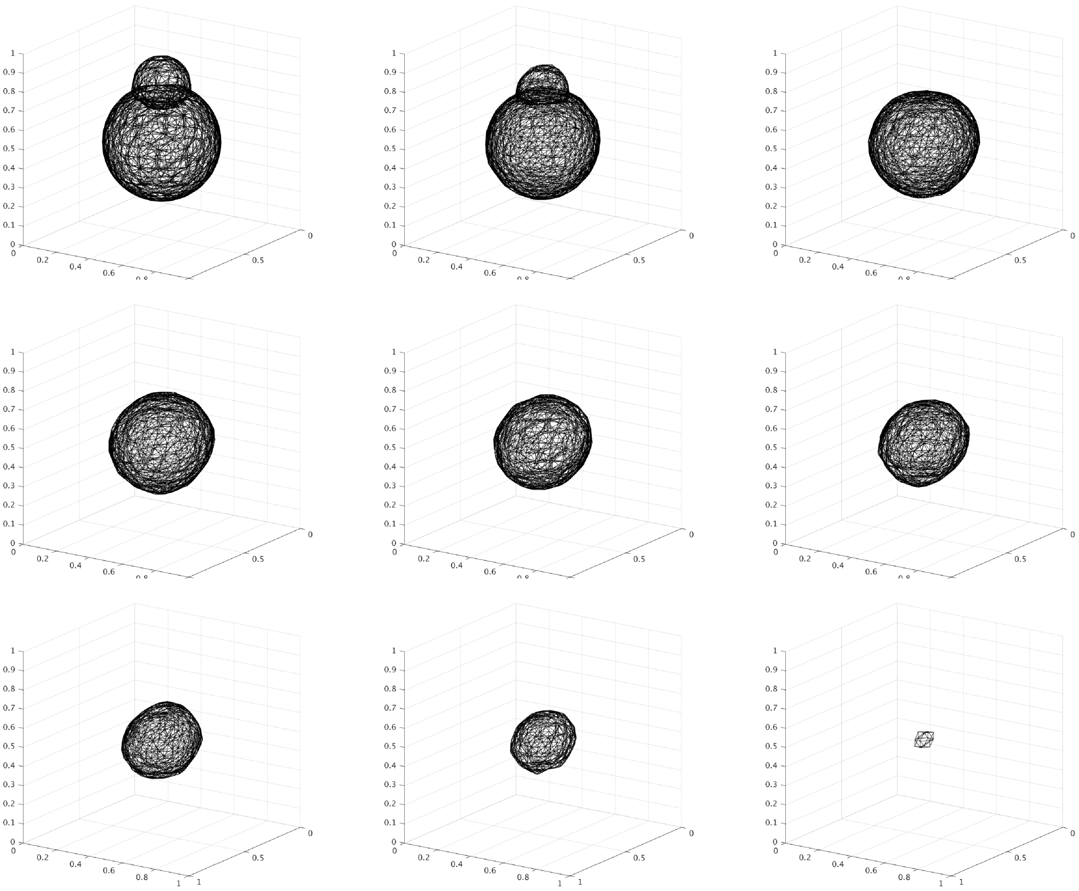

Figure 1: Evolution of a three phase mean curvature flow. Time is from top to bottom, left to right.

4.2 Hyperbolic mean curvature flow

Numerical results corresponding to a two phase HMCF are shown in figure 2. The interfacial motions were simulated using the HMBO algorithm with the initial condition shown in the first image of the figure. The initial velocity of the interface was taken as zero, and we utilize the same triangulation as in the previous curvature flow simulation.

4.3 Minimizing movements and volume preserving motions

In this section, we will explain the basic idea behind minimizing movements and exemplify its application to the simulation of constrained oscillatory interfacial motions.

For a given Lagrangian L and boundary conditions, consider the problem of constructing stationary points of the action integral

$$\int_0^T \left\{ \frac{1}{2} \int_\Omega u_t^2 \, dx - \mathcal{E}(u) \right\} dt, \tag{9}$$

