Efficient Algorithms for Diffusion-Generated Motion by Mean Curvature

Steven J. Ruuth*

July 28, 1997

Abstract

The problem of simulating the motion of evolving surfaces with junctions according to some curvature-dependent speed arises in a number of applications. By alternately diffusing and sharpening characteristic functions for each region, a variety of motions have been obtained which allow for topological mergings and breakings and produce no overlapping regions or vacuums. However, the usual finite difference discretization of these methods are often excessively slow when accurate solutions are sought, even in two dimensions.

We propose a new, spectral discretization of these diffusion-generated methods which obtains greatly improved efficiency over the usual finite difference approach. These efficiency gains are obtained, in part, through the use of a quadrature-based refinement technique, by integrating Fourier modes exactly and by neglecting the contributions of rapidly decaying solution transients. Indeed, numerical studies demonstrate that the new algorithm is often more than 1000 times faster than the usual finite difference discretization.

Our findings are demonstrated on several examples.

*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.

1 Introduction

In a variety of applications, one wants to follow the motion of a front that moves with some curvature-dependent speed. Such motions can be particularly challenging to approximate when more than two phase regions are present because junctions of moving surfaces can occur. To simulate the evolution of such models, a number of numerical methods have been developed.

Front tracking methods (e.g., [4]), 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 are often possible, front tracking methods can be impractical to implement, especially in more than two dimensions.

Other approaches also have limitations. Monte-Carlo methods for Potts models (e.g., [10]) can introduce unwanted anisotropy into the motion due to the spatial mesh [24] and are typically too slow to find accurate approximations of the model. Phase field methods (e.g., [5]) are often inherently too expensive for practical computation [14] 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., each interface moves with a normal velocity equal to its mean curvature, κ), a method (MBO) based on the model of diffusion-dependent motion of level sets was proposed by Merriman, Bence and Osher [13, 14]. This method naturally handles complicated topological changes with junctions in several dimensions and has been proven to converge when two phase regions are present [8, 2]. Furthermore, a generalization [19] of this method allows each interface, Γ_{ij}, to move with a normal velocity,

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

as is shown in Figure 1. Unfortunately, the usual finite difference discretization of these methods is often exceedingly slow when accurate results are sought, even in two dimensions.

Alternatively, a recent variational approach [25] may be used to approximate the motion (1) when topological mergings and breakings occur. This

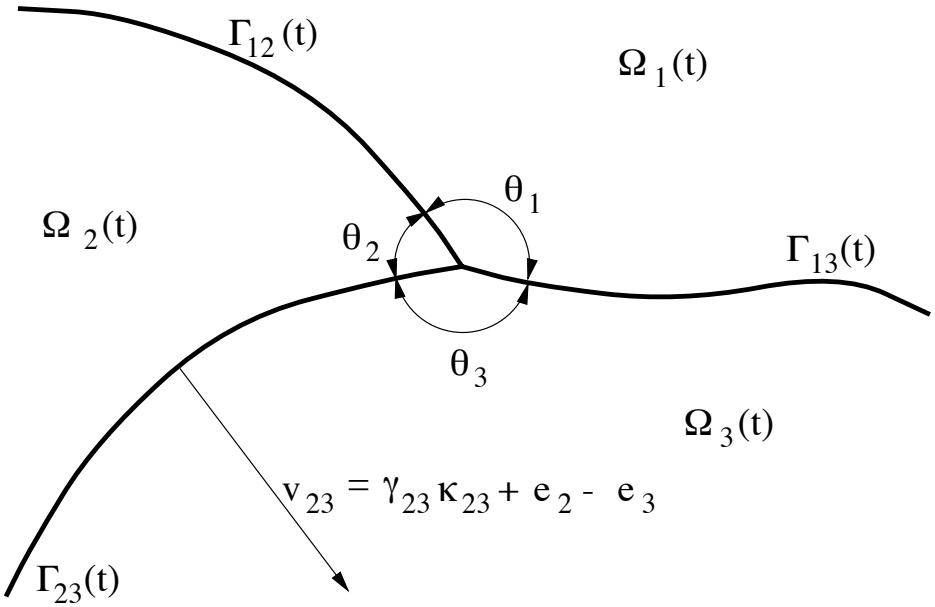


Figure 1: The interfaces, Γ_{ij}, move with a normal velocity $v_{ij} = \gamma_{ij}\kappa_{ij} + e_i - e_j$ and are subject to angles θ_1, θ_2, θ_3.

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., [22])

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

at triple points.

In this paper, we propose a new, spectral discretization for the diffusion-generated methods which obtains greatly improved efficiency over the usual finite difference discretization. Although these algorithms are given (for simplicity) for the MBO-method, we note that they may also be applied to the diffusion-generated approach described in [19]. Our algorithms, when applied to the MBO-method give a simple, fast way of approximating motion by mean curvature and when applied to the diffusion-generated approach [19] provide a practical tool, not available hitherto, for accurately treating a wide variety of motions described by the multiphase model (1). An outline of the paper follows.

In Section 2, we give the MBO-method for two phase and multiple phase problems. For the case of the finite difference discretizations originally proposed [13], the selection of the step size is discussed and several limitations of the method are identified.

In Section 3, a new, spectral method for the realization of the MBO-method is proposed and described in detail. A spatial discretization is given and an efficient quadrature for calculating the corresponding Fourier coefficients is provided. This quadrature obtains accurate approximations to the front using a piecewise linear approximation to the surface and a gradual refinement technique. Unequally spaced transform methods for the rapid evaluation of the Fourier coefficients are also applied.

Section 4 gives a comparison of the proposed method and the usual finite difference approach. In particular, numerical experiments are presented to illustrate the efficiency gains which arise from our method.

In Section 5, we use our algorithms to examine the numerical convergence properties of the MBO-method and apply our algorithms to the motion by mean curvature of surfaces. This section also demonstrates that extrapolation can be used in conjunction with the new, spectral method to produce improved estimates of certain quantities of interest (e.g., phase areas).

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 [13, 14]. In this section, we describe the method for the two phase and multiple phase problems.

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:

[1]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 [4]. Alterna-

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[1] 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.

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 $i = 1, \ldots, r$

---

tively, one may minimize the effects of the boundary by selecting non-reflecting boundary conditions, $\frac{\partial^2 U}{\partial n^2} = 0$, (cf. [25]) or use Dirichlet conditions to produce a constrained motion.

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 $\begin{cases} \frac{\partial U_i}{\partial t} &= \nabla^2 U_i, \\ \frac{\partial U_i}{\partial n} &= 0 \text{ on } \partial\mathcal{D}. \end{cases}$
(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

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

2.3 Time Step Selection

To accurately resolve the motion of features of the interface, it is also important to select Δt appropriately. In particular, diffusion must proceed long enough so that the motion of the interface over each step can be resolved by the spatial discretization. For the case of a finite difference discretization, the level set $U = \frac{1}{2}$ must move at least one grid point, otherwise the interface remains stationary [14]. This produces the restriction that

$$(\text{speed of motion of the interface}) \times \Delta t \gg \text{grid spacing}$$

Letting κ be the curvature and h the grid spacing, we arrive at a restriction for the finite difference approach [14],

$$\kappa\Delta t \gg h. \tag{2}$$

As we shall see, the restriction (2) does not appear for the new, spectral method that we propose in Section 3.

2.4 Limitations of Finite Difference Discretizations

We have seen from the previous section that we are restricted in our choice of h for finite difference discretizations of the MBO-method. Specifically, once a sufficiently small Δt is selected, the mesh spacing, h, must be chosen small enough (see Eq. (2)) so that the level set moves at least one grid point, otherwise the sharpening step leaves the front stationary.

