UCLA

COMPUTATIONAL AND APPLIED MATHEMATICS

A Diffusion-Generated Approach to Multiphase Motion

Steven J. Ruuth

June 1997

CAM Report 97-28

Department of Mathematics
University of California, Los Angeles
Los Angeles, CA. 90095-1555

A diffusion-generated approach to multiphase motion

Steven J. Ruuth*

June 16, 1997

Abstract

In this article, we present a diffusion-generated approach for evolving multiple junctions. This work generalizes an earlier method by Merriman, Bence and Osher which alternately diffuses and sharpens characteristic functions for each phase region to produce pure mean curvature flow. Specifically, our new method produces a normal velocity equal to a positive multiple of the curvature of the interface plus the difference in bulk energies for prescribed junction angles. This simple method naturally treats topological mergings and breakings, produces no overlapping regions or vacuums and can be made very fast.

Numerical studies are provided which show that our method agrees with front tracking and a recent variational approach for a variety of examples. Asymptotic expansions are also carried out near junctions to justify our algorithms.

*Department of Mathematics, University of California at Los Angeles. (`ruuth@math.ucla.edu`).
The work of this author was partially supported by an NSERC Postdoctoral Scholarship and NSF DMS94-04942.

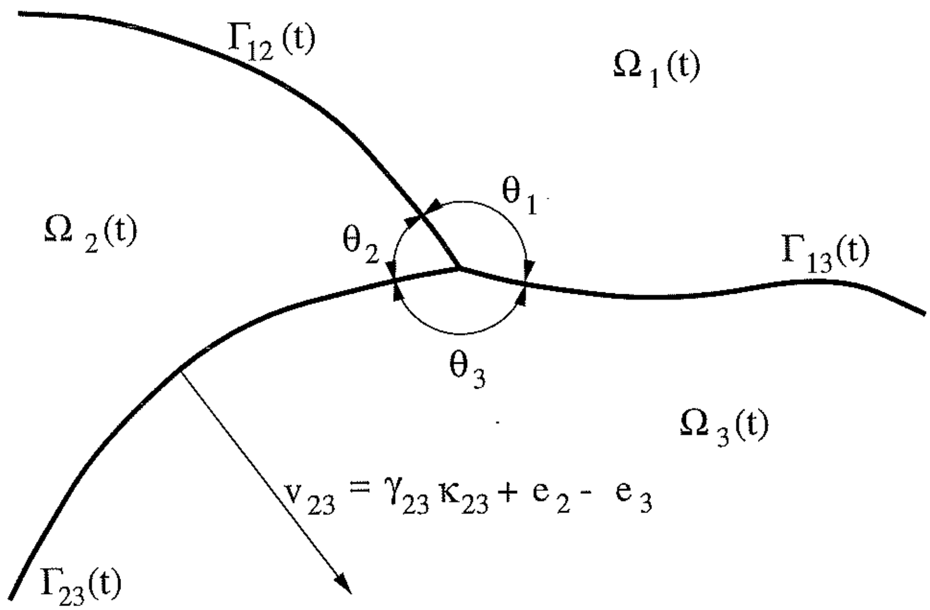


Figure 1: The interfaces, Γ_{ij}, move with a normal velocity $v_{ij} = \gamma_{ij}\kappa_{ij} + e_i - e_j$ and are subject to angles θ_1, θ_2, θ_3.

1 Introduction

In a variety of applications, one wants to follow the motion of a front that moves with some curvature-dependent speed. For the special case of pure mean curvature flow, junctions of moving surfaces have been treated by alternately diffusing and sharpening the characteristic functions for each phase region [8, 9]. In this work, we generalize this diffusion-generated approach to allow for a normal velocity equal to a positive multiple of the mean curvature, κ, of the interface plus the difference in bulk energies.

In two dimensions, the simplest model that we consider involves three curves meeting at a point with prescribed angles θ_1, θ_2 and θ_3. Each interface, Γ_{ij}, separates regions Ω_i and Ω_j and moves with a normal velocity,

$$v_{ij} = \gamma_{ij}\kappa_{ij} + e_i - e_j \tag{1}$$

as is shown in Figure 1.

To treat such motions, several methods have been developed. Front tracking methods (e.g., [2]), for example, are often well-suited for curves that never cross because they explicitly approximate the motion of the interface rather than a level set of some higher dimensional function. When line or planar segments interact, however, decisions must be made as to whether to insert or delete segments. Because complicated topological changes can occur for the model problem (1) implementation of front tracking methods is often impractical, especially for more than two dimensions.

Other approaches also have limitations. Monte-Carlo methods for Potts models (e.g., [6]) can introduce unwanted anisotropy into the motion due to the spatial mesh [15] and are typically too slow to find accurate approximations of the model. Phase field methods (e.g., [3]) may also be used, but these are often inherently too expensive for practical computation [9] because they represent the interface as an internal layer, and thus require an extremely fine mesh (at least locally) to resolve this layer.

To address these concerns for the case of pure mean curvature flow (i.e., $\gamma_{ij} = 1$ and $e_i = 0$), a method (MBO) based on the model of diffusion-dependent motion of level sets was proposed by Merriman, Bence and Osher [8, 9]. This method naturally handles complicated topological changes with junctions in several dimensions. Furthermore, this method can be made very efficient by discretizing in space using a Fourier spectral basis and using a quadrature to determine the Fourier coefficients at each step [11, 13]. Similarly to all other methods for multiple phase problems, no convergence results are known for the MBO-method. However, [5, 1] do give rigorous convergence proofs for two phase mean curvature motion and [7, 11] give some further asymptotic results.

To allow for the more general motion (1), a variational approach was recently proposed [16] which gives a practical method for treating junctions even when topological mergings and breakings occur. This approach is especially well-suited for treating problems with additional constraints. Unfortunately, it is unable to approximate many problems involving $r > 3$ phase regions since only r independent γ_{ij} may be prescribed. Furthermore, this method limits angles to the classical condition (see, e.g., [14])

$$\frac{\sin(\theta_1)}{\gamma_{23}} = \frac{\sin(\theta_2)}{\gamma_{13}} = \frac{\sin(\theta_3)}{\gamma_{12}} \tag{2}$$

at triple points and is relatively slow when compared to the MBO-method for the case of pure mean curvature flow.

In this paper, we develop algorithms for the multiphase model (1) for any number of phase regions which retain the speed and much of the simplicity of the MBO-method. Although the methods given throughout this paper are semi-discrete, we note that very efficient[1] implementations are possible using the algorithms described in [11, 13]. An outline of the paper follows.

In Section 2, we give the MBO-method for two phase and multiple phase problems.

Section 3 generalizes the MBO-method to nonsymmetric junctions by replacing the sharpening step with a new decision. Asymptotic and numerical justifications of our algorithm are also given.

In Section 4, we diffuse each characteristic function a number of times (once for each γ_{ij}) and combine the results with the nonsymmetric junction algorithm. This gives a method for evolving each branch with a normal speed $v = \gamma_{ij}\kappa$ for prescribed angle conditions. For the special (but important) case where the angles obey the classical condition (2), asymptotic and numerical justifications of our algorithms are given.

By changing the sharpening decision, Sections 5 and 6 extend these methods to models which involve bulk energies and any number of phase regions. Numerical justifications of our methods are given and an example of a four-phase problem which cannot be treated using the variational approach is also provided.

Chapter 7 concludes by summarizing our results and discussing some possible areas of future research.

2 The MBO-Method

An algorithm for following interfaces propagating with a normal velocity equal to mean curvature was introduced by Merriman, Bence and Osher [8, 9]. In this section, we describe the method for the two phase and multiple phase problems. Subsequent sections describe new algorithms which generalize these methods to the multiphase motions described by equation (1).

[1]Using a step size Δt, $\mathcal{O}(\frac{1}{\Delta t}\log^2(\Delta t))$ floating point operations are required per step of the algorithm [11, 13].

2.1 The Two Phase Problem

Suppose we wish to follow an interface moving with a normal velocity equal to its mean curvature. To evolve a surface according to this motion, we may use the MBO-method for two regions:

MBO-Method (Two Regions)
BEGIN
(1) Set U equal to the characteristic function for the initial region.
i.e., set $U(\vec{x},0) = \begin{cases} 1 & \text{if } \vec{x} \text{ belongs to the initial region} \\ 0 & \text{otherwise.} \end{cases}$

REPEAT for all steps, j, from 1 to the final step:
BEGIN
(2) Apply diffusion[2] to U for some time, Δt.
i.e., find $U(\vec{x}, j\Delta t)$ using $\begin{cases} U_t & = & \nabla^2 U, \\ \frac{\partial U}{\partial n} & = & 0 \text{ on } \partial\mathcal{D} \end{cases}$
starting from $U(\vec{x},(j-1)\Delta t)$.
(3) "Sharpen" the diffused region by setting
$U(\vec{x}, j\Delta t) = \begin{cases} 1 & \text{if } U(\vec{x}, j\Delta t) > \frac{1}{2} \\ 0 & \text{otherwise.} \end{cases}$
END
END

For any time t, the level set $\{\vec{x} : U(\vec{x},t) = \frac{1}{2}\}$ gives the location of the interface.

An extension to the case where the normal velocity equals the mean curvature plus a constant,

$$v_n = a + \kappa$$

is also possible. This motion can be obtained by following the level set

$$\frac{1}{2} - \frac{1}{2}a\sqrt{\frac{\Delta t}{\pi}} \tag{3}$$

[2]Here we have selected zero flux boundary conditions to ensure that the curve meets the boundary at right angles, as is appropriate for certain grain growth models [2]. Alternatively, one may minimize the effects of the boundary by selecting non-reflecting boundary conditions, $\frac{\partial^2 U}{\partial n^2} = 0$, (cf. [16]) or use Dirichlet conditions to produce a constrained motion.

instead of the usual level set of $\frac{1}{2}$ [7].

2.2 Multiple Regions

To obtain a normal velocity equal to the mean curvature for symmetric junctions (e.g., a 120-120-120 degree junction in two dimensions), we may apply the MBO-method for multiple regions:

MBO-Method (Multiple (r) Regions)
BEGIN
(1) For $i = 1, \ldots, r$
Set $U_i(\vec{x}, 0)$ equal to the characteristic function for the ith region.

