JOURNAL OF COMPUTATIONAL PHYSICS 120, 66–87 (1995)

A Numerical Method for Tracking Curve Networks Moving with Curvature Motion

LIA BRONSARD*

Department of Mathematics, McMaster University, Hamilton, Ontario, Canada L8S4K1

AND

BRIAN T. R. WETTON†

Department of Mathematics, University of British Columbia, Vancouver, BC, Canada V6T1Z2

Received July 25, 1994

A finite difference method is proposed to track curves whose normal velocity is given by their curvature and which meet at different types of junctions. The prototypical example is that of phase interfaces that meet at prescribed angles, although eutectic junctions and interactions through nonlocal effects are also considered. The method is based on a direct discretization of the underlying parabolic problem and boundary conditions. A linear stability analysis is presented for our scheme as well as computational studies that confirm the second order convergence to smooth solutions. After a singularity in the curve network where the solution is no longer smooth, we demonstrate "almost" second-order convergence. A numerical study of singularity types is done for the case of networks that meet at prescribed angles at triple junctions. Finally, different discretizations and methods for implicit time stepping are presented and compared.

1. INTRODUCTION

In this paper we present numerical simulations for the motion of phase boundaries in two-dimensional physical systems with several phases, such as grain growth in crystalline structures, lamellar eutectic growth and volume preserving mean curvature flow for binary mixtures. Our simulation is based on the discretization of a system of nonlinear partial differential equations which models the evolution of each interface. In this model, the principal contribution to the normal velocity of an interface (or curve) is given by its curvature. Moreover, we allow the curves to meet at several types of junctions. The first considered, our main example, allows prescribed angles at triple junctions where three curves meet and at the boundary of the domain. This model is used by material scientists in the study of the evolution of idealized grain growth [18]. In this case the triple junction angles are all 120° (for other angles see, e.g., [3, 22]). Junctions modeling those found in lamellar eutectic growth are also considered.

The nonlinear system that we use is the formal asymptotic limit of a vector-valued Ginzburg–Landau equation with a triple-well potential introduced in [1] to model physical systems with more than two phases. In the limit, the curves representing phase boundaries evolve by their curvature with prescribed angles at the triple junctions which depend on the potential. For a symmetric potential, the angles are 120° and this corresponds to the model of grain growth. In the case of three curves meeting at one point, the problem is given mathematically by a system of 6×6 parabolic partial differential equations for which short-time existence was proved [1] (the problem with a junction of more than three curves and given angle conditions is ill-posed). Their result extends to networks of more than three curves connected by multiple triple junctions. In this paper, we use a second-order finite difference method to compute approximations to this parabolic system describing the evolution of the curves. The discretization is based on a staggered grid using straight forward second-order discretization at the interior grid points. In order to handle the boundary conditions we use an extrapolation of the interior grid points up to second order in the actual boundary conditions. We show second-order convergence to smooth solutions in computational studies and perform a linear stability analysis for the junction discretization using a novel energy technique. This last technical discussion is delayed to an appendix.

The short-time existence result of Bronsard–Reitich shows that the above discretization is well defined until the parametrization breaks down. As can be seen numerically, this occurs when the length of one of the curves shrinks to zero, i.e., when a geometric singularity occurs. This "singularity time" can be

* E-mail: bronsard@icarus.math.mcmaster.ca. Supported by NSERC Canada and SERB (McMaster) grants.

† E-mail:wetton@math.ubc.ca. Supported by an NSERC Canada Grant.

0021-9991/95 $12.00

detected numerically and at that time we perform a "surgery" which depends on the physical system being modeled. This coarsening process can then be continued until one phase is left (we restrict ourselves to computations in convex domains). At singularity time there are derivative singularities in our parametrization of the curve network: A detailed numerical study of the performance of the method through singularity and surgery is presented, showing "almost" second order convergence.

There are several numerical methods which have been recently developed to simulate geometrical motions of curves and surfaces. We refer to [4] for a survey of many approaches to defining geometric motion of interfaces. More recently, there has been some work in which a level-set formulation is introduced to study the evolution of triple junctions [17]. There have also been several discretized models developed more specifically for the motion of grain boundaries using the Potts model (see [10] and references therein), vertex and boundary dynamics models (see [14] and references therein), mean-fields theories (see [6] and references therein) as well as the work of Ceppi and Nasello [5]. Our method differs from these approaches since it is based on a direct discretization of the evolution equation of each interface. Other authors [8, 7] have proposed similar interface "tracking" methods for this problem using heuristic arguments rather than a discretization of an underlying set of equations. We present a detailed comparison between the methods and demonstrate that our discretization at the junctions is more accurate. We also present two more efficient techniques for solving the systems arising from the implicit time discretization of curve networks with junctions. This problem is particularly suited to tracking because curves can never cross and the domains we consider are convex so curves cannot cross the domain boundaries. Tracking interfaces that can cross in nontrivial ways is in general a much harder problem (see, e.g., [20]).