Satisfying the restriction (2) can be computationally impractical even for smooth, two phase problems in two dimensions. Consider, for example, the motion by mean curvature of the boundary of a spiral-shaped region. Since the local curvature of the boundary of such a problem can vary tremendously, it may be impractical to satisfy (2) everywhere using a uniform mesh.

To achieve a more efficient finite difference algorithm, one might consider discretizing the MBO-method using a local mesh refinement at the level of the PDE. However, carrying out local mesh refinement is rather involved for level set methods when curvature terms arise (see [15]). An alternative approach is to place a narrow band of grid points around the front (cf. [1]). Even this optimized, finite difference approach can lead to a prohibitive number of operations per step when an accurate solution is sought.

For example, consider the motion by mean curvature of a smooth curve. For such a curve, each step of the MBO-method produces an $\mathcal{O}(\Delta t^2)$ error in the position of the front [18]. To preserve the overall accuracy of the method, grid points must be at most a distance $\mathcal{O}(\Delta t^2)$ apart since each step produces an error which is comparable to the mesh spacing. Noting that the front travels a distance $\mathcal{O}(\Delta t)$ per step of the method, it is clear that a minimum of $\mathcal{O}\left(\frac{1}{\Delta t^3}\right)$ grid points are needed to safely band a curve (see, e.g., Figure 2). Thus, a minimum of $\mathcal{O}\left(\frac{1}{\Delta t^3}\right)$ operations per step are required to preserve the overall accuracy of the method, which is often prohibitively expensive when accurate results are sought.

A further limitation of the finite difference approach is that the error is not regular. Specifically, very small differences in the position of the level set $1/2$ before sharpening can produce jumps in the front location after sharpening. This type of error is undesirable because it makes the construction of higher order accurate, extrapolated results impractical. Figures 3 and 4 illustrate how a small change in the position of the level set $1/2$ can lead to a jump in the front location after sharpening. (We shall see in the next two sections that our proposed method essentially eliminates the spatial error to allow for

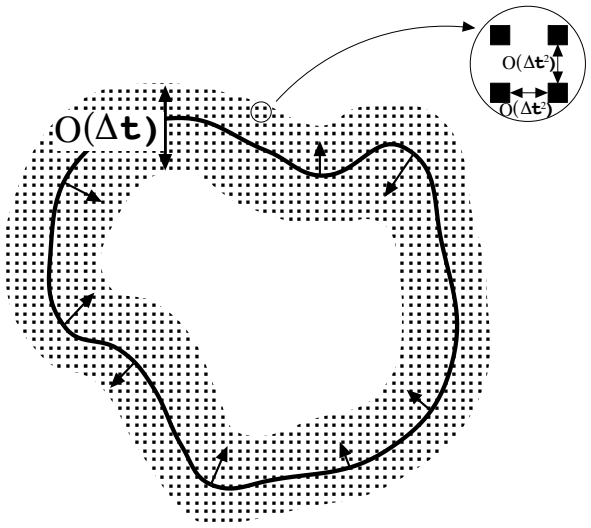


Figure 2: A banded, finite difference mesh.

higher order accurate extrapolations in Δt.)

To avoid the limitations outlined in this section, we introduce a new, spectral method for realizing the MBO-method in the next section.

3 A New, Spectral Method

As we shall see later in this section, accurate computation of solutions using the usual finite difference discretization of the MBO-method can be expensive, even for simple two dimensional problems. Since we are mainly interested in three dimensional problems or problems involving more than two phases, a faster method is desired. This section describes a new, spectral method for realizing the MBO-method which is typically much faster than the usual finite difference approach.

For notational simplicity, the algorithm focuses on the two phase case over the domain, $\mathcal{D} = [0,1] \times [0,1]$. Certain extensions to three spatial dimensions and more phases are also discussed.

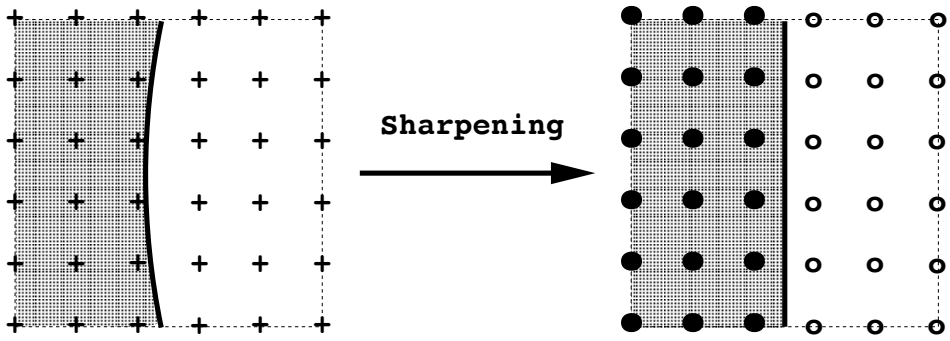


Figure 3: Sharpening a shape.

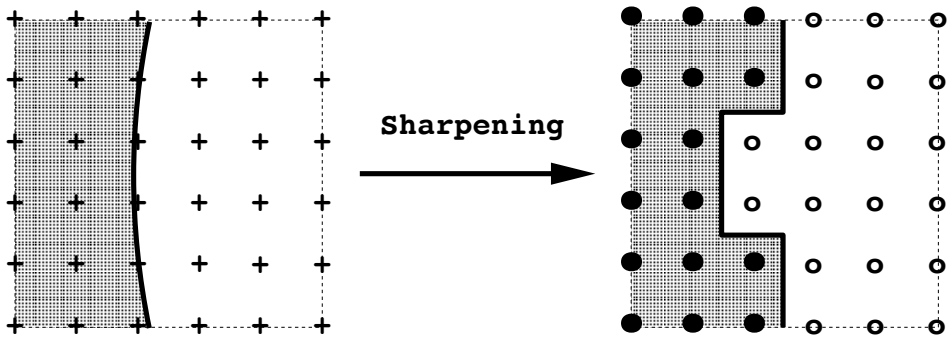


Figure 4: Sharpening a perturbed shape.

3.1 Discretization of the Heat Equation

As we have seen in the previous section, carrying out diffusion-generated motion by mean curvature requires us to solve the heat equation

$$\begin{aligned} u_t &= \Delta u, \\ \frac{\partial u}{\partial n} &= 0 \text{ on } \partial\mathcal{D} \end{aligned} \tag{3}$$

repeatedly over time intervals of (possibly variable) length Δt, starting from the characteristic function of the region to be followed. Over any of these time intervals, u may be approximated by the Fourier cosine tensor product,

$$U(x,y,t) = \sum_{i,j=0}^{n-1} c_{ij} \exp(-\pi^2[i^2+j^2][t-t_{start}]) \cos(\pi i x) \cos(\pi j y) \tag{4}$$

for $t_{start} \le t \le t_{start} + \Delta t$, where t_{start} is the time when the current interval starts.

One might expect that a Fourier spectral approximation for u would be unwise because u is initially discontinuous at interfaces. We are only interested in the solution after a time Δt, however. After a sufficiently large time Δt, high frequency modes have dissipated. Since the problem is linear, different modes do not interact and thus there is never a need to approximate high frequency modes (not even near t_{start}, when high frequency modes make an important contribution to the solution). For this reason, an accurate approximation to (3) at time Δt can be obtained using far fewer basis functions than might otherwise might be expected. Indeed, to approximate the position of the front to within a distance $\mathcal{O}(\epsilon)$, our implementations simply select an n satisfying

$$n \ge \sqrt{\frac{|\ln(\epsilon)|}{\pi^2 \Delta t}} \tag{5}$$

and verify the corresponding results by repeating the calculation with a different n (see also [18]).