REPEAT for all steps, j, from 1 to the final step:
BEGIN
(2) For $i = 1, \ldots, r$, starting from $U_i(\vec{x}, (j-1)\Delta t)$,
Apply diffusion to U_i for some time slice, Δt.
i.e., find $U_i(\vec{x}, j\Delta t)$ using $\left\{ \begin{array}{rcl} \frac{\partial U_i}{\partial t} & = & \nabla^2 U_i, \\ \frac{\partial U_i}{\partial n} & = & 0 \text{ on } \partial\mathcal{D}. \end{array} \right.$
(3) "Sharpen" the diffused regions by setting the largest U_i equal to 1 and the others equal to 0 for each point on the domain.
END
END

For any time t, the interfaces are given by

$$\bigcup_{i=1,\ldots,r} \{\vec{x} : U_i(\vec{x}, t) = \max_{j \neq i}\{U_j(\vec{x}, t)\}\}. \tag{4}$$

3 Nonsymmetric Junctions

The MBO-method for regions uses a symmetric projection step which results in an approximation of a 120-120-120 degree junction. We now extend the method to allow for nonsymmetric junctions and justify our algorithm asymptotically and experimentally.

Throughout the next three sections, we will consider the three phase case. See Section 6 for an extension to more phase regions.

3.1 Nonsymmetric Junction Algorithm

We now generalize the sharpening step for the MBO-method to obtain an algorithm for nonsymmetric junctions.

Begin by noting that

$$\begin{aligned} 0 \leq \ & U_i(\vec{x},t) & \leq 1, \\ \sum_{i=1}^{3} \ & U_i(\vec{x},t) & = 1 \end{aligned}$$

for all t since diffusion is linear and $\sum_{i=1}^{3} U_i(\vec{x},0) = 1$. Thus, the ordered triplets, (U_1, U_2, U_3), form a triangular region with corners $(0,0,1), (0,1,0)$ and $(1,0,0)$ in $\Re^3$. By mapping this triangular region onto its corner points we obtain a useful representation of the sharpening step [7, 9]. For example, the symmetric sharpening is obtained by setting

$$(U_1, U_2, U_3) = \begin{cases} (0,0,1) & \text{if } (U_1, U_2, U_3) \in R_1 \\ (0,1,0) & \text{if } (U_1, U_2, U_3) \in R_2 \\ (1,0,0) & \text{if } (U_1, U_2, U_3) \in R_3 \end{cases}$$

where R_1, R_2 and R_3 divide the triangular domain symmetrically, as shown in Figure 2. Other, nonsymmetric, angle configurations are obtained by taking different choices for R_1, R_2 and R_3.

We now develop a method for determining R_1, R_2 and R_3 for curves which meet at a stable θ_1-θ_2-θ_3 angle configuration. To derive this method we note that in the absence of boundary effects (cf. [7]):

> Straight lines which form a junction satisfying the desired angle conditions must remain stationary for all subsequent times, t (e.g., Figure 3).

By enforcing this simple, but necessary condition we are lead to the following algorithm for constructing projection triangles:

Projection Triangle Algorithm
Given an angle configuration θ_1, θ_2, θ_3:

1. Define lines

$$\begin{aligned}\tilde{\Gamma}_{12} &= \{(r, \frac{1}{2}\theta_1) : r > 0\},\\ \tilde{\Gamma}_{13} &= \{(r, -\frac{1}{2}\theta_1) : r > 0\},\\ \tilde{\Gamma}_{23} &= \{(r, \frac{1}{2}\theta_1 + \theta_2) : r > 0\}\end{aligned}$$

   and regions, $\tilde{\Omega}_1, \tilde{\Omega}_2, \tilde{\Omega}_3$ as indicated by Figure 4. Here, a circular domain has been selected for simplicity.

2. Set χ_i equal to the characteristic function for $\tilde{\Omega}_i$, $1 \leq i \leq 3$, as shown in Figure 5.

3. Apply diffusion to each $\chi_i, 1 \leq i \leq 3$, for a time $\tau \leq \Delta t$ as is illustrated in Figure 6.

4. Map each line $\tilde{\Gamma}_{ij}$ onto the projection triangle to form the boundaries, $\tilde{\tilde{\Gamma}}_{ij}$ between regions R_i and R_j,

$$\tilde{\tilde{\Gamma}}_{ij} = \{(\chi_1(\vec{x}), \chi_2(\vec{x}), \chi_3(\vec{x})) : \vec{x} \in \tilde{\Gamma}_{ij}\}$$

   as is illustrated[2] in Figure 7. It is convenient to represent $\tilde{\tilde{\Gamma}}_{ij}$ in polar coordinates centered about the junction (see Figure 8). Using this representation it is straightforward to determine which region a point $P = (r_p, \theta_p)$ belongs since

$$P \in \begin{cases} R_1 & \text{if } \theta_{12}(r_p) \leq \theta_p < \theta_{13}(r_p) \\ R_2 & \text{if } \theta_{13}(r_p) \leq \theta_p < \theta_{23}(r_p) \\ R_3 & \text{if } \theta_{23}(r_p) \leq \theta_p < \theta_{12}(r_p) \end{cases}$$

Having constructed our projection triangle, it is straightforward to derive the following properties [7]:

[2] On non-circular domains, only half of each line $\tilde{\Gamma}_{ij}$ should be mapped (starting from the junction). By connecting this result to the midpoint of the nearest edge of the projection triangle, an excellent approximation of $\tilde{\tilde{\Gamma}}_{ij}$ is formed, provided τ is sufficiently small.

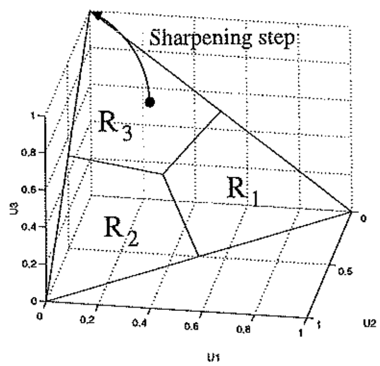


Figure 2: The sharpening decision can be represented using a projection triangle. For the symmetric case, the regions R_1, R_2 and R_3 meet at straight lines which pass through $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ and the midpoints of the edges of the triangular domain.

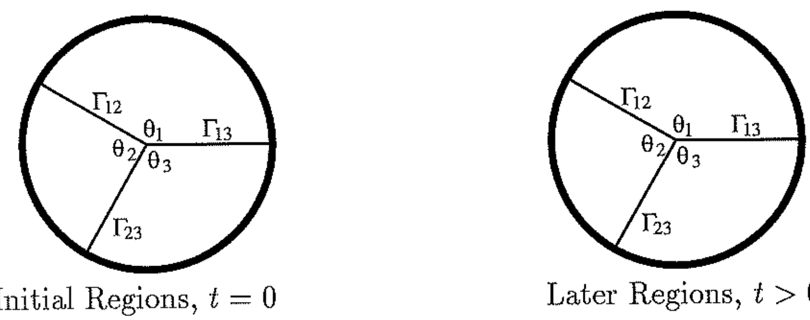


Figure 3: Straight lines forming θ_1-θ_2-θ_3 angles should remain stationary.

- Each boundary curve passes through $(\frac{\theta_1}{2\pi}, \frac{\theta_2}{2\pi}, \frac{\theta_3}{2\pi})$.
- Each curve must also meet the midpoint of an edge of the triangular domain since this case reduces to the MBO-algorithm for two phases.

Note, however, that the lines connecting these endpoints are typically curved. This is quite clearly illustrated for a $150 - 90 - 120$ degree junction in Figure 9 and for a wedge-shaped junction in Figure 10. In fact, only the symmetric case and the $180 - 90 - 90$ degree "T-Junction" are comprised of straight lines (see Figure 11).

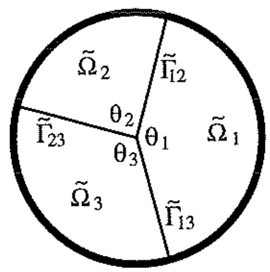


Figure 4: Initial Regions

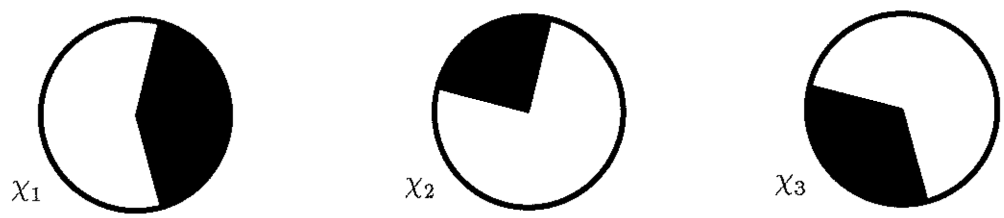


Figure 5: Characteristic Sets

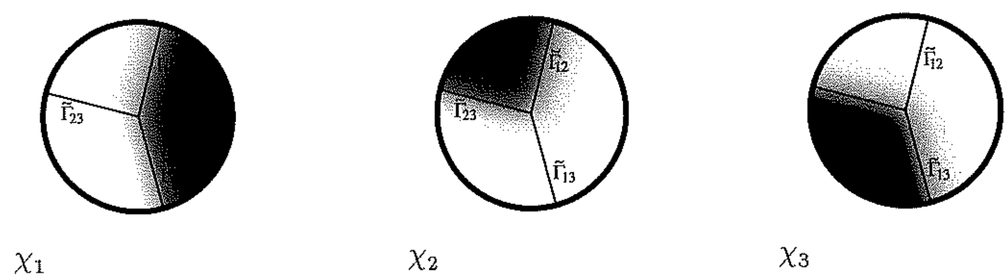


Figure 6: After a Time τ

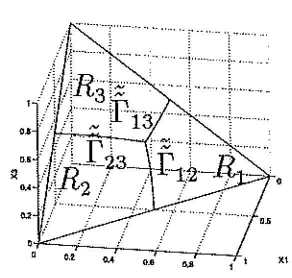


Figure 7: Projection triangle formed by mapping $\tilde{\Gamma}_{ij}$ into the plane $\chi_1 + \chi_2 + \chi_3 = 1$.