We present the interface equations and discretization procedure for the system corresponding to grain growth in the next two sections. Then, we present simulations for several networks to illustrate the type of singularities that can be observed. Two physical models are considered: first one with three different phases and then the case of grain growth in an isotropic material in which each grain has its own lattice orientation and hence there are as many phases as there are grains. We concentrate on computations where the interfaces meet at the triple junction with an angle of 120° and at the boundary of the domain with normal angle. In this case, areas enclosed by curves obey the Von Neumann–Mullins parabolic law (see Section 4.3 below). Moreover, the total network length must decrease. We demonstrate that our numerical technique approximates these laws accurately. The short-time existence in [1] also applies to other angle conditions and we extend our method to this case and present some results. Following this, we present a numerical convergence study for the method through several singularities. Then, other spatial discretizations and methods for implementing implicit time stepping are presented and compared. Finally, the curve network "tracking" ideas developed for the problem of grain growth are applied to two other problems: a model of lamellar eutectic growth and a non-local area preserving network problem.

2. EQUATIONS OF INTERFACE MOTION

The case of grain growth, where interfaces move with curvature motion and meet at triple junctions with specified angles, is our first example of a curve network. For simplicity, we describe the situation in which there are three phase boundaries described by curves $x^i(\sigma, t)$ for $i = 1, 2, 3$, where $\sigma \in [0, 1]$ is a parameterization of the curve. The situation is shown in Fig. 1. The three curves evolve normally according to curvature motion. One choice that describes this motion (which is arbitrary up to any velocity in the tangential direction) is

$$x_t^i = x_{\sigma\sigma}^i / |x_\sigma^i|^2, \tag{1}$$

since the projection of the RHS of the above equation in the normal direction is equal to the curvature. At $\sigma = 0$ the curves meet the domain boundary of the problem at a normal angle. To be more precise, let $b(s)$ parametrize the boundary in terms of an arc length parameter s. Then the junction conditions at the domain boundary are given by

$$x^i(0, t) = b(s^i(t)) \tag{2}$$

for some $s^i(t)$ and

$$x_\sigma^i(0, t) \cdot b'(s^i(t)) = 0, \tag{3}$$

where the prime denotes differentiation with respect to s.

Finally, at $\sigma = 1$ the curves meet at a common point with given angles θ^i between curves i and $i + 1$, i.e.,

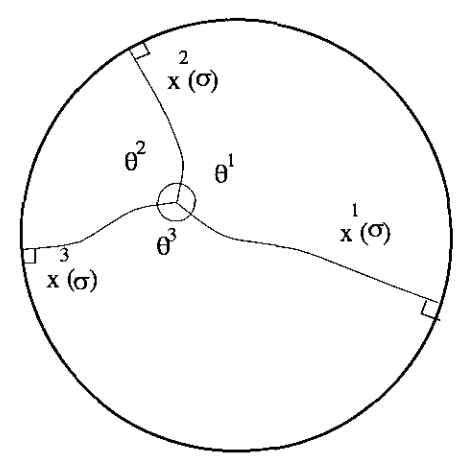

FIG. 1. A three curve network.

$$x^1(1, t) = x^2(1, t) = x^3(1, t) \tag{4}$$

and

$$\frac{x^1_\sigma(1,t)}{|x^1_\sigma(1,t)|} \cdot \frac{x^2_\sigma(1,t)}{|x^2_\sigma(1,t)|} = \cos\theta^1 \tag{5}$$

$$\frac{x^2_\sigma(1,t)}{|x^2_\sigma(1,t)|} \cdot \frac{x^3_\sigma(1,t)}{|x^3_\sigma(1,t)|} = \cos\theta^2 \tag{6}$$

We refer to (2)–(6) naturally as boundary conditions of the network evolution problem (1) although only (2) and (3) are associated with the domain boundary

3. DISCRETIZATION

In this section we present in detail the discretization procedure we use to simulate (1)–(6). Our basic approach is to use a staggered grid in σ and we shall denote the approximations by capital letters, i.e., $X^i_j(t) \approx x^i((j - \frac{1}{2})h, t)$ where h is the grid spacing and $N = 1/h$ is the number of interior grid points for $\sigma \in [0, 1]$. As the notation above suggests, we consider semi-discrete or method of lines approximations.

In order to write the discretized equations, we introduce some additional notation. Let D_k denote the second-order centered approximation of the kth derivative, i.e.,

$$D_1 X_j = (X_{j+1} - X_{j-1})/2h$$
$$D_2 X_j = (X_{j+1} - 2X_i + X_{j-1})/h^2,$$

and let D_+ and $\mathcal{F}$ denote forward differencing and forward averaging, respectively:

$$D_+ X_j = (X_{j+1} - X_j)/h$$
$$\mathcal{F} X_j = (X_{j+1} + X_j)/2.$$

For later use, we note that due to the staggered grid, $D_+ X_j$ is a second-order approximation of $x_\sigma(jh)$ and $\mathcal{F} X_j$ is a second-order approximation of $x(jh)$.

We now write down the discretization procedure. Equations (1) are approximated at all grid points using standard differences,

$$\dot{X}^i_j = \frac{D_2 X^i_j}{|D_1 X^i_j|^2}, \tag{7}$$

where the dot denotes time derivative. Formally, these discrete equations require values of X_0 and X_{N+1} outside the computational domain. We shall use the boundary conditions to extrapolate the interior values X_1 and X_N to the unknown exterior values X_0 and X_{N+1}. Next, we give the details of this procedure for the boundary conditions at the domain boundary $\sigma = 0$ and the junction $\sigma = 1$.