3.2 Calculation of the Fourier Coefficients

The values of the Fourier coefficients, c_{ij}, of equation (4) must still be determined at the beginning of each time step (i.e., immediately following the

sharpening of the previous step). In fact, we calculate these coefficients as part of the sharpening step using an adaptive quadrature method rather than a pseudospectral method. Begin by defining

$$R_t = \{(x,y) : U(x,y,t) > \frac{1}{2}\}$$

to be the approximation of the phase we are following. By multiplying equation (4) at time $t = t_{start}$ by $\cos(\pi i x)\cos(\pi j y)$, integrating over the domain and simplifying via the usual orthogonality conditions we find

$$\begin{array}{rcll} c_{00} & = & \int_0^1 \int_0^1 U(x,y,t_{start})\ dx\ dy, & \\ c_{i0} & = & 2\int_0^1 \int_0^1 U(x,y,t_{start})\cos(\pi i x)\ dx\ dy & \text{for } i \neq 0, \\ c_{0j} & = & 2\int_0^1 \int_0^1 U(x,y,t_{start})\cos(\pi j y)\ dx\ dy & \text{for } j \neq 0, \\ c_{ij} & = & 4\int_0^1 \int_0^1 U(x,y,t_{start})\cos(\pi i x)\cos(\pi j y)\ dx\ dy & \text{for } i,j \neq 0. \end{array}$$

Immediately after sharpening,

$$U(x,y,t) = \begin{cases} 1 \text{ if } (x,y) \in R_t \\ 0 \text{ otherwise} \end{cases}$$

which implies that

$$c_{ij} = \gamma_{ij} \int\int_{R_t} \cos(\pi i x)\cos(\pi j y)\ dA \tag{6}$$

where

$$\gamma_{ij} = \begin{cases} 1 & \text{if } i = j = 0 \\ 4 & \text{if } i \neq 0 \text{ and } j \neq 0 \\ 2 & \text{otherwise} \end{cases} \tag{7}$$

Thus, simple functions must be integrated over a complicated, non-rectangular region, R_t. This may be accomplished by recursively subdividing the domain (cf. [21, 20]), as we illustrate for the region, R, given in Figure 5a.

We begin by evaluating U at the corners of a number of equally-sized subregions, so as to capture the large-scale features of the shape. Typically, $n \times n$ subregions are selected because the corresponding U-values can be evaluated in just $\mathcal{O}(n^2 \log(n))$ operations using a fast Fourier transform (see, e.g., [6]). If the phase at all four corners of any subregion corresponds to

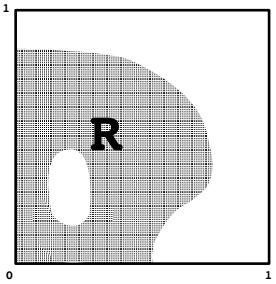


Fig. 5a. *Initial Region, R*

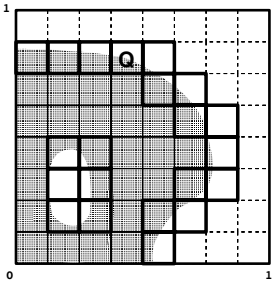


Fig. 5b. *Coarsest Subdivisions*

Figure 5: Subdividing the domain into its coarsest subregions.

white, then we assume that the subregion does not intersect with R and hence no contribution to the Fourier coefficients is made. This case corresponds to the subregions of Figure 5b which have at least one dashed edge. If all four corners of a subregion, $\tilde{Q}$, correspond to grey, however, we assume that $\tilde{Q} \subset R$ and add a contribution

$$\gamma_{ij} \int\int_{\tilde{Q}} \cos(\pi i x) \cos(\pi j y)\, dA$$

to each of the Fourier coefficients, c_{ij}, for $0 \leq i, j \leq n-1$. This case corresponds to the subregions of Figure 5b which have at least one thin, solid edge. Finally, if two phases occur, further subdivisions are carried out. We demonstrate this subdivision procedure for the subregion, Q, of Figure 5b.

Because Q is a mixed region, we divide it into quadrants, as shown in Figure 6b. Since the phase color at all corner points of quadrant Q_1^1 is white, we assume that this quadrant does not intersect with R and hence does not contribute to the Fourier coefficients. For each of the remaining quadrants, Q_1^2, Q_1^3 and Q_1^4, two phases occur, so further subdivision is required. See Figure 6c.

Focusing on the refinement of the subregion, Q_1^3, we find that the phase of the upper right hand corner of Q_2^1 is different than that of the other corners. Thus, Q_2^1 is also subdivided. Corner points of the remaining subregions are

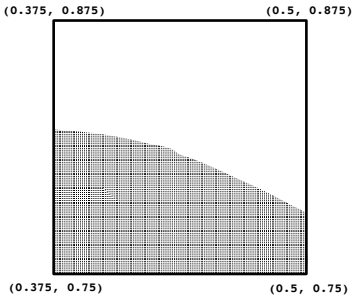


Fig. 6a. *Initial Subregion*

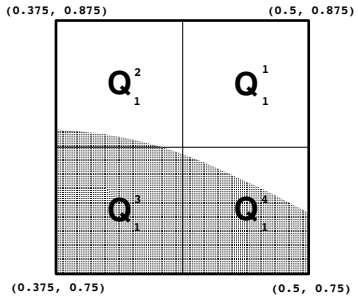


Fig. 6b. *One Subdivision*

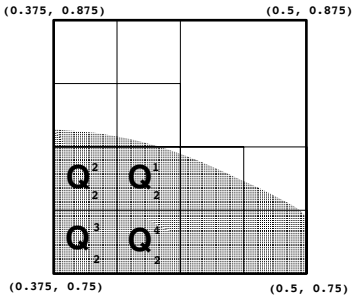


Fig. 6c. *Two Subdivisions*

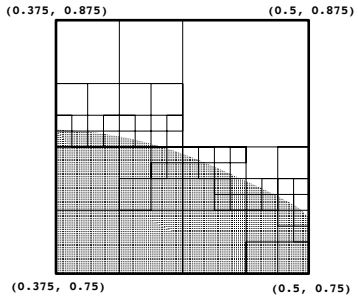


Fig. 6d. *Four Subdivisions*

Figure 6: Dividing a subregion.

grey, so we assume $Q_2^k \subset R$ for $k = 2, 3, 4$ and add contributions

$$\gamma_{ij} \int\int_{Q_2^k} \cos(\pi i x) \cos(\pi j y)\, dA$$

to each of the Fourier coefficients, c_{ij}, for $0 \leq i, j \leq n-1$. Recursive subdivisions of the domain continue (see, e.g., Figure 6d) until regions containing multiple phases can be safely approximated by some simple numerical technique. The next section discusses methods for approximating the regions at the finest grid subdivisions.

3.3 Approximation of the Finest Subregions

In the previous section, a method was introduced for recursively dividing the domain into rectangles. At some point, however, we must stop subdividing and treat the finest cells. This section discusses how to approximate the contributions to the Fourier coefficients at the finest grid subdivisions.

3.3.1 Piecewise Linear Approximation for Two-Phase Problems

To produce an $\mathcal{O}\left(\frac{h^3}{\Delta t} + h^2\right)$ approximation of the interface, a simplicial decomposition of the region, R, with a piecewise linear approximation to the boundary can be used. We now describe such a method for two phase problems in two and three dimensions.

Two Dimensional Problems

There are three main steps for approximating the integrals (6) over the finest grid subdivisions for two phase problems in two dimensions. These are detailed below.

Step 1. Divide the Square Cell into Two Triangles.
To simplify the implementation of Step 2, we begin by breaking the square subdomain into two triangles and consider each separately.