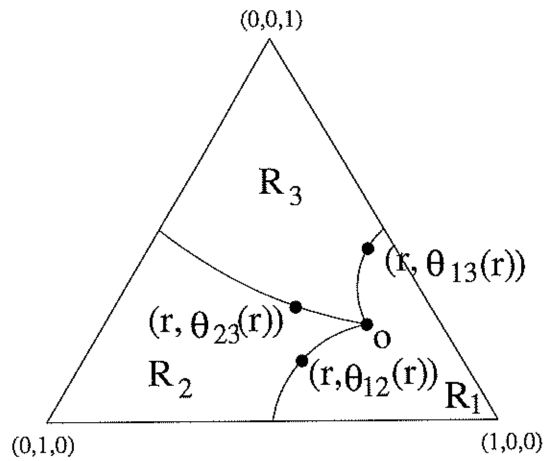


Figure 8: The boundaries between regions are conveniently represented in polar coordinates centered about the junction.

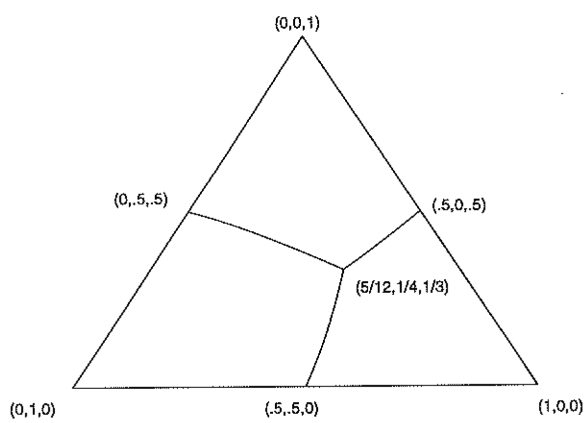


Figure 9: The projection triangle for a $150 - 90 - 120$ degree junction.

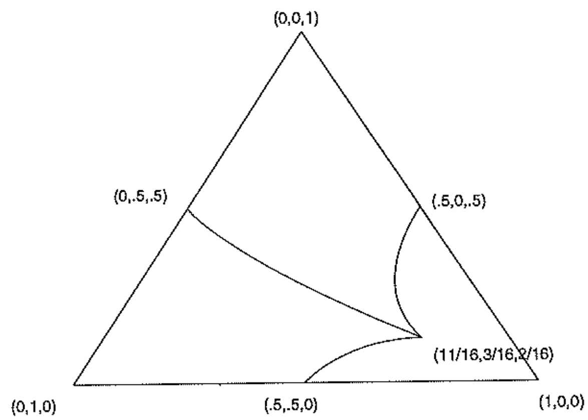


Figure 10: The projection triangle for a $247.5 - 67.5 - 45$ degree junction.

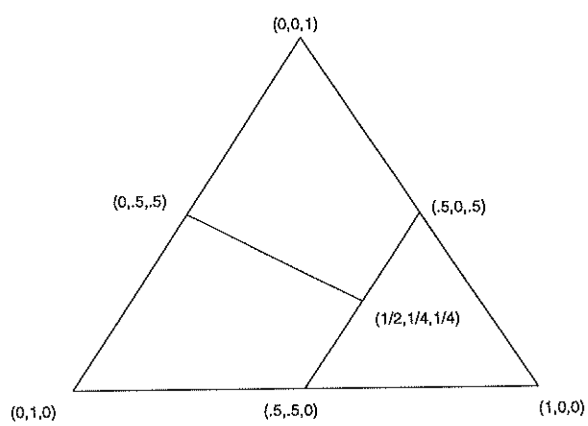


Figure 11: The projection triangle for a $180 - 90 - 90$ degree junction.

3.2 Error Analysis

In the previous subsection, we proposed an algorithm for evolving junctions which meet at a stable $\theta_1 - \theta_2 - \theta_3$ angle configuration. We now outline a derivation that shows, symbolically (using Maple [4]), that each step of the method produces an $\mathcal{O}(\sqrt{\Delta t})$ error in the junction angles which is rapidly dissipated in subsequent steps.

Due to the length of the expressions arising in our derivation, we provide the main steps of the algorithm, but omit most of the intermediate results. See [11] for greater details for the special case of a symmetric junction.

3.2.1 The Initial Junction

We wish to derive an expansion for the angles of a two dimensional triple junction after one step of our method assuming that the angles initially approximate the desired $\theta_1 - \theta_2 - \theta_3$ configuration.

We begin by orienting a polar coordinate system so that some phase region is centered about $\theta = 0$. Denote the initial interfaces by Γ_{12}, Γ_{13} and Γ_{23} and the initial regions by Ω_1, Ω_2 and Ω_3 as in Figure 12.

To represent the small deviations from the $\theta_1 - \theta_2 - \theta_3$ junction configuration we define

$$\begin{aligned}
\epsilon_1 &= \angle\Gamma_{13}\Gamma_{12} - \theta_1, \\
\epsilon_2 &= \angle\Gamma_{12}\Gamma_{23} - \theta_2, \\
\epsilon_3 &= \angle\Gamma_{13}\Gamma_{23} - \theta_3
\end{aligned}$$

where $\angle\Gamma_{ij}\Gamma_{kl}$ is the angle between Γ_{ij} and Γ_{kl}.

In order to carry out our expansions, we want an expression for each interface,

$$\Gamma_{ij} = \{(r, \theta_{\Gamma_{ij}}(r)) : r \geq 0\}$$

for some function, $\theta_{\Gamma_{ij}}(r)$. Using the above definitions it is straightforward to show that

$$\begin{aligned}
\theta_{\Gamma_{12}}(r) &= \frac{1}{2}\theta_1 + \frac{1}{2}\epsilon_1 + \frac{1}{2}\kappa_{12}r + \beta_{12}r^2 + \mathcal{O}(r^3), \\
\theta_{\Gamma_{23}}(r) &= \frac{1}{2}\theta_1 + \theta_2 + \frac{1}{2}\epsilon_2 + +\epsilon_2 + \frac{1}{2}\kappa_{23}r + \beta_{23}r^2 + \mathcal{O}(r^3),
\end{aligned}$$

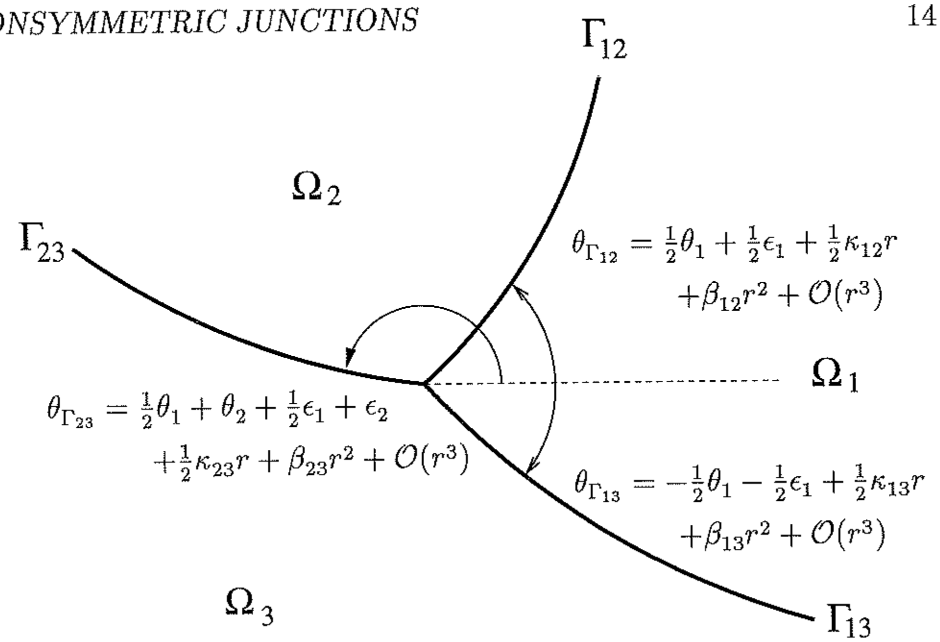


Figure 12: The initial junction.

$$\theta_{\Gamma_{13}}(r) \;=\; -\frac{1}{2}\theta_1 - \frac{1}{2}\epsilon_1 + \frac{1}{2}\kappa_{13} r + \beta_{13} r^2 + \mathcal{O}(r^3)$$

where κ_{ij} is the curvature of line Γ_{ij} at the origin and β_{ij} are constants independent of r.

3.2.2 Approximation of U_i and χ_i

We now want to estimate U_i at time Δt and χ_i at time, τ. Initially,

$$U_i(r,\theta,0) = \begin{cases} 1 & \text{if } (r,\theta) \in \Omega_i \\ 0 & \text{otherwise,} \end{cases}$$

for $1 \le i \le 3$. Thus, the Green's function representation of $U_1(r,\theta,\Delta t)$ gives

$$U_1(r,\theta,\Delta t) \;=\; \frac{1}{4\pi\Delta t}\exp\left(-\frac{r^2}{4\Delta t}\right)\int_0^\infty \exp\left(-\frac{R^2}{4\Delta t}\right)\int_{\theta_{\Gamma_{13}}(R)}^{\theta_{\Gamma_{12}}(R)} \exp\left(\frac{rR\cos(\phi-\theta)}{2\,\Delta t}\right) R\; d\phi\; dR.$$