3.1. Domain Boundary

We obtain an expression for X_0 by formally approximating the boundary conditions (2) and (3) to second order,

$$\mathcal{F} X^i_0 = B^i, \tag{8}$$

where $B^i := b(s^i)$ is some point on the boundary and

$$D_+ X^i_0 \cdot T^i = 0, \tag{9}$$

where $T^i = b'(s^i)$. These discrete boundary conditions have a geometrical interpretation. In what follows, we omit the superscript i since the same reasoning applies to all three curves.

Equation (8) can be rewritten as

$$X_0 = 2B - X_1 \tag{10}$$

which geometrically means that X_0 is obtained by reflecting X_1 through the (as yet unknown) boundary point B. The point B is determined by substituting (10) into (9) to obtain

$$(X_1 - b(s)) \cdot b'(s) = 0.$$

Consulting Fig. 2, we see that $B = b(s)$ must be the point on the boundary of minimum distance to X_1. This point is uniquely determined if X_1 is close enough to the boundary. For instance, if the boundary is the unit circle and X_1 has polar coordinates (ρ, θ) then $B = (\cos\theta, \sin\theta)$. Having computed B we obtain X_0 by reflection (10).

3.2. Junction Boundary

We begin as above by formally approximating the boundary conditions (4)–(6) to second-order accuracy:

$$\mathcal{F} X^1_N = \mathcal{F} X^2_N = \mathcal{F} X^3_N := C \tag{11}$$

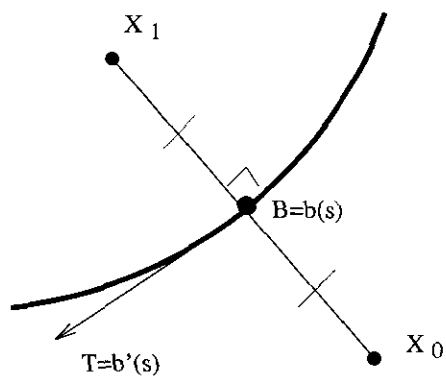

FIG. 2. Details of the implementation of the domain boundary conditions.

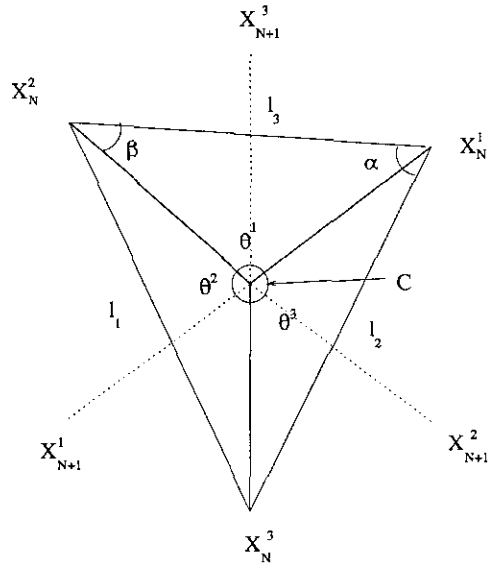

FIG. 3. Details of the implementation of the triple junction conditions.

and

$$\frac{D_+X_N^1}{|D_+X_N^1|} \cdot \frac{D_+X_N^2}{|D_+X_N^2|} = \cos\theta^1 \tag{12}$$

$$\frac{D_+X_N^2}{|D_+X_N^2|} \cdot \frac{D_+X_N^3}{|D_+X_N^3|} = \cos\theta^2 \tag{13}$$

As was the case at the domain boundary, these conditions have a geometrical interpretation. Rewriting (11) like (10), we see that X_{N+1}^i is obtained by reflecting X_N^i through the unknown point C. We substitute this fact into (12)–(13) to obtain

$$\frac{\Delta^1}{|\Delta^1|} \cdot \frac{\Delta^2}{|\Delta^2|} = \cos\theta^1$$

$$\frac{\Delta^2}{|\Delta^2|} \cdot \frac{\Delta^3}{|\Delta^3|} = \cos\theta^2,$$

where $\Delta^i = C - X_N^i$. The situation is pictured in Fig. 3. The point C is a point for which the vectors Δ^i meet at angles θ^1, θ^2, and θ^3 as shown in the figure. This point C is uniquely determined inside the triangle $\angle X_N^1, X_N^2, X_N^3$ if the angles in this triangle are not too large (specifically, the angle at X_N^1 must be smaller than the opposite junction angle θ^2, etc.). This condition is certainly true of the exact values x_N^i if the step size h is sufficiently small and is never violated in the computations described below.

Using the notation of Fig. 3 and simple geometric formulas, we find that

$$\beta = \tan^{-1}\left(\frac{\sin\delta}{l_1 \sin\theta^3/l_3 \sin\theta^1 + \cos\delta}\right), \tag{14}$$

where $\delta = \theta^2 - \alpha$. The angle β is now known and using θ^1 and length l_3 the triangle $\angle X_N^1, X_N^2, C$ is determined completely. Now C can be determined from X_N^1 and then the values X_{N+1}^i are obtained from X_N^i by reflection.

At this point, we have eliminated the values of the approximation X outside the computational domain in terms of values inside using appropriate extrapolation. With these modifications, (7) is an ordinary differential equation involving only interior values. Discretization of curve networks with many curves and junctions is handled similarly.