Step 2. Approximate Regions Using Triangles.
We next approximate the desired phase with a number of triangular subregions. Details for this approximation method are now given for each of the four possible cases.

Case 0. If none of the corners of the triangle belong to R, then we assume that R and the triangular subdomain do not overlap. No contribution to the Fourier coefficients is made.

Case 1. If one corner is in R, then linear interpolation is used to determine a triangular approximation to the subregion. For example, consider approximating the gray phase in $\triangle ABC$ of Figure 7. Letting $U(A), U(B)$ and $U(C)$ be the function values at points A, B

and C, we approximate where the curve crosses the edges of the triangle by points a and b,

$$\begin{aligned} a &= B + \frac{\frac{1}{2} - U(B)}{U(C) - U(B)}(C - B), \\ b &= A + \frac{\frac{1}{2} - U(A)}{U(C) - U(A)}(C - A). \end{aligned} \tag{8}$$

This gives a triangular approximation, $\triangle baC$, to the desired subregion which can be used to approximate the contributions to the Fourier coefficients (see Step 3 below).

Case 2. If two corners are in R, then we represent the shape as the difference of shapes which are treated using Cases 1 and 3. See Figure 8.

Case 3. If three corners are in R, then we assume that the entire subdomain belongs to R, and we approximate the integrals (6) over the entire subdomain.

We seek an estimate of the error produced by this step for a smooth curve. One source of error occurs when smooth curves are approximated by line segments. By Figure 9, this approximation produces an $\mathcal{O}(h^2)$ error in the position of the front, since the curvature is independent of h.

We also produce errors by replacing the actual front position with the interpolation (8). In this case we expect an $\mathcal{O}\left(\frac{h^3}{\Delta t}\right)$ error, based on the one dimensional studies given in [18].

Taking into account both of the contributions to the error, we find that this triangular approximation of regions produces an $\mathcal{O}\left(\frac{h^3}{\Delta t} + h^2\right)$ error in the position of the front.

Step 3. Integrate over each Triangular Subregion.
We are now left with the task of adding a contribution

$$I_{ijk} = \gamma_{ij} \int\int_{T_k} \cos(\pi i x) \cos(\pi j y)\, dA \tag{9}$$

to each Fourier coefficient, c_{ij}, for each triangular subregion, T_k.

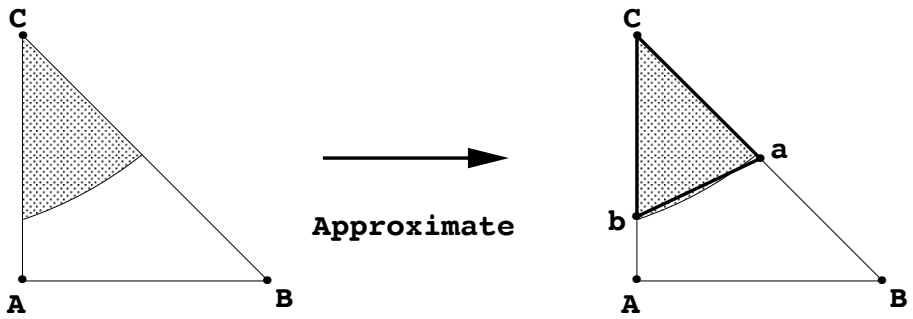


Figure 7: A shape represented by a triangle.

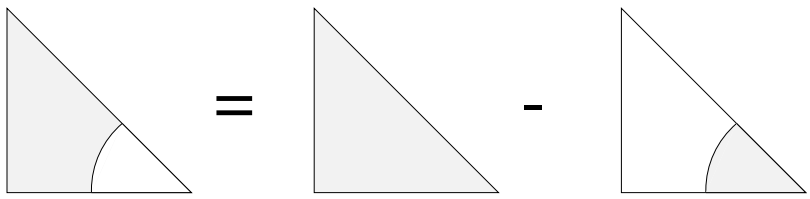


Figure 8: A shape represented by a difference of triangles.

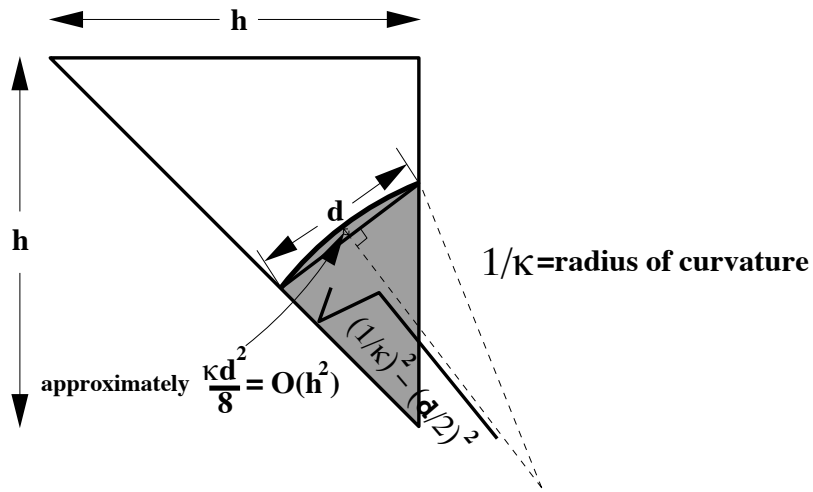


Figure 9: Errors approximating curved segments.

Expanding the integrand about the centroid, $(\tilde{x}, \tilde{y})$, of T_k yields

$$I_{ijk} = \gamma_{ij} \text{Area}(T_k) \cos(\pi i \tilde{x}) \cos(\pi j \tilde{y}) + \mathcal{O}([i^2 + j^2]h^4) \qquad (10)$$

where $\text{Area}(T_k)$ is the area of triangle T_k. This approximation is preferred over the direct evaluation of the integrals (9) because it is much faster (it only requires two trigonometric evaluations) and it produces errors which are typically small relative to those arising in Step 2.

Three Dimensional Problems

There are also three steps for approximating the contributions to the Fourier coefficients over the finest grid subdivisions for two phase problems in three dimensions. These are outlined below.

Step 1. Divide the Cube into Six Tetrahedrons.
To simplify the implementation of Step 2, we begin by breaking cube-shaped subdomains into six tetrahedrons and consider each separately.

Step 2. Approximate Regions Using Tetrahedrons.
We next approximate the desired phase with a number of tetrahedrons.

For smooth surfaces[1] this step produces an $\mathcal{O}\left(\frac{h^3}{\Delta t} + h^2\right)$ error in the position of the front where h is the width of the finest grid subdivision. An outline of this approximation method is now given for each of the five possible cases.

Case 0. If none of the corners of the tetrahedron belong to R, no contribution to the Fourier coefficients is made.

Case 1. If one corner is in R, then we linearly interpolate to estimate the location of the interface and approximate the region by a tetrahedron.

Case 2. If two corners are in R, then we estimate the location of the interface and break the subregion into three tetrahedrons. For example, the shaded region of T_{ABCD} in Figure 10 is approximated by the tetrahedrons T_{Begf}, T_{Bfgh} and T_{BhDg}.

Case 3. If three corners are in R, then we represent the shape as the difference of shapes which are treated using Cases 1 and 4.

Case 4. If four corners are in R, then we assume that the entire subdomain belongs to R, and we approximate the integrals (11) over the entire subdomain.

Step 3. Integrate over each Tetrahedron.
For each tetrahedron, T_ℓ, a contribution

$$I_{ijk\ell} \;=\; 2^p \int\int\int_{T_\ell} \cos(\pi i x)\cos(\pi j y)\cos(\pi k z)\; dV \qquad (11)$$

must be added to each Fourier coefficient, c_{ijk}, where p is the number of nonzero elements of $\{i, j, k\}$.