Replacing the exponential in the inner integral by its series, integrating term by term and applying integration by parts to the result (cf. [7, 11])

yields

$$\begin{aligned}
U_1(r,\theta,\Delta t) &= \frac{\theta_1}{2\pi} + \frac{1}{2\pi}\epsilon_1 + \frac{1}{4\sqrt{\pi}}(\kappa_{12}-\kappa_{13})\sqrt{\Delta t} + \frac{2}{\pi}(\beta_{12}-\beta_{13})\Delta t \\
&+\frac{1}{2\pi}(\kappa_{12}-\kappa_{13})\cos(\theta_1)\cos(\theta)\ r + \frac{1}{2\pi}(\kappa_{12}+\kappa_{13})\sin(\theta_1)\sin(\theta)\ r \\
&+\frac{1}{2\sqrt{\pi}}\sin(\theta_1)\cos(\theta)\left(\frac{r}{\sqrt{\Delta t}}\right) + \frac{1}{2\sqrt{\pi}}\cos(\theta_1)\cos(\theta)\ \epsilon_1\left(\frac{r}{\sqrt{\Delta t}}\right) \\
&+\frac{1}{4\pi}\sin(\theta_1)\cos(\theta_1)\left(\cos^2(\theta)-\sin^2(\theta)\right)\left(\frac{r}{\Delta t}\right)^2 + \text{h.o.t.}
\end{aligned}$$

which may be written in Cartesian coordinates as

$$\begin{aligned}
U_1(x,y,\Delta t) &= \frac{\theta_1}{2\pi} + \frac{1}{2\pi}\epsilon_1 + \frac{1}{4\sqrt{\pi}}(\kappa_{12}-\kappa_{13})\sqrt{\Delta t} + \frac{2}{\pi}(\beta_{12}-\beta_{13})\Delta t \qquad (5)\\
&+\frac{1}{2\pi}(\kappa_{12}-\kappa_{13})\cos(\theta_1)\ x + \frac{1}{2\pi}(\kappa_{12}+\kappa_{13})\sin(\theta_1)\ y \\
&+\frac{1}{2\sqrt{\pi}}\sin(\theta_1)\left(\frac{x}{\sqrt{\Delta t}}\right) + \frac{1}{2\sqrt{\pi}}\cos(\theta_1)\ \epsilon_1\left(\frac{x}{\sqrt{\Delta t}}\right) \\
&+\frac{1}{4\pi}\sin(\theta_1)\cos(\theta_1)\left(\frac{y^2-x^2}{(\Delta t)^2}\right) + \text{h.o.t..}
\end{aligned}$$

To determine the expansion for χ_1, simply set

$$\begin{aligned}
\epsilon_1 = \epsilon_2 = \epsilon_3 &= 0, \\
\kappa_{12} = \kappa_{23} = \kappa_{13} &= 0, \\
\beta_{12} = \beta_{23} = \beta_{13} &= 0
\end{aligned}$$

in Equation (5) to obtain

$$\chi_1(x,y,\tau) = \frac{\theta_1}{2\pi} + \frac{1}{2\sqrt{\pi}}\sin(\theta_1)\left(\frac{x}{\sqrt{\tau}}\right) + \frac{1}{4\pi}\sin(\theta_1)\cos(\theta_1)\left(\frac{y^2-x^2}{\tau^2}\right) + \text{h.o.t..} \qquad (6)$$

Expressions for the remaining U_i and χ_i are easily obtained via rotations of Equations (5) and (6), respectively.

3.2.3 Angle Expansions

We now seek expansions for the angle configuration of the junction after a time Δt.

Begin by letting $\Gamma_{12}^{\Delta t}$, $\Gamma_{23}^{\Delta t}$ and $\Gamma_{13}^{\Delta t}$ be the diffusion-generated approximations to the branches of the junction after a time Δt and parameterize the components of $\Gamma_{ij}^{\Delta t}$ according to

$$\Gamma_{ij}^{\Delta t} = \{(x_{\Gamma_{ij}^{\Delta t}}(s), y_{\Gamma_{ij}^{\Delta t}}(s)) : s \geq 0\}$$

where s represents arclength from the triple point. Similarly define the components of each branch, $\tilde{\Gamma}_{ij}$, of the stationary problem according to

$$\tilde{\Gamma}_{ij} = \{(x_{\tilde{\Gamma}_{ij}}(s), y_{\tilde{\Gamma}_{ij}}(s)) : s \geq 0\}.$$

To approximate the angle between $\Gamma_{12}^{\Delta t}$ and $\Gamma_{13}^{\Delta t}$ we require an expansion for the location of the triple point at time $t = \Delta t$. This is found by substituting our estimates for U_i into

$$\begin{cases} U_1(x_{\Gamma_{ij}^{\Delta t}}(0), y_{\Gamma_{ij}^{\Delta t}}(0), \Delta t) & = & \frac{1}{2\pi}\theta_1, \\ U_2(x_{\Gamma_{ij}^{\Delta t}}(0), y_{\Gamma_{ij}^{\Delta t}}(0), \Delta t) & = & \frac{1}{2\pi}\theta_2 \end{cases}$$

and deriving the series solution for $(x_{\Gamma_{ij}^{\Delta t}}(0), y_{\Gamma_{ij}^{\Delta t}}(0))$.

Our next task is to find the slope of $\Gamma_{ij}^{\Delta t}$ at the triple point. This is accomplished by substituting our expressions for U_i and χ_i into

$$\begin{aligned} \frac{d}{ds}\left[U_i(x_{\Gamma_{ij}^{\Delta t}}(s), y_{\Gamma_{ij}^{\Delta t}}(s), \Delta t)\right]_{s=0} &= \frac{d}{ds}\left[\chi_i(x_{\tilde{\Gamma}_{ij}}(s), y_{\tilde{\Gamma}_{ij}}(s), \tau)\right]_{s=0} \\ \frac{d}{ds}\left[U_j(x_{\Gamma_{ij}^{\Delta t}}(s), y_{\Gamma_{ij}^{\Delta t}}(s), \Delta t)\right]_{s=0} &= \frac{d}{ds}\left[\chi_j(x_{\tilde{\Gamma}_{ij}}(s), y_{\tilde{\Gamma}_{ij}}(s), \tau)\right]_{s=0} \end{aligned}$$

where

$$\begin{aligned} \left(x_{\tilde{\Gamma}_{ij}}(0), y_{\tilde{\Gamma}_{ij}}(0)\right) &= (0,0), \\ \left(x'_{\tilde{\Gamma}_{12}}(0), y'_{\tilde{\Gamma}_{12}}(0)\right) &= \left(\cos\left(\frac{1}{2}\theta_1\right), \sin\left(\frac{1}{2}\theta_1\right)\right), \\ \left(x'_{\tilde{\Gamma}_{13}}(0), y'_{\tilde{\Gamma}_{13}}(0)\right) &= \left(\cos\left(\frac{1}{2}\theta_1\right), -\sin\left(\frac{1}{2}\theta_1\right)\right), \\ \left(x'_{\tilde{\Gamma}_{23}}(0), y'_{\tilde{\Gamma}_{23}}(0)\right) &= \left(\cos\left(\frac{1}{2}\theta_1 + \theta_2\right), \sin\left(\frac{1}{2}\theta_1 + \theta_2\right)\right) \end{aligned}$$

and deriving series solutions for $x'_{\Gamma_{ij}^{\Delta t}}(0)$ and $y'_{\Gamma_{ij}^{\Delta t}}(0)$. These expansions give the slope of $\Gamma_{ij}^{\Delta t}$ at the triple point,

$$m_{ij} = \frac{y'_{\Gamma_{ij}^{\Delta t}}(0)}{x'_{\Gamma_{ij}^{\Delta t}}(0)}.$$

From the slopes of each branch, we see that the approximation to the first angle, θ_1, is given by

$$\angle\Gamma_{12}^{\Delta t}\Gamma_{13}^{\Delta t} = \pi - \arctan\left(\frac{m_{12} - m_{13}}{1 + m_{12}m_{13}}\right).$$

Expanding this in terms of ϵ_i and Δt gives

$$\angle\Gamma_{12}^{\Delta t}\Gamma_{13}^{\Delta t} = \theta_1 + a_{11}\epsilon_1 + a_{12}\epsilon_2 + (c_{11}\kappa_{12} + c_{12}\kappa_{23} + c_{13}\kappa_{13})\sqrt{\Delta t} + \text{h.o.t.}.$$

A similar derivation gives the approximation for the second angle, θ_2,

$$\angle\Gamma_{23}^{\Delta t}\Gamma_{12}^{\Delta t} = \theta_2 + a_{21}\epsilon_1 + a_{22}\epsilon_2 + (c_{21}\kappa_{22} + c_{22}\kappa_{23} + c_{23}\kappa_{13})\sqrt{\Delta t} + \text{h.o.t.}.$$

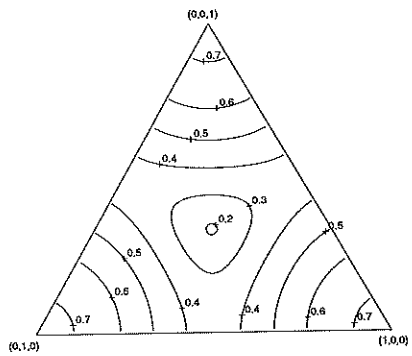
Combining these results into a single equation we obtain

$$\begin{pmatrix} \angle\Gamma_{12}^{\Delta t}\Gamma_{13}^{\Delta t} \\ \angle\Gamma_{23}^{\Delta t}\Gamma_{12}^{\Delta t} \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + A\begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} + C\begin{pmatrix} \kappa_{12} \\ \kappa_{23} \\ \kappa_{13} \end{pmatrix}\sqrt{\Delta t} + \text{h.o.t.}.$$

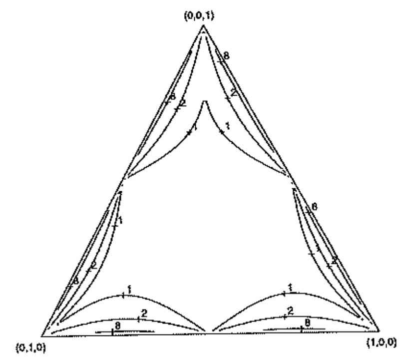
where $A = [a_{ij}]$ and $C = [c_{ij}]$.

Unfortunately, the matrices A and C are far too complicated to reproduce here. However, we do provide a contour plot of the spectral radius of A for each angle configuration $(\theta_1, \theta_2, \theta_3)$ in Figure 13. This plot indicates that the spectral radius of A is always less than 1. Similarly, we find that each element of C is bounded in the interior of the triangle (see Figure 13). Thus, each step of the MBO-method produces an $\mathcal{O}(\sqrt{\Delta t})$ error in the junction angles which is rapidly dissipated during subsequent steps. Summing up such contributions[3] over many time steps, we expect to obtain a rapidly converging geometric sum which gives rise to an $\mathcal{O}(\sqrt{\Delta t})$ error in total. This is an interesting result because it gives an explanation for the stability of junction angles and suggests a source of the $\mathcal{O}(\sqrt{\Delta t})$ error which arises in numerical experiments (see next section).