The key to implementing the junctions here and in the more complicated eutectic case described in Section 8.1 is to first formally write down approximations to the boundary conditions of the correct order involving extrapolated points and then to use a geometric or algebraic interpretation to represent the extrapolated points in terms of the interior points. Other discretizations of this problem are described in Section 6.

3.3. Time Stepping

To solve the system of ODE's describing the approximation of the phase boundary curves presented above, some time-stepping technique must be used. To accurately solve this system and easily identify points of singularity we use the explicit method, standard fourth–order Runge–Kutta (4RK). Time steps k are chosen so that the frozen coefficient parabolic problem corresponding to (7) is stable. This is satisfied if we take

$$k = \frac{h^2}{4} \min_{i,j} |D_1 X_j^i|^2. \tag{15}$$

Note that if the length of any curve goes to zero, the time step restriction becomes more severe. To minimize this difficulty for singularity studies the grid is coarsened as the length of the curve goes to zero. This is described in the next section.

It should be noted that the use of 4RK is "overkill" for this problem; i.e., it is so accurate that there is essentially no temporal error compared to the error from the spatial discretization. However, a nice feature of (15) is that the step size k decreases as the curve approaches singularity, allowing us to accurately identify the singularity times (see Section 5.2). More practical implicit time discretizations are considered in Section 7.

For now, we consider a fixed discretization away from singularity times and consider the numerical convergence of the proposed method.

3.4. Numerical Convergence Study for Smooth Solutions

We consider the evolution of three curves in the unit disk with initial data

$$x^1(\sigma, 0) = (1 - \sigma, \sin^2(\pi\sigma)/4)$$

$$x^2(\sigma, 0) = (1 - \sigma)(-1/2, \sqrt{3}/2)$$

$$x^3(\sigma, 0) = (1 - \sigma)(-1/2, -\sqrt{3}/2).$$

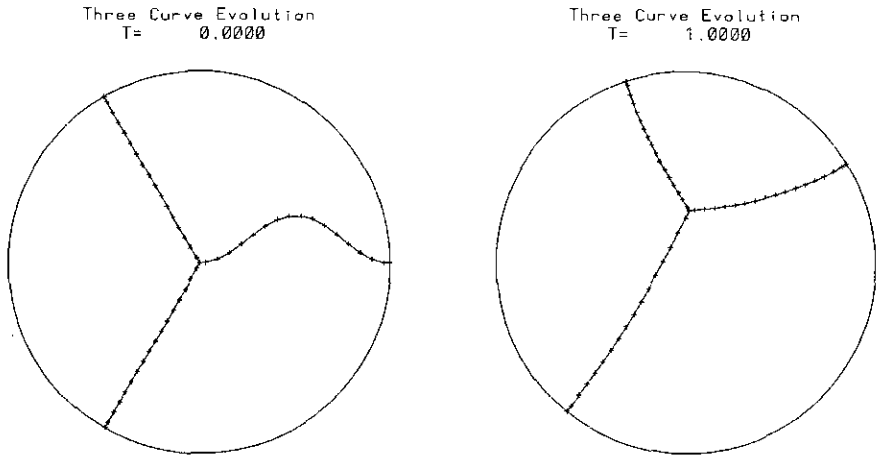

FIG. 4. Network used for the numerical convergence study at $t = 0$ and $t = 1$.

The initial data and computed approximation at $t = 1$ are shown in Fig. 4. Note that the perturbation on curve 1 is flattened and that, since curves 1 and 2 form a convex shape, the area between them will decrease further with time by the curvature flow. Eventually, this area will disappear and the length of the curves 1 and 2 will go to zero. We show how to handle this situation and continue the computation in the next section.

For now, we consider the performance of the method at time 1. Since the exact solution is not known, we compare solution values to those at finer grids to estimate the error e_h:

$$e_h := \|X_h - X_{h/2}\|,$$

where the subscript on X denotes the grid spacing used to compute the approximation and the norm $\|\cdot\|$ is the discrete maximum norm over all points of all curves. Note that linear interpolation must be used to compare the pointwise values of approximations at different grids since we use a staggered grid. Successive error estimates can be used to estimate the convergence rate ρ as follows

$$\rho \approx \rho_h := \log_2 \frac{e_{2h}}{e_h}.$$

Estimates of the error and convergence rate at $t = 1$ are given in Table I. Second-order convergence is clearly seen. A linear stability analysis for the problem using an interesting new technique specific to this discretization is given in an Appendix. Convergence rates after singularity time are examined computationally in Section 5 below.

3.5. Regridding

The "philosophy" of approximation we have chosen is to pick a parameter h that remains fixed throughout the computation and try to maintain at least one grid point for every arc length h along the discretized curves. In order to maintain this property, we double the number of grid points in a curve (refinement) when points get too far apart and half the number of grid points (coarsening) when the curve gets shorter. Note that the discrete equations (7) and corresponding boundary conditions are actually independent of the choice of h so this parameter can be kept constant essentially as a bookkeeping parameter as the number of points discretizing a given curve change in time. For each curve i we compute

$$\nu^i = \max_j |D_1 X_j^i|^2$$

$$\mu^i = \min_j |D_1 X_j^i|^2.$$

If $\nu^i > 4$ then the curve is refined; i.e., the number of discrete points on the curve is doubled and the values on the new refined staggered grid are obtained by linear interpolation. If $\nu^i < \frac{1}{4}$ then the curve is coarsened; i.e, the number of points on the curve is halved and values are again obtained by linear interpolation.