Expanding the integrand about the centroid, $(\tilde{x}, \tilde{y}, \tilde{z})$, of T_ℓ yields

$$I_{ijk\ell} \approx \gamma_{ijk}\text{Volume}(T_\ell)\cos(\pi i\tilde{x})\cos(\pi j\tilde{y})\cos(\pi k\tilde{z})$$

[1]In three dimensions, nonsmooth corners may arise from singularities in the solution. Each corner can produce an $\mathcal{O}(h^3)$ error in the phase areas. However, because these corners are rapidly smoothed away, they typically do not affect the overall order of accuracy of the method.

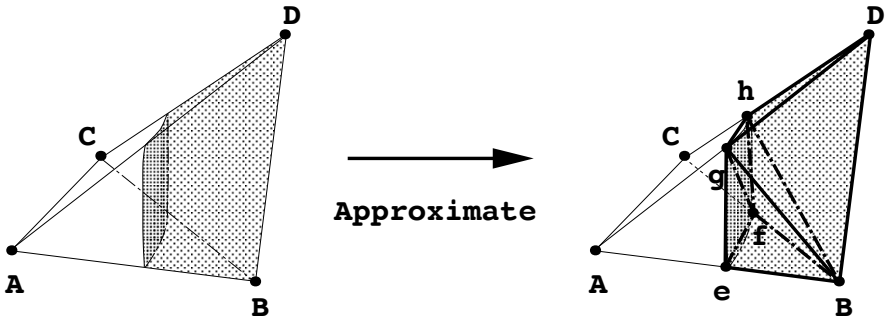


Figure 10: A shape approximated by tetrahedrons.

where Volume(T_ℓ) is the volume of tetrahedron, T_ℓ. Similar to the two dimensional case, this approximation is preferred over the direct evaluation of the integrals (11) because it is much faster and it produces errors which are typically small relative to those arising in Step 2.

3.3.2 Approximation of Junctions

A number of methods for accommodating junctions are available [18]. A particularly simple and accurate approach is to recursively subdivide any region containing more than two phases. After only a few iterations, the smallest subregions that arise can be trivially treated by assigning an equal contribution to each set of Fourier coefficients.

3.4 Refinement Techniques

In Section 3.2, a recursive algorithm for subdividing the domain was introduced. We now carry out a more detailed study of the method and introduce a gradual refinement which overcomes certain limitations of the original algorithm.

For illustrative purposes, all examples set the width of the coarsest grid to be $H = \frac{1}{8}$. Similar results arise for the usual choice of $H = \frac{1}{n}$.

3.4.1 The Original Refinement Algorithm

The original refinement algorithm of Section 3.2 is effective for a variety of problems. For certain smooth regions, however, small slivers of a region can be missed. Consider, for example, the shape found in Figure 11. Applying the subdivision algorithm gives the mesh displayed in Figure 12a. A close examination of the leftmost part of the shape indicates that a small, thin region is missed by the algorithm.

The original refinement algorithm also produces errors when applied to nonsmooth shapes. Consider, for example, the region displayed in Figure 13. Such a shape may arise when a topological breaking occurs. Applying the original subdivision algorithm to the shape gives the mesh displayed in Figure 14a. Clearly, an $\mathcal{O}(H^2)$ error in the phase area is produced at the cell containing the sharp corners. This corresponds to an $\mathcal{O}(\Delta t)$ error when $H = \frac{1}{n}$ and n is chosen according to (5).

Although the errors produced by these flaws in the refinement technique often are less than those arising from the MBO-method, we prefer a more accurate refinement to achieve a greater confidence in our results. Furthermore, a more accurate refinement is required whenever higher order, extrapolated methods are used (see [17, 18]).

3.4.2 A Method for a Gradual Refinement

As we have seen in the previous subsection, the original subdivision algorithm can miss small pieces of both smooth and nonsmooth shapes. We seek a refinement which captures the entire interface at the level of the finest grid subdivision, even for nonsmooth shapes.

To achieve this objective, a gradual refinement was implemented. This method proceeds according to the original subdivision algorithm of Section 3.2, with the following additional consideration:

> Whenever any cell is refined, check the subdivision level of the neighboring cells. Subdivide neighbors which are two or more

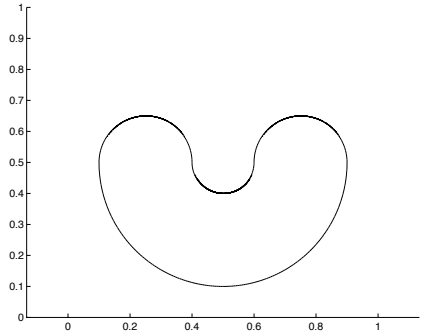


Figure 11: A smooth region.

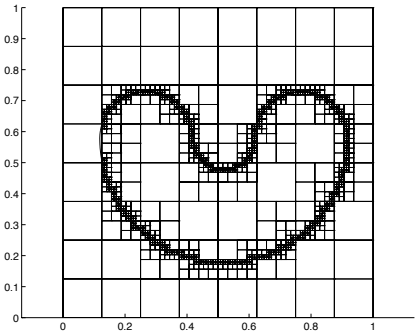


Fig. 12a. *Original Refinement*

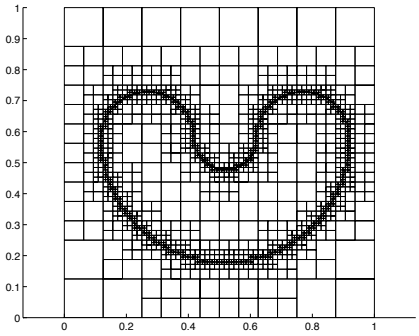


Fig. 12b. *Gradual Refinement*

Figure 12: Refinement methods for smooth regions.

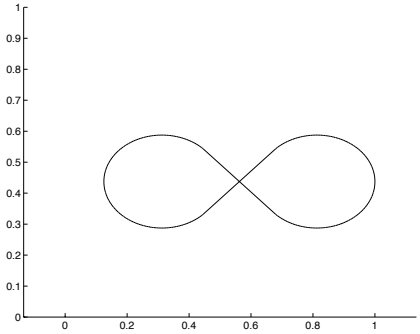


Figure 13: A problem with sharp corners.

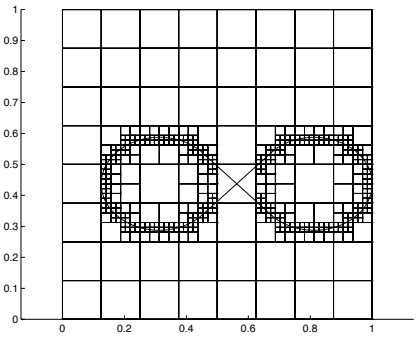


Fig. 14a. *Original Refinement*

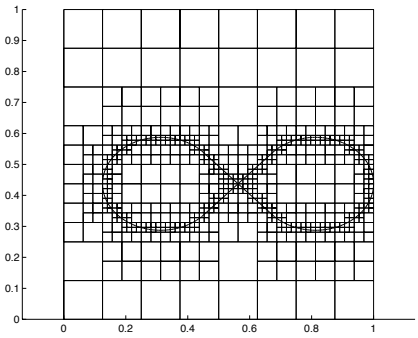


Fig. 14b. *Gradual Refinement*

Figure 14: Refinement methods for corners.

levels of refinement coarser.

This method accurately represents the narrow, sliver-shaped regions that were missed using the original refinement. By using a fine subdivision in a small neighborhood of the interface, this method even captures the rapid variations in the front that arise from corners. See Figures 12b and 14b for examples.