[3]This summation step is non-rigorous because it assumes, among other things, that κ_{12}, κ_{23} and κ_{13} are bounded independent of Δt.



The spectral radius of A.



The maximum element of C: $\max |c_{ij}|$

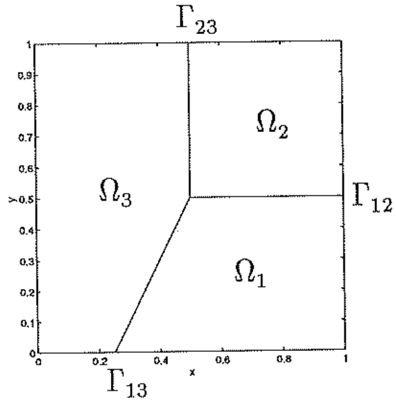
Figure 13: Matrix properties for each angle configuration $(\theta_1, \theta_2, \theta_3)$.

3.3 Numerical Experiments

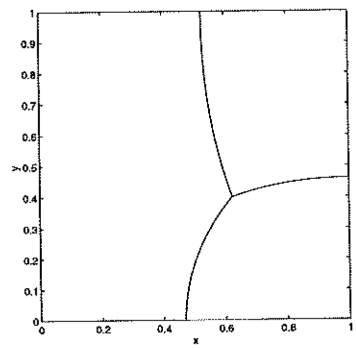
We now apply our algorithm to problems involving nonsymmetric junctions. See also [7, 9] for experimental studies of the 180−90−90 degree "T-Junction" case.

To begin, consider the motion by mean curvature of the three phase problem given in Figure 14. Using our nonsymmetric junction algorithm, the position of the triple point and the change in the area of Ω_1 were compared with the exact results[4] for several Δt. The results from a number of experiments are reported in Table 1.

[4]The "exact results" were computed using Brian Wetton's front tracking code. See [2].



Initial Regions, $t = 0$



Final Regions, $t = 0.1$

Figure 14: A test problem for a $150 - 90 - 120$ degree junction. Here, each $\gamma_{ij} = 1$ and each $e_i = 0$.

Δt	Junction Position		Phase Area Change for Ω_1	
	| *Error* |	*Conv. Rate*[5]	| *Error* |	*Conv. Rate*
0.01	5.23e-03	-	3.80e-03	-
0.005	4.03e-03	0.38	2.55e-03	0.58
0.0025	3.05e-03	0.41	1.75e-03	0.54
0.00125	2.25e-03	0.43	1.20e-03	0.54
0.000625	1.65e-03	0.45	8.35e-04	0.52
0.0003125	1.20e-03	0.46	5.86e-04	0.51

Table 1. Results for a $150 - 90 - 120$ degree junction.

These results are suggestive of an $\mathcal{O}(\sqrt{\Delta t})$ error which is experimentally the same as that found for symmetric junctions using the MBO-method [11, 10, 13].

Our new algorithm can even be applied to wedge-shaped regions or to problems which are initially inconsistent with the desired angle configuration. Consider, for example, the motion by mean curvature of the three phase problem given in Figure 15. Using our nonsymmetric junction algorithm, the position of the triple point and the change in the area of Ω_1 were

[5] If the error for a step of size Δt is $E_{\Delta t}$, then we estimate the convergence rate as $\log_2\left(\left|\frac{E_{2\Delta t}}{E_{\Delta t}}\right|\right)$.

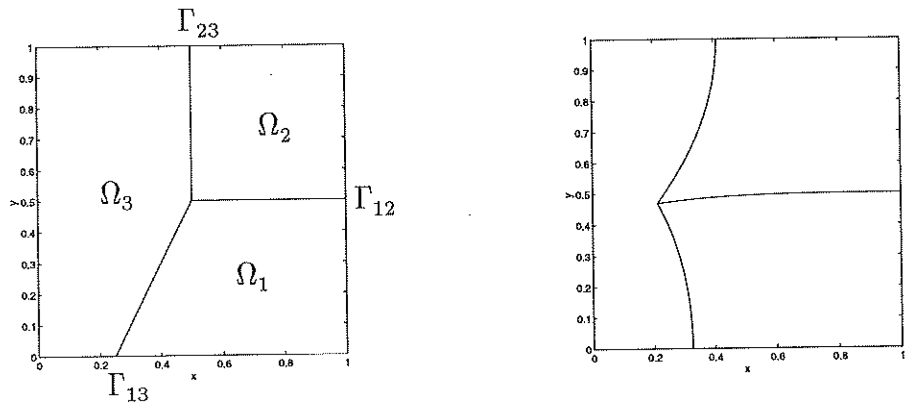


Figure 15: A test problem for a $247.5-67.5-45$ degree junction. Here, each $\gamma_{ij} = 1$ and each $e_i = 0$.

compared with the exact results for several Δt. The results from a number of experiments are reported in Table 2.

Δt	Junction Position		Phase Area Change for Ω_1	
	$\mid Error \mid$	*Conv. Rate*	$\mid Error \mid$	*Conv. Rate*
0.01	2.65e-02	-	1.76e-02	-
0.005	2.48e-02	0.09	1.41e-02	0.33
0.0025	2.14e-02	0.22	1.09e-02	0.37
0.00125	1.73e-02	0.30	8.28e-03	0.40
0.000625	1.34e-02	0.37	6.16e-03	0.43
0.0003125	9.47e-03	0.51	4.30e-03	0.52

Table 2. Results for a $247.5 - 67.5 - 45$ degree junction.

As found in the previous example, the results are suggestive of an $\mathcal{O}(\sqrt{\Delta t})$ error.

4 Generalized Mean Curvature Motions

In the previous section, we described a method that treats nonsymmetric junctions for the case of pure mean curvature flow. We now extend the algorithm to the case where each branch, Γ_{ij}, moves with a normal speed, $v_n = \gamma_{ij}\kappa$.

Although the algorithm that we provide applies to any angle configuration, our asymptotic justification and numerical experiments will assume (for simplicity) the classical condition (2) at triple points which is well-known in the material sciences literature (see, e.g., [14]).

4.1 Generalized Mean Curvature Algorithm

We now generalize diffusion-generated motion to the case where each Γ_{ij} moves with a normal velocity,

$$v_n = \gamma_{ij}\kappa. \tag{7}$$

To begin, let $U_k^{\gamma_{ij}}$ be the solution to $u_t = \gamma_{ij}\nabla^2 u$ after a time Δt, starting from the characteristic function of Ω_k and set $\vec{U}^{\gamma_{ij}} = (U_1^{\gamma_{ij}}, U_2^{\gamma_{ij}}, U_3^{\gamma_{ij}})$. We seek a function, f, which combines $\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}$ and $\vec{U}^{\gamma_{13}}$ into a single result

$$\vec{U} \equiv (U_1, U_2, U_3) = f\left(\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}, \vec{U}^{\gamma_{13}}\right)$$

which can be input to the sharpening step.

Several desirable properties for f are easily identified:

- Certainly, $f\left(\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}, \vec{U}^{\gamma_{13}}\right)$ must reduce to the appropriate $\vec{U}^{\gamma_{ij}}$ far from the triple point. Specifically,

  $$f\left(\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}, \vec{U}^{\gamma_{13}}\right) \approx \vec{U}^{\gamma_{ij}}$$

  for all points near Γ_{ij} which are a distance $d \gg \sqrt{\Delta t}$ away from the triple point.

- We want f to be a smooth combination of the $\vec{U}^{\gamma_{ij}}$ so that the interfaces corresponding to $\vec{U}$ are smooth.

- We will also assume that $\vec{U}$ is a convex combination of $\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}$ and $\vec{U}^{\gamma_{13}}$. This requirement ensures that each component of $\vec{U}$ belongs to $[0, 1]$ and that the components of $\vec{U}$ sum to 1.

One simple family of functions which satisfy these requirements is given by

$$f_n\left(\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}, \vec{U}^{\gamma_{13}}\right) = \frac{\frac{\vec{U}^{\gamma_{12}}}{|U_3^{\gamma_{12}}|^n} + \frac{\vec{U}^{\gamma_{23}}}{|U_1^{\gamma_{23}}|^n} + \frac{\vec{U}^{\gamma_{13}}}{|U_2^{\gamma_{13}}|^n}}{\sum_{i=1}^{3} \frac{U_i^{\gamma_{12}}}{|U_3^{\gamma_{12}}|^n} + \frac{U_i^{\gamma_{23}}}{|U_1^{\gamma_{23}}|^n} + \frac{U_i^{\gamma_{13}}}{|U_2^{\gamma_{13}}|^n}}. \tag{8}$$

The next two subsections justify this choice of f for the cases $n = 1$ and $n = 2$. Larger values of n were found to produce less accurate results on the test problems we tried.

We now summarize by giving the generalized mean curvature algorithm:

Generalized Mean Curvature Algorithm
Given an angle configuration $(\theta_1, \theta_2, \theta_3)$ and coefficients $(\gamma_{12}, \gamma_{23}, \gamma_{13})$:

BEGIN
(1) Construct a projection triangle according to the projection triangle algorithm.
(2) For $i = 1, \ldots, 3$
Set $U_i(\vec{x}, 0)$ equal to the characteristic function for the ith region.

REPEAT for all steps, j, from 1 to the final step:
BEGIN
(3) For each coefficient $\gamma = \gamma_{12}, \gamma_{23}, \gamma_{13}$ and each region $i = 1, 2, 3$,

Find $U_i^\gamma(\vec{x}, j\Delta t)$ using $\begin{cases} \frac{\partial U_i^\gamma}{\partial t} &= \gamma \nabla^2 U_i^\gamma \\ \frac{\partial U_i^\gamma}{\partial n} &= 0 \text{ on } \partial\mathcal{D} \end{cases}$

starting from $U_i^\gamma(\vec{x}, (j-1)\Delta t) = U_i(\vec{x}, (j-1)\Delta t)$.
(4) Set $\vec{U}(\vec{x}, j\Delta t) = f_n\left(\vec{U}^{\gamma_{12}}, \vec{U}^{\gamma_{23}}, \vec{U}^{\gamma_{13}}\right)$ where f_n is given by Equation (8).
(5) "Sharpen" $\vec{U} = (U_1, U_2, U_3)$ according to the projection triangle defined in step (1).
END
END

4.2 Error Analysis

In the previous subsction, we proposed an algorithm for evolving junctions with a normal velocity, $v_n = \gamma_{ij}\kappa$, for arbitrary angle configurations. We now give asymptotic estimates for the angles arising from this algorithm when the classical condition (2) holds.