The rules above basically ensure that there will be at least one grid point for every $2h$ length with a maximum distance between points of more than $h/2$. Refinement helps keep resolution in curves that are stretched and coarsening reduces the

TABLE I

Estimated Errors and Convergence Rates for Smooth Solutions

$N = 1/h$	e_h	ρ_h
8	0.3601e-2	
16	0.9133e-3	1.98
32	0.2311e-3	1.98
64	0.5818e-4	1.99

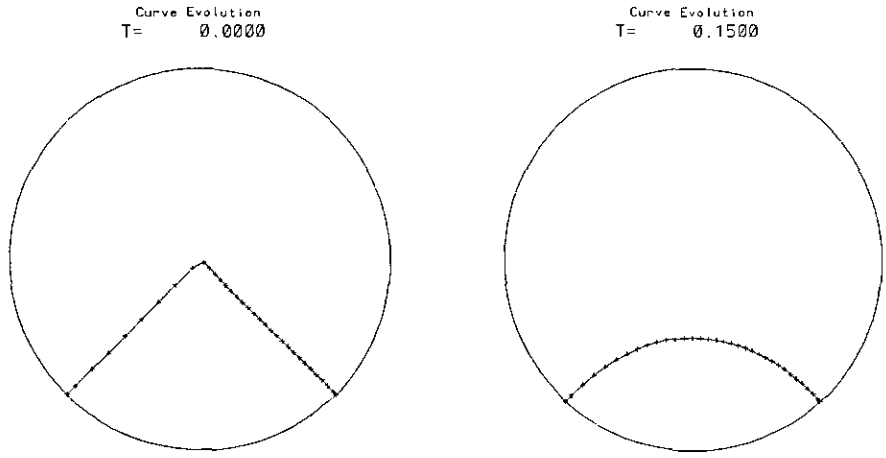

FIG. 5. A discrete smoothing property.

time step restriction (15). Computational studies show that second-order convergence for smooth solutions is maintained through single refinement or coarsening. Studies showing the performance of the method through many regriddings and singularities are described in Section 5.

The regridding described above is of "global" type; if a curve is locally deficient in points, the whole curve is refined. However, for curvature motion with our parametrization this is not a problem, as shown in the next section.

3.5.1. *A Property of the Equations*

We consider the evolution of a single curve x (attached at both ends to the boundary) with piecewise linear initial data.

$$x(\sigma, 0) = \begin{cases} (1 - 4\sigma)(-1, -1)/\sqrt{2} & \text{if } \sigma \leq \frac{1}{4} \\ (\sigma - \frac{1}{4})(1, -1)/\sqrt{2} & \text{if } \sigma > \frac{1}{4}. \end{cases} \quad (16)$$

This is the kind of situation that occurs after a singularity is detected and removed and the accuracy of our method applied to this problem is discussed below. For the present, we wish to show another property of solutions to Eqs. (1). The discrete solution at $t = 0$ and $t = 0.15$ with the above initial data is shown in Fig. 5, where the grid points are marked. Note that the spacing between the grid points becomes more uniform with time. This is also true for curve networks away from singularities and allows us to do global regridding as described above since local distortions in the grid are smoothed out.

This property is due to the full parabolicity in the system of Eq. (1). We compare (1) to the modified system

$$x_t^i = \left(\frac{x_{\sigma\sigma}^i}{|x_\sigma^i|^2} \cdot \hat{n} \right) \hat{n}, \quad (17)$$

where $\hat{n}$ is the normal vector. This system corresponds to that studied in, *e.g.*, [9, 11] to describe the evolution of single curves by curvature motion. This system also describes curvature motion but is not fully parabolic. In fact, a linearization shows that this system is of mixed type, parabolic in the normal component and hyperbolic in the tangential component. A discretization of this system will not have the smoothing property as above and may lead to theoretical difficulties when trying to prove the well-posedness of the boundary conditions (2)–(6).

It is appropriate to mention here the technique in [24] where tangential motion is introduced in the tracking points exactly to maintain equal spacing in arc length. Unfortunately, this technique cannot be used here due to the Dirichlet component to the junction and domain boundary conditions.

3.6. Singularity Detection

Curves are never coarsened to fewer than two points. A curve i that has been coarsened to two points and has $\mu^i < \varepsilon h^2$ is scheduled for deletion and is removed. This condition is designed to give a second-order error (in h) in the computed singularity time assuming the junctions have finite speed at singularity. The parameter ε is chosen to be 0.1 in the rest of the computations described in this paper. Computationally, we see that the dependence of the singularity time estimates on ε is negligible compared to the h dependence.

After a curve is deleted, any remaining curves are reorganized so that computation can be continued. The type of "surgery" performed depends on the particular phase model under consideration. The details for two models are given below.