where

$$\mathcal{E}(u) = \int_\Omega L(\nabla u(x), u(x), x) \, dx.$$

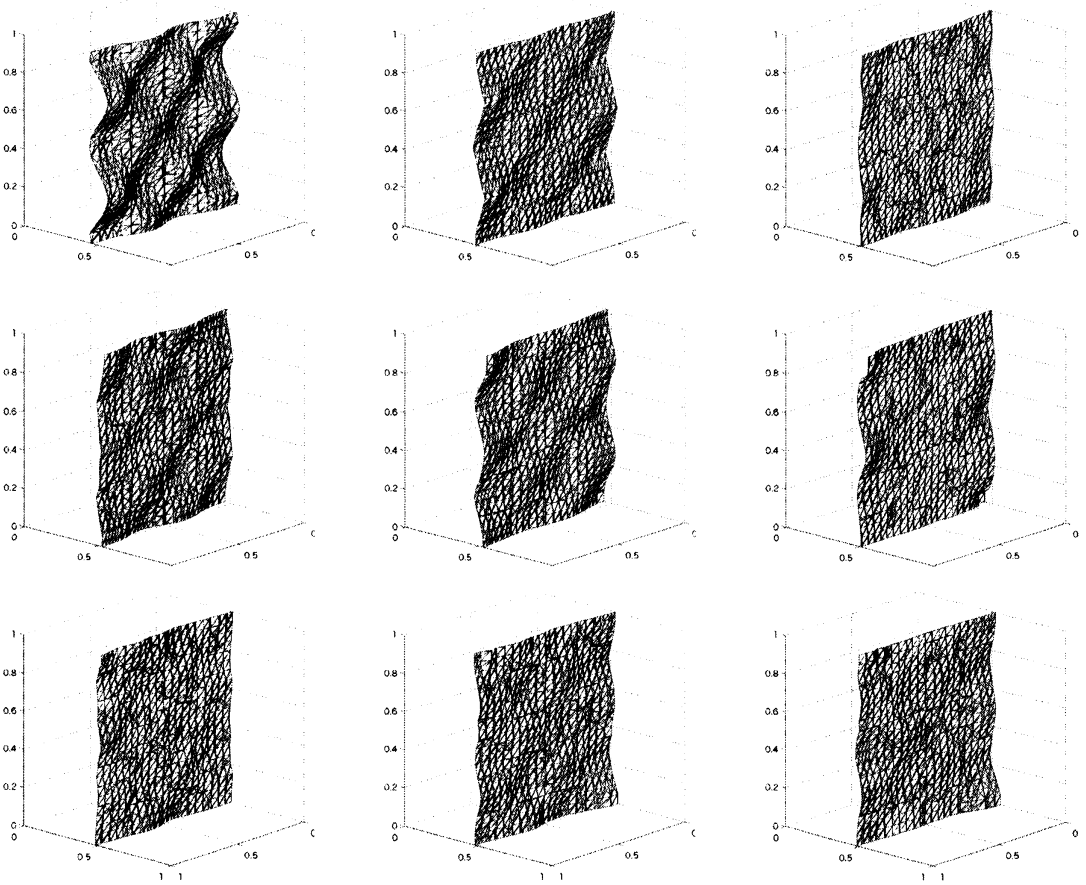

Figure 2: Evolution of a two phase hyperbolic mean curvature flow. Time is from top to bottom, left to right.

The method of minimizing movements can be used to produce a sequence of functions $\{u_n\}$ which approximates stationary points of (9) by recursively minimizing functionals of the form:

$$\mathcal{F}_n(u) = \int_\Omega \frac{|u - 2u_{n-1} + u_{n-2}|^2}{2h^2} dx + \mathcal{E}(u),$$

in a suitable function space. Here u_{n-1} and u_{n-2} are appropriately given functions (constructed from initial conditions) and $h > 0$ is the time step.

The Euler-Lagrange equation of each functional $\mathcal{F}_n$ expresses a local approximation of the stationary point:

$$u = 2u_{n-1} - u_{n-2} - h^2 \frac{\delta \mathcal{E}(u)}{\delta u},$$

where $\frac{\delta \mathcal{E}(u)}{\delta u}$ denotes the functional derivative. For example, when the Lagrangian is taken as $L = |\nabla u|^2 /2$, we obtain a functional whose Euler-Lagrange equation is a time-discretization of the wave equation. This allows one to treat "equation of motion" problems, which are often of the hyperbolic type. We remark that the mathematical properties of the parabolic and hyperbolic minimizing movements have been investigated in detail (see e.g., [1, 18]).

In combination with minimizing movements, the algorithm in section (3) also enables one to investigate volume constrained motions. Solutions to the infinite dimensional minimization problems are approximated by solutions to corresponding finite dimensional minimizations. We remark that computation of functional minimizers can be achieved in a number of ways, for example by nonlinear conjugate gradient methods, or even by steepest descent.

In particular, we use hyperbolic minimizing movements to approximate solutions to the wave equation (2) as a sequence of minimization problems. By adding a penalty term for the volume preservations, this approach allows us to investigate multiphase volume preserving motions. Figure 3 shows a numerical result obtained though utilizing minimizing movements corresponding to functionals with the form:

$$\mathcal{F}_n(\boldsymbol{u}) = \int_\Omega \frac{|\boldsymbol{u} - 2\boldsymbol{u}_{n-1} + \boldsymbol{u}_{n-2}|^2}{2h^2} dx + \mathcal{E}(\boldsymbol{u}) + \frac{1}{\tilde{\epsilon}} \sum_{k=1}^{N-1} (vol(P_k) - V_k)^2,$$

where V_k denotes the prescribed volume of phase k and P_k is the region corresponding to phase k within $\boldsymbol{u}$. The initial condition is shown in bold, and the initial velocities were zero. We observe the interfaces oscillate, while individual phase volumes are approximately preserved.