Begin by letting $\Gamma_{12}^{\Delta t}$, $\Gamma_{23}^{\Delta t}$ and $\Gamma_{13}^{\Delta t}$ be the diffusion-generated approximations to the branches of the junction after a time Δt and let

$$\begin{aligned} \epsilon_1 &= \angle\Gamma_{13}\Gamma_{12} - \theta_1, \\ \epsilon_2 &= \angle\Gamma_{12}\Gamma_{23} - \theta_2, \\ \epsilon_3 &= \angle\Gamma_{13}\Gamma_{23} - \theta_3 \end{aligned}$$

be the initial errors in each junction angle (see Figure 12). As outlined in Section 3.2, it is straightforward (but tedious) to derive asymptotic estimates for the junction angles,

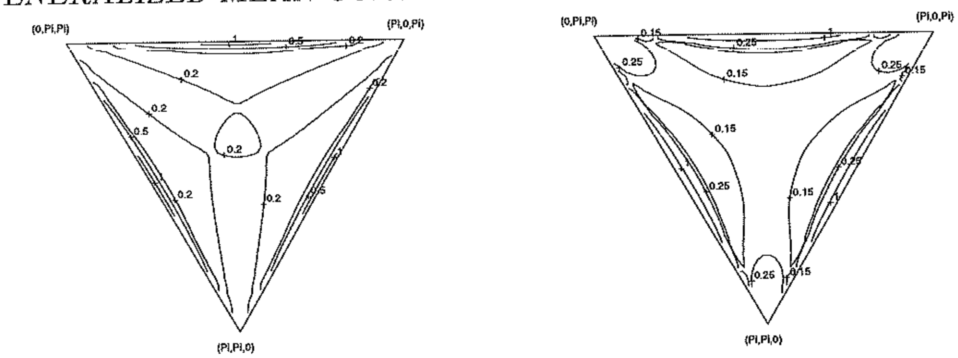
$$\begin{pmatrix} \angle\Gamma_{12}^{\Delta t}\Gamma_{13}^{\Delta t} \\ \angle\Gamma_{23}^{\Delta t}\Gamma_{12}^{\Delta t} \end{pmatrix} = \begin{pmatrix} \theta_1 \\ \theta_2 \end{pmatrix} + A \begin{pmatrix} \epsilon_1 \\ \epsilon_2 \end{pmatrix} + C \begin{pmatrix} \kappa_{12} \\ \kappa_{23} \\ \kappa_{13} \end{pmatrix} \sqrt{\Delta t} + \text{h.o.t.}.$$

Unfortunately, the matrices A and C are far too complicated to reproduce here. However, Figure 16 gives contour plots of the spectral radius of A for each angle configuration $(\theta_1, \theta_2, \theta_3)$ for the choices $f = f_1$ and $f = f_2$ (see Equation (8)).

From these plots, it is clear than that the spectral radius is less than 1 (indeed, it is typically much less than 1) for most angle configurations. For example, the spectral radius is less than 1 whenever the following simple (but crude) bound holds:

$$\max(\theta_1, \theta_2, \theta_3) < \begin{cases} 175.9 \text{ degrees} & \text{if } f = f_1 \\ 173.5 \text{ degrees} & \text{if } f = f_2 \end{cases} \tag{9}$$

Furthermore, contour plots indicate that each element of C is bounded independent of Δt provided each $\gamma_{ij} > 0$. Thus, each step of the method produces an $\mathcal{O}(\sqrt{\Delta t})$ error in the junction angles which is rapidly dissipated during subsequent steps provided the spectral radius of A is less than 1 (e.g., whenever condition (9) holds). Summing up such contributions over many time steps, we expect to obtain a rapidly converging geometric sum which gives rise to an $\mathcal{O}(\sqrt{\Delta t})$ error in total. This is an interesting result because it gives an explanation for the stability of junction angles and suggests a source of the $\mathcal{O}(\sqrt{\Delta t})$ error which arises in numerical experiments (see next section).



Smoothest Selection, $n = 1$ Alternative Choice, $n = 2$

Figure 16: The spectral radii arising for each angle configuration.

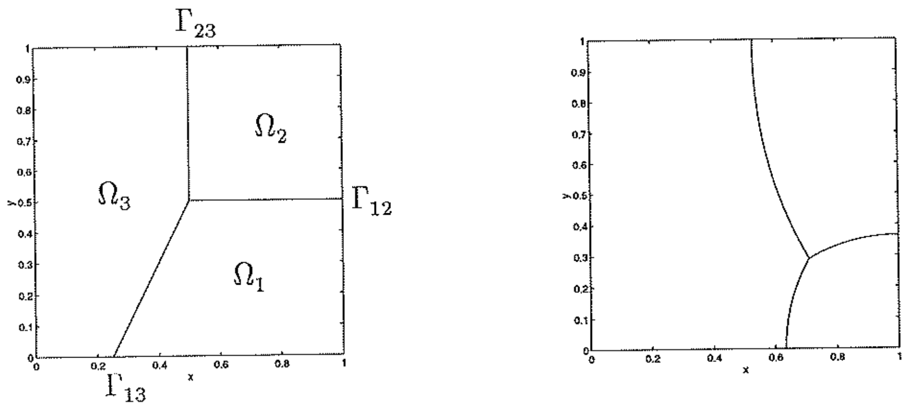
4.3 Numerical Experiments

We now apply the Generalized Mean Curvature Algorithm to the case where each branch of a junction moves with a different normal velocity, $v_n = \gamma_{ij}\kappa$.

For example, consider the evolution of the three phase problem given in Figure 17. Using the Generalized Mean Curvature Algorithm, the position of the triple point and the change in the area of Ω_1 were compared with the exact results[2] for several Δt. The results from a number of experiments are reported in Table 3 for $f = f_1$ and in Table 4 for $f = f_2$.

Δt	Junction Position		Phase Area Change for Ω_1	
	$\mid Error \mid$	*Conv. Rate*	$\mid Error \mid$	*Conv. Rate*
0.01	1.27e-02	-	1.97e-03	-
0.005	9.83e-03	0.37	1.76e-03	0.16
0.0025	7.37e-03	0.42	1.44e-03	0.29
0.00125	5.38e-03	0.45	1.12e-03	0.37
0.000625	3.83e-03	0.49	8.32e-04	0.43
0.0003125	2.67e-03	0.52	6.01e-04	0.47

Table 3. Results for a $150 - 90 - 120$ degree junction for $f = f_1$.



Initial Regions, $t = 0$ Final Regions, $t = 0.2$

Figure 17: A test problem for a $150 - 90 - 120$ degree junction. Here, $(\gamma_{12}, \gamma_{23}, \gamma_{13}) = \left(\sin\left(\frac{5}{6}\pi\right), \sin\left(\frac{1}{2}\pi\right), \sin\left(\frac{2}{3}\pi\right)\right)$ and each $e_i = 0$.

Δt	Junction Position		Phase Area Change for Ω_1	
	$\mid$ *Error* $\mid$	*Conv. Rate*	$\mid$ *Error* $\mid$	*Conv. Rate*
0.01	1.03e-02	-	4.76e-03	-
0.005	7.41e-03	0.47	3.00e-03	0.66
0.0025	5.32e-03	0.48	1.96e-03	0.62
0.00125	3.79e-03	0.49	1.31e-03	0.58
0.000625	2.67e-03	0.50	8.94e-04	0.55
0.0003125	1.87e-03	0.51	6.12e-04	0.55

Table 4. Results for a $150 - 90 - 120$ degree junction for $f = f_2$.

These results are suggestive of an $\mathcal{O}(\sqrt{\Delta t})$ error which is experimentally the same as that found for pure motion by mean curvature.

Although either choice $f = f_1$ or $f = f_2$ is adequate for a wide variety of problems (see previous section), we usually select $f = f_2$ for our simulations. The errors arising from this choice are often more regular (e.g., compare Tables 3 and 4) which is a desirable property for determining an appropriate step size and for developing accurate, extrapolated algorithms [13].

5 Multiphase Motions

In the previous section we described a method to treat the case where each branch, Γ_{ij}, of a junction moves with a normal speed, $v_n = \gamma_{ij}\kappa$. We now extend our algorithm to allow for more general multiphase motions involving bulk energies (e.g., Figure 1) and justify our algorithm experimentally.

5.1 Multiphase Motion Algorithm

To carry out a sharpening appropriate for the multiphase model, we must construct new projection triangles. In particular, our projection triangles must satisfy the following:

- Along each edge, the sharpening decision must reduce to the case of two phase flow (3) since edges correspond to regions which are infinitely far from triple points [7].
- In the limit $\Delta t \to 0$, the projection triangle must coincide with the case $(e_1, e_2, e_3) = (0, 0, 0)$ to obtain junction angles which are consistent with the desired configuration.

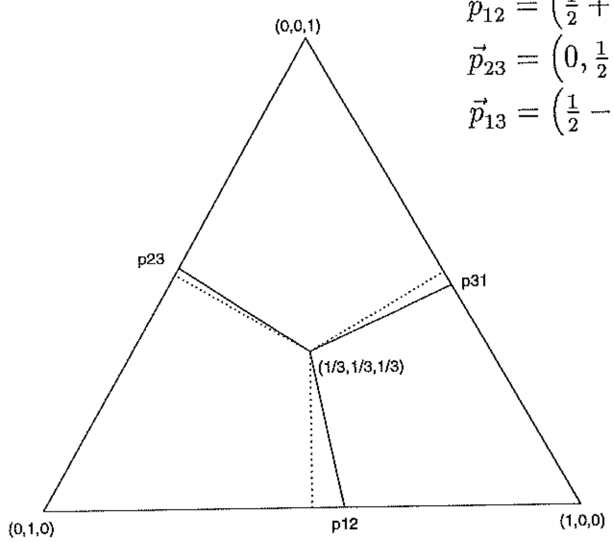
For the special case of symmetric junctions, these objectives are easily attained. We simply set each branch of the decision triangle to a straight line from $(\frac{1}{3}, \frac{1}{3}, \frac{1}{3})$ to the point dictated by Equation (3). See Figure 18 for an illustration of this construction.