4. SURGERY AND NUMERICAL RESULTS

The discretization in the previous section applies to the evolution of curve networks up to singularity time, when one or more of the curves shrinks to zero length. At this point, information about the physical system being modeled must be used to be able to continue the computation. The details of this selection are given below for two different phase models. We also demon-

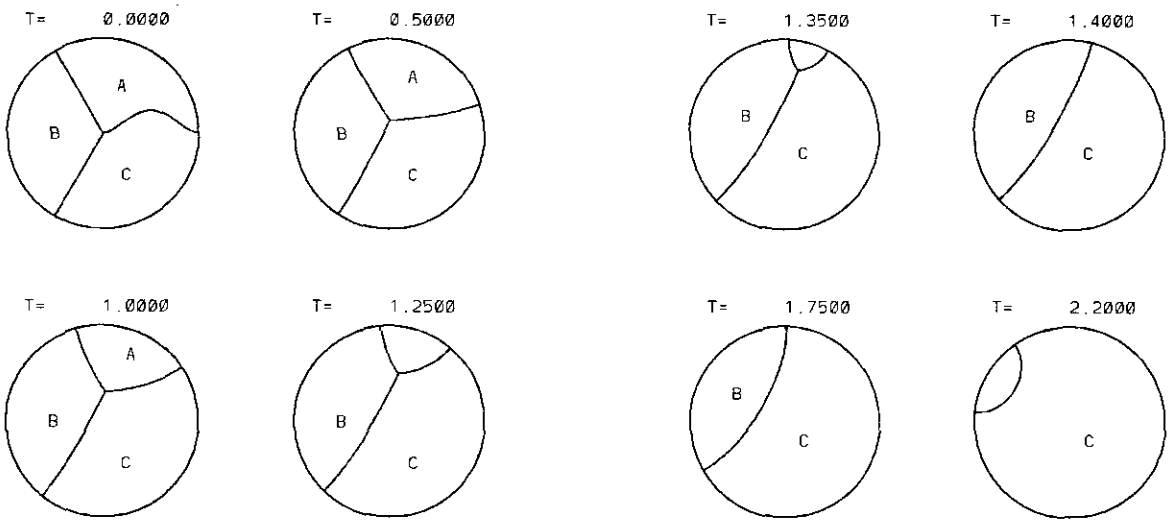

FIG. 6. A single node network for the three phase model.

strate that our method agrees with the derived area and length laws for the case of symmetric angles at triple junctions. Finally, we present some simulations with non-symmetric angles at the triple junction and simulations in other physical domains.

4.1. Three Phase Model

The first phase model we consider is one in which only three different phases (labeled A, B, and C) are present in the domain and equal angles of 120° are imposed at triple junctions. This corresponds to the case that the potential is symmetric as mentioned in the Introduction. As our first example, we consider the evolution of three curves in the unit disk that separate these phases. The initial data and computed approximations are shown in Fig. 6. In this computation and the others depicted in this section we take $N = 16$. We specify initial data by giving initial node (triple or boundary junction) positions and orientations. Then the curves are filled in by using a spline fit through the node points and specified curve mid-points.

We now discuss the results shown in Fig. 6. At $t = 0.5$ the initial perturbation on the right curve has been significantly reduced and the convexity of the region of phase A indicates that its area will decrease further with time. Eventually, this region disappears and the length of the two curves that bound it will go to zero (at $t \approx 1.396$). In order to continue the computation we perform the following physically motivated surgery: we delete the two curves of vanishing length and attach the end of the remaining curve normally to the domain boundary. Some technical details are given in Section 4.1.1 below of the numerical implementation of the surgery. We call this singularity of type 1 in what we conjecture is a complete generic list of six singularity types shown in Figs. 7 and 8. We continue the computation after surgery and as seen in Fig. 6, the remaining curve will eventually disappear and be removed (a type 2 singularity as shown in Fig. 7) leaving no curves and only phase C remaining in the domain.

The evolution of a network of curves with six triple junctions is shown in Fig. 9. Here, the phase labels have been dropped, but each region between the curves can be assigned uniquely to one of the three phases (up to relabeling). Six singularities are observed in the sequence 424252 using the notation of Figs. 7 and 8. We shall describe next the first singularity observed

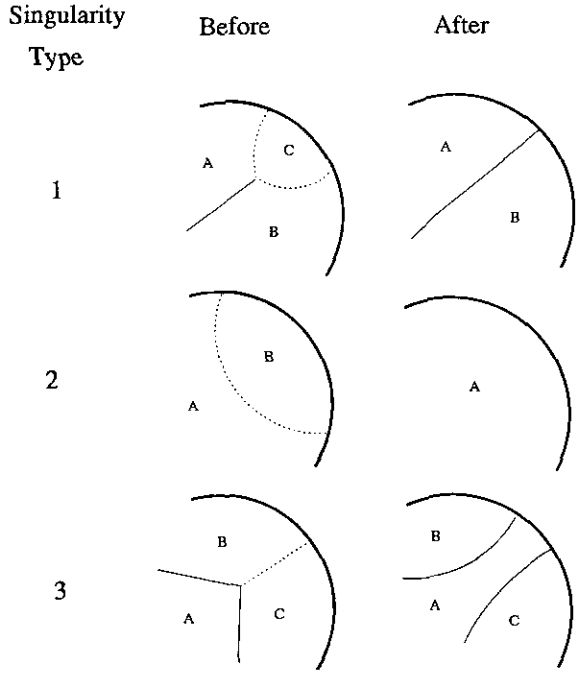

FIG. 7. Generic singularities for the three phase model involving the domain boundary. Dashed lines show curves whose length is going to zero.