Certainly, this gradual refinement produces more cells than the original approach. The order of the number of cells is unchanged, however. To see this, note that cells of width

$$h_\ell = 2^{-\ell} H, \text{ where } 0 < \ell \leq \log_2 \left(\frac{H}{h} \right)$$

form a band at most two cells wide on each side of the interface. The length of each band can be bounded by a constant, K, independent of h (e.g., bands for a convex region are shorter than the perimeter of the domain). Letting $n_{\tilde{h}}$ be the number of cells of width $\tilde{h}$, we observe that

$$\begin{aligned} \text{Total number of cells} &= n_h + n_{2h} + \ldots + n_{\frac{H}{2}} + n_H, \\ &< \frac{4K}{h} + \frac{4K}{2h} + \ldots + \frac{4K}{\frac{H}{2}} + n^2, \\ &< \frac{8K}{h} + n^2. \end{aligned}$$

Thus, $\mathcal{O}(\frac{1}{h} + n^2)$ cells are required, which matches in order the result for the original refinement. Implementation of this gradual refinement is somewhat more involved than the original approach because cell neighbors must be found. Many data structures appropriate for this task have been considered [21, 20]. Our implementations define the grid as a list of vertices (cf. [9]), and access the cells and their neighbors indirectly by traversing their vertices [18].

3.5 Fast, Transform-Based Algorithms

The refinements of the previous sections lead to a large number of function evaluations,

$$U(x,y) = \sum_{j,j'=0}^{n-1} c_{jj'} \exp(-\pi^2[j^2 + (j')^2]\Delta t)\cos(\pi j x)\cos(\pi j' y). \tag{12}$$

Because these evaluations occur on an unequally spaced grid, a fast Fourier transform cannot be used. Direct evaluation of Eq. (12) at N_q points, however, is often prohibitively expensive because $\mathcal{O}(n^2 N_q)$ operations are required. Similarly, evaluation of the Fourier coefficients using Eqs. (6) and (10) leads to a Fourier sum of the form

$$c_{jj'} = \sum_{\ell=0}^{N_p-1} d_\ell \cos(\pi j x_\ell)\cos(\pi j' y_\ell) \tag{13}$$

where $0 \le j, j' \le n-1$ and (x_ℓ, y_ℓ) are unequally spaced. Once again, a fast Fourier transform cannot be used, and direct evaluation leads to $\mathcal{O}(n^2 N_p)$ operations.

Several methods for the fast evaluation of Eqs. (12) and (13) have been developed [3, 7, 23]. In [3], for example, an efficient and practical method based on multiresolution analysis was developed that evaluates Eq. (12) at N_q points in

$$\mathcal{O}\left(N_q \log^2\left(\frac{1}{\epsilon}\right) + n^2 \log(n)\right) \tag{14}$$

operations, and evaluates all n^2 Fourier coefficients of (13) in

$$\mathcal{O}\left(N_p \log^2\left(\frac{1}{\epsilon}\right) + n^2 \log(n)\right) \tag{15}$$

operations, where ϵ is the precision of the computation.

Using the fact that $\mathcal{O}\left(\frac{1}{h}\right)$ refined cells arise (see Section 3.4.2), it is clear that $N_q = \mathcal{O}\left(\frac{1}{h}\right)$ and $N_p = \mathcal{O}\left(\frac{1}{h}\right)$. The remaining $\mathcal{O}(n^2)$ coarse grid cells may be treated with a fast Fourier transform in $\mathcal{O}(n^2 \log(n))$ operations. Applying these relationships, along with $h \ll \frac{1}{n}$, we see that a total of

$$\mathcal{O}\left(\left(\frac{1}{h}\right)\log^2(h) + n^2 \log(n)\right)$$

operations arise at each iteration of the spectral discretization of the MBO-method. As was shown in [18], the MBO-method produces an $\mathcal{O}(\Delta t^2)$ error in the position of a smooth curve at each step of the method when two phase regions occur. To avoid degrading this accuracy (see Section 3.3.1), we select $h = \mathcal{O}(\Delta t)$ to arrive at

$$\mathcal{O}\left(\frac{1}{\Delta t}\log^2(\Delta t)\right)$$

operations per step. For the case of junctions, we may apply the same considerations to determine that

$$\mathcal{O}\left(\frac{1}{\Delta t}\log(\Delta t)\right)$$

operations are required per step to avoid degrading the overall accuracy of the method [18].

4 Comparison to the Usual Finite Difference Discretization

There are several reasons why the spectral method described in this article is preferred over the usual finite difference approach. These reasons are outlined below.

1. As has been discussed in Section 3.1, only low frequency modes need to be approximated provided Δt is not taken very small. A large amount of computational work is saved by only treating these low frequency modes.

2. The new, spectral method does not require any time-stepping between t_{start} and $t_{start} + \Delta t$. This eliminates a possible source of error and produces large savings in computational work.

3. Local refinement is much simpler to implement for the new, spectral approach because it is done in the context of a quadrature, rather than a discretization of a differential equation.

4. By using a spectral method, the error arising from discretizing the heat equation can be nearly eliminated. This is an attractive feature, because it makes extrapolation in Δt practical (see [18, 17]), which in turn allows for larger time steps. When larger time steps are taken, even fewer basis functions are required to solve the heat equation to a given accuracy.

5. The original finite difference algorithm must satisfy (2) globally, or part of the front may erroneously remain stationary. By recursively refining near the interface, the new, spectral approach can essentially eliminate this restriction.

6. The new, spectral method also gives an $\mathcal{O}\left(\frac{h^3}{\Delta t} + h^2\right)$ approximation of the location of the front, which is greatly superior to the first order approximation arising for finite differences. As we saw in the previous section, this improved accuracy, in part, explains why

$$\mathcal{O}\left(\frac{1}{\Delta t}\log^2(\Delta t)\right)$$

operations are needed per step for the basic method. This compares very favorably to the idealized finite difference result for smooth curves, $\mathcal{O}(\frac{1}{\Delta t^3})$, which was derived in Section 2.4.

These are indeed formidable advantages for the new, spectral method over the usual finite difference approach. To illustrate the performance improvement, consider the motion by mean curvature of the kidney-shaped region displayed in Figure 15. Using the new, spectral method and an optimized finite difference approach[2], we compare the area lost over a time $t = 0.0125$ with the exact answer, $0.0125 \times 2\pi = 0.0785398$ (see [16]). From Table I, we see that the new, spectral method is adequate for finding solutions to within a 1% error. The finite difference approach, however, becomes impractical when accurate solutions are sought (see Table II).

Numerical tests for the problems described in the next section also found that the new, spectral method often requires less than 0.1% of the computational time of the usual finite difference approach. For this reason, the

[2]The difference algorithm uses an adaptive time stepping method on a uniform mesh. A multigrid technique was used to solve the implicit equations which arose from a backward Euler time-stepping scheme. See [18].

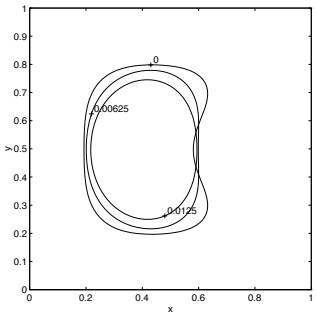


Figure 15: A smooth interface at time, t.

numerical studies in the following section are carried out using the new, spectral method.

Δt	h	*Error*	*Time*[3]
0.003125	2^{-9}	4%	0.4 s
0.00078125	2^{-11}	1%	8 s

Table I. New, spectral method

Δx	*Error*	*Time*
$\frac{1}{128}$	4%	85 s
$\frac{1}{512}$	3%	10341 s

Table II. Finite difference discretization

5 Numerical Experiments

In this section we report on various experiments using our algorithm. For further quantitative studies for both mean curvature flow and other, more general motions, see [18, 19].