For nonsymmetric junctions, appropriate projection triangles may be constructed by scaling and rotating the result of our original algorithm (see, e.g., Figure 19) as follows:

Projection Triangle Algorithm for Multiphase Motion
Given an angle configuration $(\theta_1, \theta_2, \theta_3)$ and bulk energies (e_1, e_2, e_3):

1. Construct a projection triangle using the algorithm given in Section 3.1. Represent the boundaries between the regions R_i and R_j of the triangle in polar coordinates, $\{(r, \theta_{ij}(r))\}$, as is shown in Figure 8.
2. Rotate and scale each curve defined by $\theta_{ij}(\cdot)$ according to

$$\tilde{\theta}_{ij}(r) = \theta_{ij}\left(\frac{r_0}{r_{\vec{p}_{ij}}} r\right) + \theta_{\vec{p}_{ij}} - \theta_{ij}(r_0)$$



$$\vec{p}_{12} = \left(\tfrac{1}{2} + \tfrac{1}{2}(e_1 - e_2)\sqrt{\tfrac{\Delta t}{2\pi}}, \tfrac{1}{2} - \tfrac{1}{2}(e_1 - e_2)\sqrt{\tfrac{\Delta t}{2\pi}}, 0\right)$$
$$\vec{p}_{23} = \left(0, \tfrac{1}{2} + \tfrac{1}{2}(e_2 - e_3)\sqrt{\tfrac{\Delta t}{2\pi}}, \tfrac{1}{2} - \tfrac{1}{2}(e_2 - e_3)\sqrt{\tfrac{\Delta t}{2\pi}}\right)$$
$$\vec{p}_{13} = \left(\tfrac{1}{2} - \tfrac{1}{2}(e_3 - e_1)\sqrt{\tfrac{\Delta t}{2\pi}}, 0, \tfrac{1}{2} + \tfrac{1}{2}(e_3 - e_1)\sqrt{\tfrac{\Delta t}{2\pi}}\right)$$

Figure 18: The projection triangle for a $120-120-120$ degree junction with (solid) and without (dotted) a constant component to the motion.

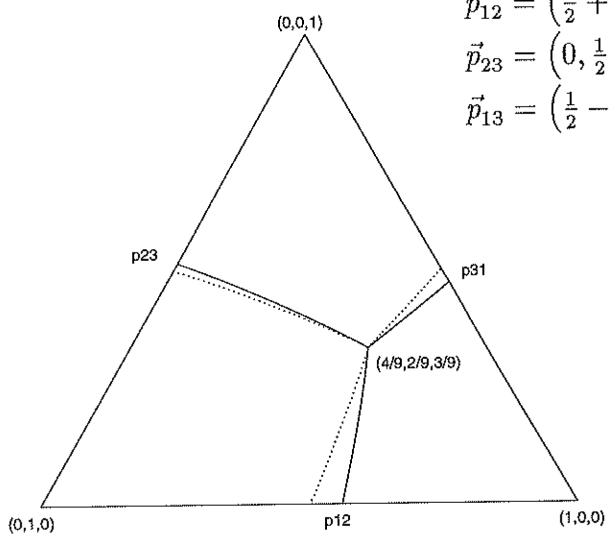
where

$$\begin{aligned}
\vec{p}_{ij} &= \left(\frac{1}{2} + \frac{1}{2}(e_i - e_j)\sqrt{\frac{\Delta t}{2\pi}}\right)\hat{e}_i + \left(\frac{1}{2} + \frac{1}{2}(e_j - e_i)\sqrt{\frac{\Delta t}{2\pi}}\right)\hat{e}_j, \\
r_0 &= \left\| \left(\frac{\theta_1}{2\pi}, \frac{\theta_2}{2\pi}, \frac{\theta_3}{2\pi}\right) - \frac{1}{2}\left(\hat{e}_i + \hat{e}_j\right) \right\|, \\
\hat{e}_i &= \begin{cases} (1,0,0) & \text{if } i = 1 \\ (0,1,0) & \text{if } i = 2 \\ (0,0,1) & \text{if } i = 3 \end{cases}
\end{aligned}$$

and $(r_{\vec{p}_{ij}}, \theta_{\vec{p}_{ij}})$ are the polar coordinates of $\vec{p}_{ij}$ to obtain each branch of the desired projection triangle,

$$\left\{\left(r, \tilde{\theta}_{ij}(r)\right) : 0 \le r \le r_{\vec{p}_{ij}}\right\}.$$

Combining this algorithm with that of the previous section gives a method for evolving junctions according to the multiphase model:



$$\vec{p}_{12} = \left(\tfrac{1}{2} + \tfrac{1}{2}(e_1 - e_2)\sqrt{\tfrac{\Delta t}{2\pi}}, \tfrac{1}{2} - \tfrac{1}{2}(e_1 - e_2)\sqrt{\tfrac{\Delta t}{2\pi}}, 0\right)$$
$$\vec{p}_{23} = \left(0, \tfrac{1}{2} + \tfrac{1}{2}(e_2 - e_3)\sqrt{\tfrac{\Delta t}{2\pi}}, \tfrac{1}{2} - \tfrac{1}{2}(e_2 - e_3)\sqrt{\tfrac{\Delta t}{2\pi}}\right)$$
$$\vec{p}_{13} = \left(\tfrac{1}{2} - \tfrac{1}{2}(e_3 - e_1)\sqrt{\tfrac{\Delta t}{2\pi}}, 0, \tfrac{1}{2} + \tfrac{1}{2}(e_3 - e_1)\sqrt{\tfrac{\Delta t}{2\pi}}\right)$$

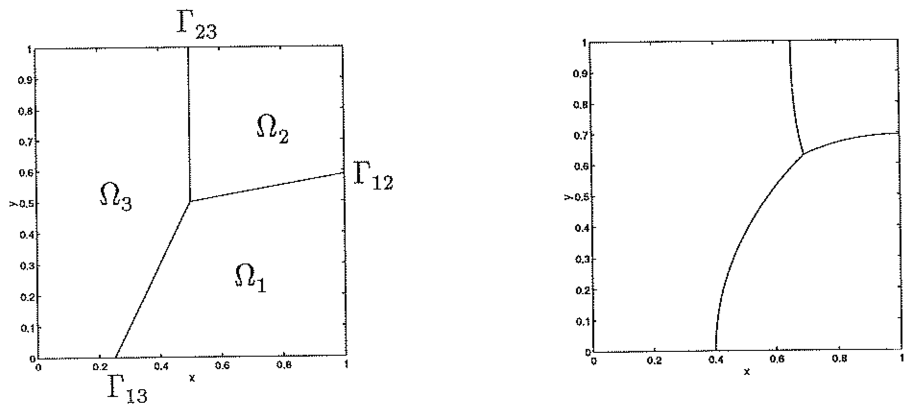
Figure 19: The projection triangle for a $160-80-120$ degree junction with (solid) and without (dotted) a constant component to the motion.

Multiphase Motion Algorithm
Given an angle configuration $(\theta_1, \theta_2, \theta_3)$, coefficients $(\gamma_{12}, \gamma_{23}, \gamma_{13})$
and bulk energies (e_1, e_2, e_3):

BEGIN
(1) Construct a projection triangle according to
the Projection Triangle Algorithm for Multiphase Motion.
(2) Carry out steps (2)-(5) of the Generalized Mean Curvature Algorithm
using the projection triangle derived in step (1).
END

5.2 Numerical Experiments

We now apply the Multiphase Motion Algorithm to the case where each branch of a junction moves with a different normal velocity, $v_n = \gamma_{ij}\kappa + e_i - e_j$.



Initial Regions, $t = 0$ Final Regions, $t = 0.1$

Figure 20: A test problem for a $160 - 80 - 120$ degree junction. Here, $(\gamma_{12}, \gamma_{23}, \gamma_{13}) = \left(\sin\left(\frac{8}{9}\pi\right), \sin\left(\frac{4}{9}\pi\right), \sin\left(\frac{2}{3}\pi\right)\right)$ and $(e_1, e_2, e_3) = \left(0, 2, \frac{1}{2}\right)$.

For example, consider the evolution of the three phase problem given in Figure 20. Using the Multiphase Motion Algorithm, the position of the triple point and the change in the area of Ω_1 were compared with the exact results[2] for several Δt. The results from a number of experiments are reported in Table 5.

Δt	Junction Position		Phase Area Change for Ω_1	
	$\mid$ *Error* $\mid$	*Conv. Rate*	$\mid$ *Error* $\mid$	*Conv. Rate*
0.01	6.82e-02	-	2.61e-02	-
0.005	4.64e-02	0.56	1.83e-02	0.51
0.0025	3.20e-02	0.54	1.29e-02	0.51
0.00125	2.22e-02	0.53	9.11e-03	0.50
0.000625	1.55e-02	0.52	6.43e-03	0.50
0.0003125	1.09e-02	0.51	4.56e-03	0.50

Table 5. Results for a $160 - 80 - 120$ degree junction.

These results are suggestive of an $\mathcal{O}(\sqrt{\Delta t})$ error which is experimentally the same as that found for pure motion by mean curvature.

6 Shape Changes with Many Phase Regions

In the previous section, we described a method for evolving a three-phase junction with a normal velocity, $v_n = \gamma_{ij}\kappa + e_i - e_j$. To extend this method to r phase regions, we apply two additional considerations:

1. For each point on the domain, the three largest $U_i, 1 \leq i \leq r$, are sharpened according to the projection triangle for those components. All remaining components are set to zero during sharpening.