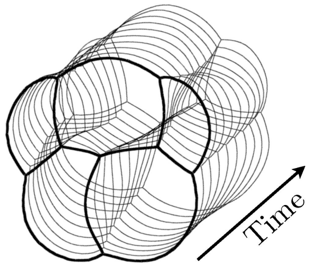

Figure 3: Multiphase volume preserving hyperbolic mean curvature flow.

5 Conclusion

The HBMO algorithm was presented and we described a formal method for detecting and encoding multiphase geometries. Our approximation method allows one to naturally deal with topological changes, junctions and nonlocal constraints. Using our methods, we then simulated motions of hypersurfaces embedded in $\mathbf{R}^3$ and, by detecting the precise location of interfaces, we were able to compute the volume of individual phase regions. This technique allowed us to simulate multiphase interfacial motion by a volume preserving hyperbolic mean curvature flow. For such motions, the hyperbolic setting is considered a highly challenging topic in mathematics. We expect that our approximation scheme can provide a system for further understanding such motions and this is a topic that we aim to pursue.

References

[1] L. Ambriosio, N. Gigli, G. Savaré, *Gradient flows in metric spaces and in the space of probability measures*, Birkhäuser, 2005.

[2] G. Barles, C. Georgelin, *A simple proof of convergence of an approximation scheme for computing motions by mean curvature*, SIAM J. Numer. Anal. **32**:2, pp. 484–500, 1995.

[3] M. Bonafini, *A BMO-type Scheme for the Relativistic Hyperbolic Mean Curvature Flow*, Master's Thesis, Università degli Studi di Verona, 2015.

[4] E. De Giorgi, *Movimenti minimizzanti*, talk given at the meeting "Aspetti e problemi della matematica oggi", Lecce, October 20-22, 1992.

[5] S. Esedoglu, S. Ruuth, R. Tsai, *Diffusion generated motion using signed distance functions*, Journal of Computational Physics **229**:4, pp. 1017–1042, 2010.

[6] L.C. Evans, *Convergence of an algorithm for mean curvature motion*, Indiana U. Math. J. **42**, pp. 533–557, 1993.

[7] E. Ginder, K. Svadlenka, *On an algorithm for curvature-dependent interfacial acceleration*, JSCES **19**, 2014.

[8] E. Ginder, K. Svadlenka, *Wave-type threshold dynamics and the hyperbolic mean curvature flow*, preprint, 2015.

[9] Y. Goto and K. Ishii, T. Ogawa, *Method of the distance function to the Bence-Merriman-Osher algorithm for motion by mean curvature*, Comm. Pure Appl. Anal. **4**, pp. 311–339, 2005.

[10] N. Kikuchi, *An approach to the construction of Morse flows for variational functionals*, Nematics: Mathematical and Physical Aspects, pp. 195–199, 1991.

[11] P. G. LeFloch, K. Smoczyk, *The hyperbolic mean curvature flow*, J. Math. Pures Appl. **90**, pp. 591–614, 2008.

[12] B. Merriman, J. K. Bence, S. J. Osher, *Motion of multiple junctions: A level set approach*, J. Comp. Phys. **112**, pp. 334–363, 1994.

[13] R.Z. Mohammad, K. Svadlenka, *Multiphase volume-preserving interface motions via localized signed distance vector scheme*, Discrete and Continuous Dynamical Systems - Series S, 2014.

[14] R.Z. Mohammad, K. Svadlenka, *On a penalization method for an evolutionary free boundary problem with volume constraint*, Advances in Mathematical Sciences and Applications, **24**:1, pp. 85–101, 2014.

[15] S. Ruuth, B. Wetton, *A simple scheme for volume-preserving motion by mean curvature*, J. Sci. Comput. **19**, pp. 373–384, 2003.

[16] D. Ševčovič, S. Yazaki, *Evolution of plane curves with a curvature adjusted tangential velocity*, Japan J. Indust. Appl. Math., **28**, pp. 413-442.

[17] K. Svadlenka, E. Ginder, S. Omata, *A variational method for multiphase volume-preserving interface motions*, Journal of Computational and Applied Mathematics **257**, pp. 157–179, 2014.

[18] A. Tachikawa, *A variational approach to constructing weak solutions of semilinear hyperbolic systems*, Advances in Mathematical Sciences and Applications **4**, pp. 93–103, 1994.