[3]All timings were carried out on an HP735/100 workstation.

5.1 A Smooth Three Phase Problem

To begin, consider the motion by mean curvature of the three phase problem given in Figure 16. Using our new spectral method, the change in the area of the central region was compared to the exact result[4] for several Δt. Because an $\mathcal{O}(\sqrt{\Delta t})$ error seems plausible from the asymptotic results given in [18, 19], an extrapolation in the area, $\frac{1}{\sqrt{2}-1}\left(\sqrt{2}A^{\Delta t} - A^{2\Delta t}\right)$, was also computed to eliminate the conjectured leading order error term. The results for a number of experiments are given in Table III, below.

Δt	*Error in* $A^{\Delta t}$	*Conv. Rate*	*Error in Extrapolation*	*Conv. Rate*
0.00625	2.60e-03	0.67	-1.08e-03	1.33
0.003125	1.70e-03	0.61	-4.63e-04	1.22
0.0015625	1.14e-03	0.57	-2.12e-04	1.13
0.00078125	7.79e-04	0.55	-1.00e-04	1.08
0.000390625	5.37e-04	0.54	-4.86e-05	1.05

Table III. Results for a smooth three phase problem.

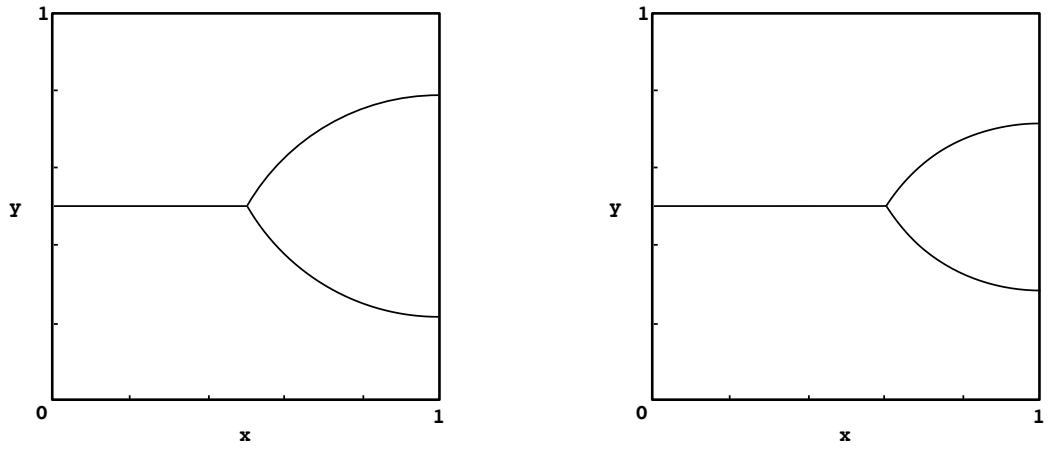
These results support the conjecture that the MBO-method is $\mathcal{O}(\sqrt{\Delta t})$ for the case of junctions and suggest that extrapolation can be used in conjunction with the new, spectral method to produce higher order estimates of certain quantities of interest such as phase areas.

5.2 The Evolution of a Junction Through a Singularity

The evolution of more complicated problems may also be simulated using our new, spectral discretization. Consider, for example, the motion by mean curvature of the three phase problem given in Figure 17. Using our new, spectral discretization of the MBO-method, estimates of the disappearance

[4] Applying the Von Neumann-Mullins parabolic law [16], gives us that the area of the central phase obeys

$$\frac{\mathrm{d}A}{\mathrm{d}t} = \frac{1}{2} \times \frac{\pi}{3}(2-6).$$



$t = 0.0$ $t = 0.04$

Figure 16: A smooth three phase problem.

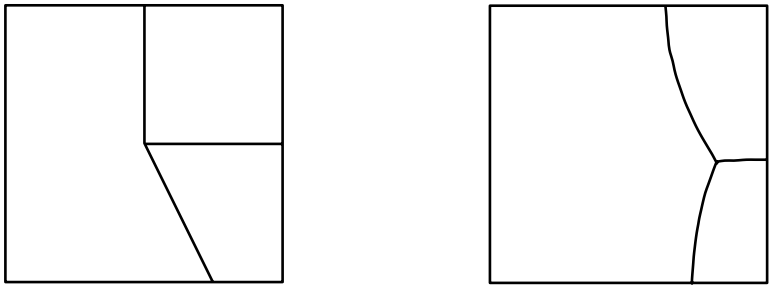
time, $T^{\Delta t}$, of the smallest phase were compared to the exact answer[5] for several Δt. The results for a number of experiments are reported in Table IV, below.

Δt	*Error in* $T^{\Delta t}$	*Conv. Rate*
0.0025	0.04606	0.46
0.00125	0.03338	0.46
0.000625	0.02421	0.46
0.0003125	0.01754	0.46

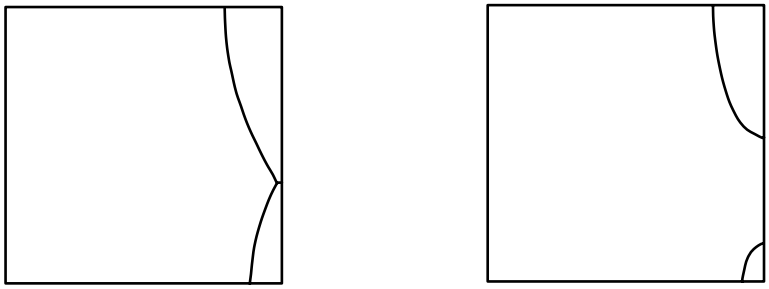
Table IV. Results for the disappearance time of a phase region.

These results are suggestive of an approximately $\mathcal{O}(\sqrt{\Delta t})$ error for the basic method.

[5]This result, $T = 0.33051$, was obtained using Brian Wetton's front tracking code; see [4].