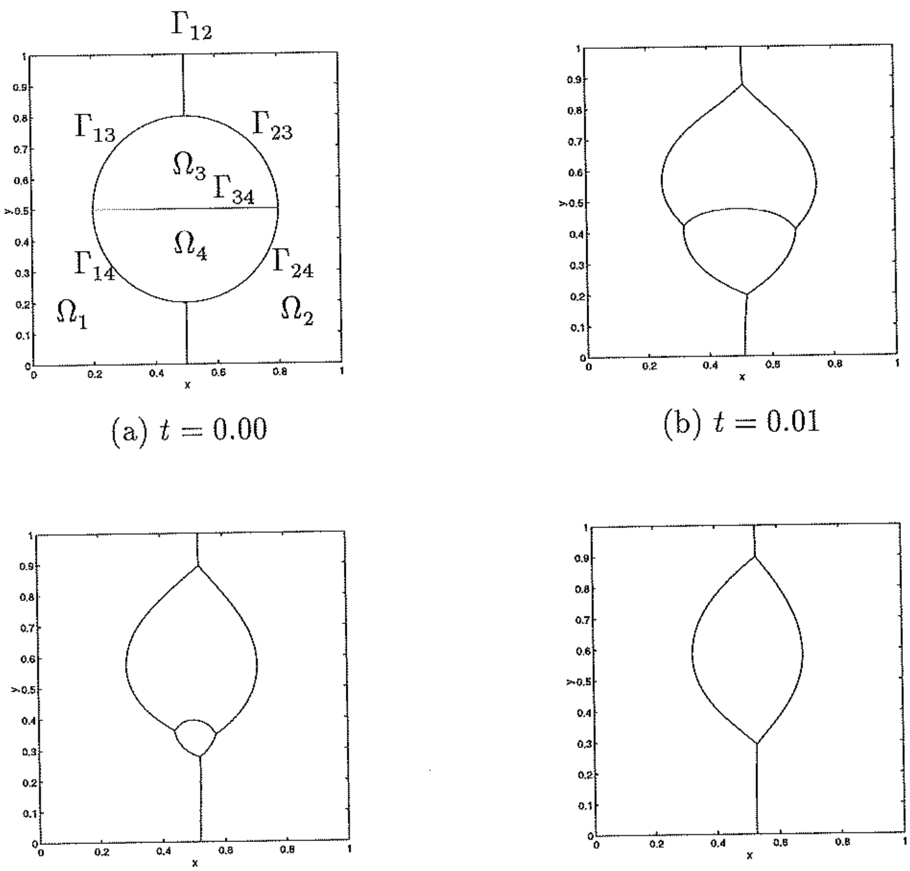
2. The function f (see Equation (8)) is extended to r phase regions:

$$f(\vec{U}) = \frac{\sum_{1 \leq i < j \leq r} \frac{\vec{U}^{\gamma_{ij}}}{|\prod_{1 \leq k \leq r: k \neq i, k \neq j} U_k^{\gamma_{ij}}|}}{\sum_{1 \leq i < j \leq r} \frac{\sum_{1 \leq \ell \leq r} U_\ell^{\gamma_{ij}}}{|\prod_{1 \leq k \leq r: k \neq i, k \neq j} U_k^{\gamma_{ij}}|}}$$

Applying these modifications to the Multiphase Motion Algorithm gives a method for approximating the model (1) when many phase regions are present. We have found that the results from this method agree with the recent variational approach given in [16] even when topological mergings and breakings occur.

For example, consider the evolution of the four phase problem given in Figure 21a. Using our diffusion-generated approach with a step size of $\Delta t = 0.000125$, the interfaces were determined for several times, t (see Figures 21b-d). These results agree well with the variational approach (cf. Figure 22).

Our new algorithm also naturally treats problems which involve the formation of junctions. Consider, for example, the evolution of the four regions given in Figure 23a. Using our diffusion-generated approach with a step size $\Delta t = 0.00025$, the interfaces were determined for several times, t. Here, we find that the interface between the regions Ω_1 and Ω_2 travels to the right to form two new junctions (see Figure 23b). These triple points eventually move to the top and bottom of regions Ω_3 and Ω_4 as is shown in Figures 23c and 23d. It is noteworthy that this example cannot be treated using the variational approach [16], since that method is inconsistent with the given values of γ_{ij}.



(a) $t = 0.00$

(b) $t = 0.01$

(c) $t = 0.02$

(d) $t = 0.03$

Figure 21: A test problem at various times, t. Here $(e_1, e_2, e_3, e_4) = (0.5, 1, 0, 2)$, $(\gamma_{12}, \gamma_{13}, \gamma_{14}, \gamma_{23}, \gamma_{24}, \gamma_{34}) = (1.25, 0.75, 1.25, 1, 1.5, 1)$ and all angles are prescribed by the classical condition (2).

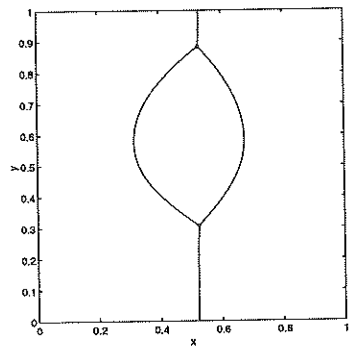


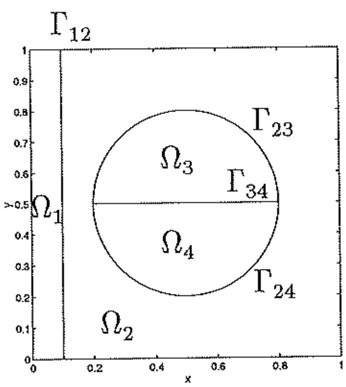
Figure 22: The solution from the variational approach at $t = 0.03$.

7 Summary

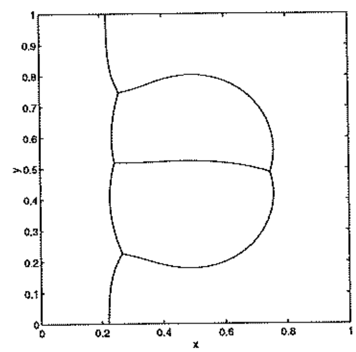
In this work, we have presented a diffusion-generated approach for evolving multiple junctions according to the multiphase model (1). Our method naturally treats topological mergings and breakings, produces no overlapping regions or vacuums and can be made very fast. We have also shown that our approach may be applied to an important class of problems which cannot be treated using other methods (see previous section).

Asymptotic expansions were also given to explain why our method reproduces the correct junction angles (to within $\mathcal{O}(\sqrt{\Delta t})$) and numerical studies were provided to show that our approach agrees with front tracking [2] and a recent variational method [16] on a variety of simple problems.

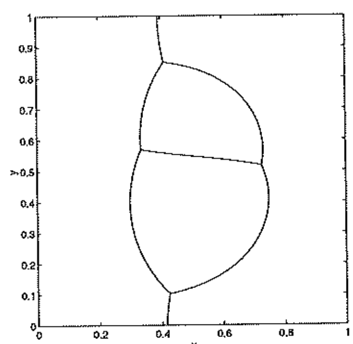
Further work suggested by the results of this paper include a more detailed theoretical investigation of our method and an extension to the full range of possible model problems (Currently, our approach cannot be applied if some γ_{ij} is sufficiently small. See Section 4.2). Finally, extensions to a variety of constrained motions would be of interest (e.g., [12]).



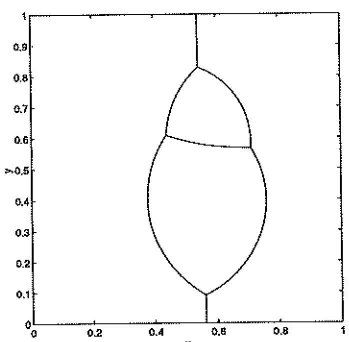
(a) $t = 0.00$



(b) $t = 0.03$



(c) $t = 0.06$



(d) $t = 0.09$

Figure 23: A test problem at various times, t. Here $(e_1, e_2, e_3, e_4) = (0, 4, 1, 0)$, $(\gamma_{12}, \gamma_{13}, \gamma_{14}, \gamma_{23}, \gamma_{24}, \gamma_{34}) = (1, 1, 1, \mathrm{sqrt}(3)/2, 1, 0.5)$ and all angles are prescribed by the classical condition (2).

8 Acknowledgments

I would like to thank Barry Merriman and Stan Osher for helpful discussions. I would also like to thank Brian Wetton and Hong-Kai Zhao for the use of their tracking and variational codes.

References

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

[2] L. Bronsard and B.T.R. Wetton. A numerical method for tracking curve networks moving with curvature motion. *Journal of Computational Physics*, 120(1):66–87, 1993.

[3] G. Caginalp. The dynamics of a conserved phase field system: Stefan-like, Hele-Shaw, and Cahn-Hilliard models as asymptotic limits. *IMA J. Applied Math.*, 44(1):77–94, 1990.

[4] B.W. Char, K.O. Geddes, G.H. Gonnet, B.L. Leong, M.B. Monagan, and S.M. Watt. *Maple V Language Reference Manual.* Springer-Verlag, 1991.

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

[6] J. Glazier, M. Anderson, and G. Grest. Coarsening in the two-dimensional soap froth and the large-Q Potts model: a detailed comparison. *Philosophical Magazine B*, 62:615–645, 1990.

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

[8] B. Merriman, J. Bence, and S. Osher. Diffusion generated motion by mean curvature. In J.E. Taylor, editor, *Computational Crystal Growers Workshop*, pages 73–83. American Mathematical Society, Providence, Rhode Island, 1992.

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

[10] S.J. Ruuth. An algorithm for generating motion by mean curvature. In *Proc. 12th International Conference on Analysis and Optimization of Systems Images, Wavelets and PDE's*, pages 82–91, Paris, France, 1996.

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

[12] S.J. Ruuth. A diffusion-generated approach to a nonlocal, volume-preserving motion. *In Preparation*, 1997.

[13] S.J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *In Preparation*, 1997.

[14] C.S. Smith. Grain shapes and other metallurgical applications of topology. In *Metal Interfaces*, pages 65–108. American Society for Metals, Cleveland, 1952.

[15] J.E. Taylor, J.W. Cahn, and C.A. Handwerker. I-Geometric models of crystal growth. *Acta Metall. Meter.*, 40(7):1443–1474, 1992.

[16] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *Journal of Computational Physics*, 127:179–195, 1996.