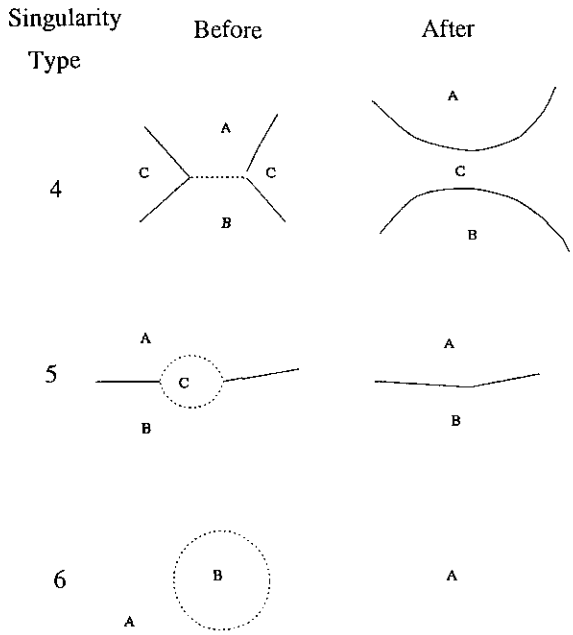

FIG. 8. Generic singularities for the three-phase model that do not involve the domain boundary. Dashed lines show curves whose length is going to zero.

and show how we use the information given by the physical model under consideration to do the surgery. The first singularity (at $t \approx 1.3428$) is of type 4; i.e., the interface which separates two distinct phases shrinks to zero length. At this time, since one phase is now on both sides of this unstable quadruple junction, we join it by splitting the junction as in Fig. 8. The other singularities are handled as in Figs. 7 and 8.

In order to observe the remaining generic singularity types, we present the evolution of an eight-node network (Fig. 10) and a four-node network (Fig. 11). The first singularity in the eight-node network is of type 3, and in the four-node network, a "bubble" is squeezed off leading to a type-6 singularity.

We note that the generic singularity list conjectured in Figs. 7 and 8 allows for only one or two triple junctions to collide at one time. It should also be noted that the surgery performed in these computations is physically "reasonable" but has not been justified in a rigorous way. Numerical or asymptotic limits of solutions of the original Ginzburg–Landau model [1] at these singularity times should justify our procedure but such a program has not been carried out.

4.1.1. *Numerical Implementation of Surgery*

We keep track of the changing topology of the curve network using the following information: for each curve, the number and positions of interior discrete points and the phases to the right and left of each curve; for each curve end, whether the end is attached to the domain boundary or the interior junction it is attached to; for each interior junction, an oriented list of incident curves. The number of interior points for each curve is always kept to be a power of 2 for ease of regridding. Away from the singularity times, the information about the curve ends is used to obtain appropriate extrapolated values to be able to update the interior points near the ends of the curves. When a singularity is detected (i.e., a curve is scheduled for deletion using the criteria of Section 3.6) the information listed above is modified to reflect the changed topology of the network. We give two examples below.

A type-1 singularity (see Fig. 7) is characterized by two curves that simultaneously satisfy the condition for deletion and meet at a common junction at one end and the domain boundary at the other end. These curves are removed. The remaining curve end that is incident on the common junction is changed to the domain boundary type and the junction is removed since it no longer has any incident curves. In this and future time steps, the extrapolation needed to update the interior points of the remaining curve will be done using the domain boundary conditions.

A type-4 singularity (see Fig. 8) is detected when a single curve satisfies the condition for deletion and this curve is connected at both ends to interior nodes. This curve is deleted. Using the right and left side phase information, the two pairs of curves that need to be joined can be grouped from the remaining four curves incident on the two junctions. These two junctions are then removed. For each pair of curves, points are put in to a common curve, after the curve with the fewer points has been refined so that both curves have the same number of points (2^n). The resulting curve will have 2^{n+1} points (again, this is convenient for regridding). In this and future time steps, points near the old nodes will evolve as interior points on the new curves. Derivative discontinuities will smooth out as shown in Section 3.5.1. Other singularities are handled similarly.

4.2. Crystal Grain Boundary Model

In crystal grain growth, every crystal grain has a different lattice orientation so every grain corresponds to a different phase. We again consider first the symmetric case at the triple junction. If we use this material model in our computations, we must replace the surgery procedure 4 of Fig. 8 because this would now join two different materials. Instead, we use a surgery labeled 4* shown in Fig. 12, in which a new curve is introduced between the two distinct phases with the appropriate 120° angle conditions.

A calculation of a network with six triple junctions using this model is shown in Fig. 13. The same initial data as the computation shown in Fig. 9 is used so the evolution of the network will be identical up to the first singularity time ($t \approx 1.3428$). Singularities of type 4*14*14*4*51 are observed in this computation during the times depicted.