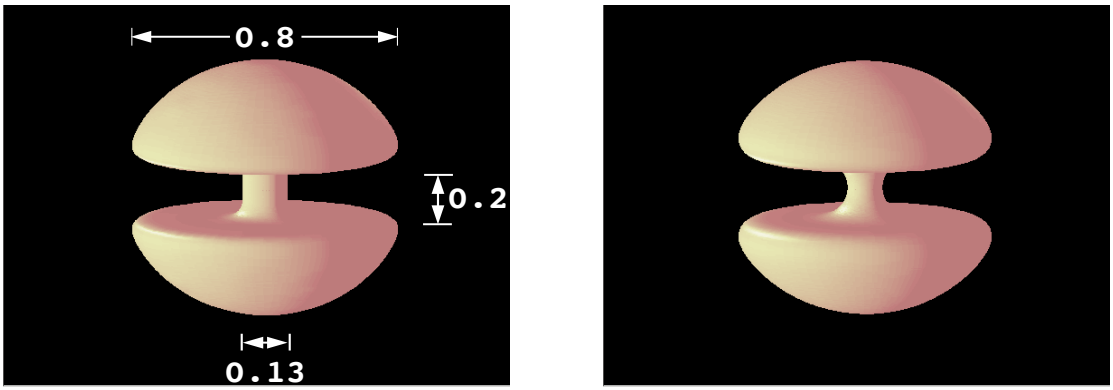


$t = 0.000$ $t = 0.150$



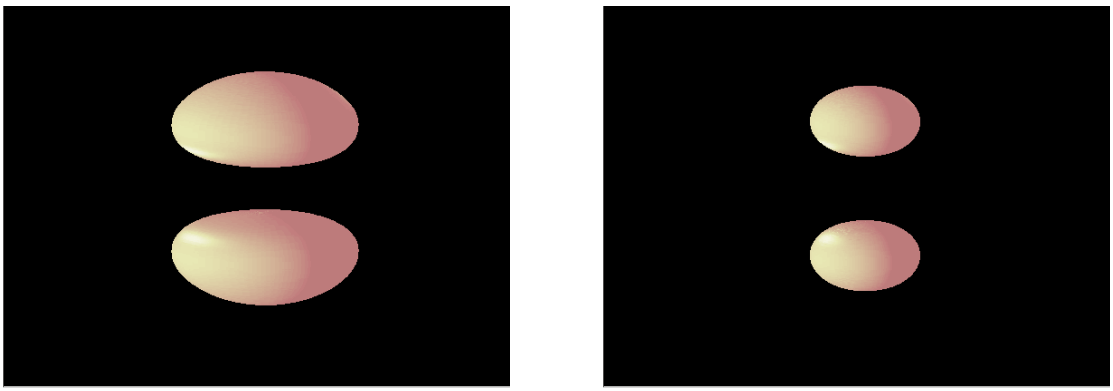
$t = 0.300$ $t = 0.325$

Figure 17: The evolution of a junction through a singularity.



$t = 0.0000$ $t = 0.0008$

$t = 0.0020$ $t = 0.0032$



$t = 0.0064$ $t = 0.0128$

Figure 18: A thin-stemmed barbell moving by mean curvature motion.

5.3 A Three Dimensional Example

Interesting examples in three dimensions are also naturally handled by the method. For example, Figure 18 displays the motion of a thin-stemmed barbell using a step size, $\Delta t = 0.0004$. From these images, it is clear that the center handle pinches off to form two pieces. As expected from [12], these convex shapes become nearly spherical as they disappear. Note, however, that a wider stem can produce a qualitatively different motion. For example, [18] gives the motion of a thick-stemmed barbell which exhibits no topological shape changes, and eventually becomes ellipsoidal and more spherical as it disappears.

5.4 A Multiple Phase Example in Three Dimensions

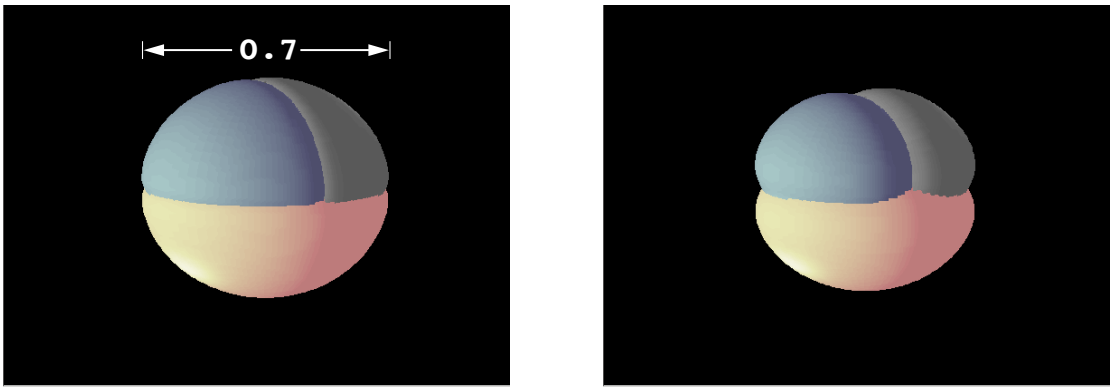
The evolution of multiple phase junctions may also be studied using our new, spectral discretization of the MBO-method. For example, Figure 19 displays the motion of a spherical four phase shape using a step size, $\Delta t = 0.0004$. From these images, we see that the four phase junction is stable under mean curvature motion, as is expected from experimental studies of recrystallized metal [11].

6 Acknowledgments

Many thanks to Uri Ascher and Brian Wetton for helpful discussions. I also thank Leslie Greengard for suggesting the use of the unequally spaced fast Fourier transform.

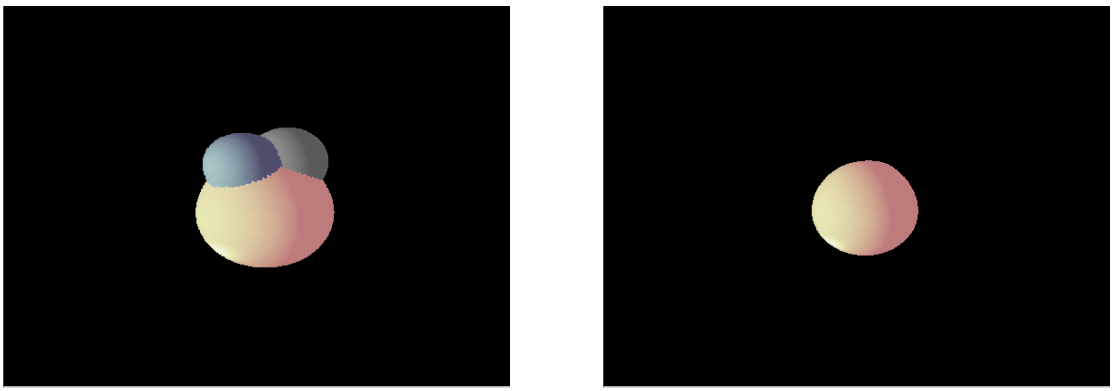
References

[1] D. Adalsteinsson and J.A. Sethian. A fast level set method for propagating interfaces. *J. Comput. Phys.*, 118(2):269–277, 1995.



$t = 0.0000$ $t = 0.0036$

$t = 0.0072$ $t = 0.0108$



$t = 0.0144$ $t = 0.0180$

Figure 19: A four phase example moving by mean curvature motion.

[2] 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.

[3] G. Beylkin. On the fast Fourier transform of functions with singularities. *Applied and Computational Harmonic Analysis*, 2:363–381, 1995.

[4] L. Bronsard and B.T.R. Wetton. A numerical method for tracking curve networks moving with curvature motion. *J. Comput. Phys.*, 120(1):66–87, 1993.

[5] 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.

[6] C. Canuto, M.Y. Hussaini, A. Quarteroni, and T.A. Zang. *Spectral Methods in Fluid Dynamics*. Springer-Verlag, 1987.

[7] A. Dutt and V. Rokhlin. Fast Fourier transforms for nonequispaced data. *SIAM Journal on Scientific and Statistical Computing*, 14(6):1368–1393, 1993.

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

[9] R.E. Ewing. Adaptive mesh refinements in large-scale fluid flow simulation. In *International Conference on Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, pages 299–314, Lisbon, Portugal, 1984.

[10] 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.

[11] D. Harker and E.R. Parker. Grain shape and grain growth. In *Transactions of the A.S.M*, pages 156–161, Cleveland, 1945.

[12] G. Huisken. Flow by mean curvature of convex surfaces into spheres. *J. Differential Geometry*, 20:237–266, 1984.

[13] 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.

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

[15] B. Milne. *Adaptive Level Set Methods Interfaces*. PhD thesis, University of California, Berkeley, 1995.

[16] W.W. Mullins. Two-dimensional motion of idealized grain boundaries. *Journal of Applied Physics*, 27(8):900–904, 1956.

[17] 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.

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

[19] S.J. Ruuth. A diffusion-generated approach to multiphase motion. CAM Report 97-28, University of California, Los Angeles, 1997.

[20] H. Samet. *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley, 1990.

[21] H. Samet. *The Design and Analysis of Spatial Data Structures*. Addison-Wesley, 1990.

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

[23] E. Sorets. Fast Fourier transforms of piecewise constant functions. *J. Comput. Phys.*, 116:369–379, 1995.

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

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