The singularity sequence 4*4*5 near time 1.8 is too small to see clearly in Fig. 13 and a blow-up (of factor 12.5) of the network near these singularities is given in Fig. 14. Such small scale multiple collapse of curves is common in the numerical

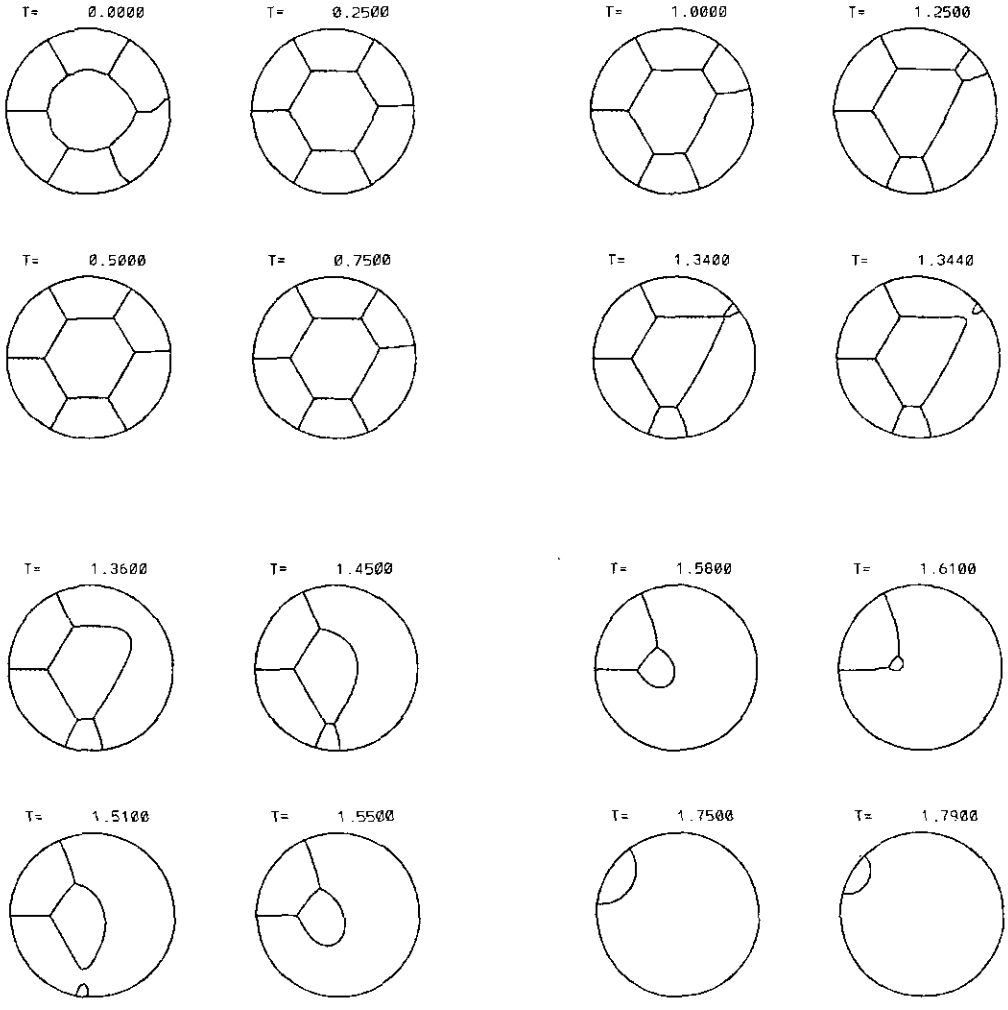

FIG. 9. A six-node network for the three-phase model.

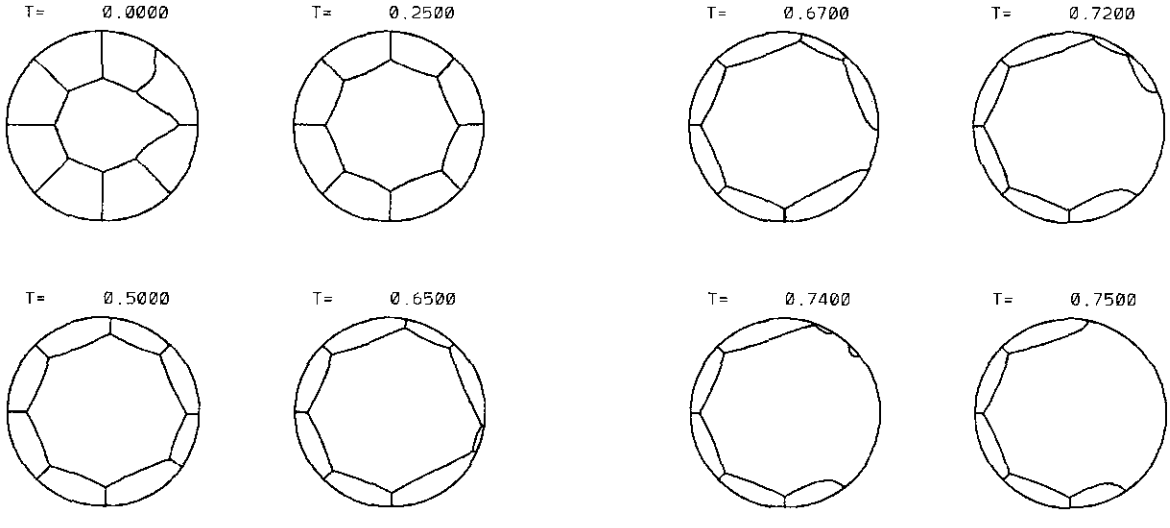

FIG. 10. An eight-node network for the three-phase model.

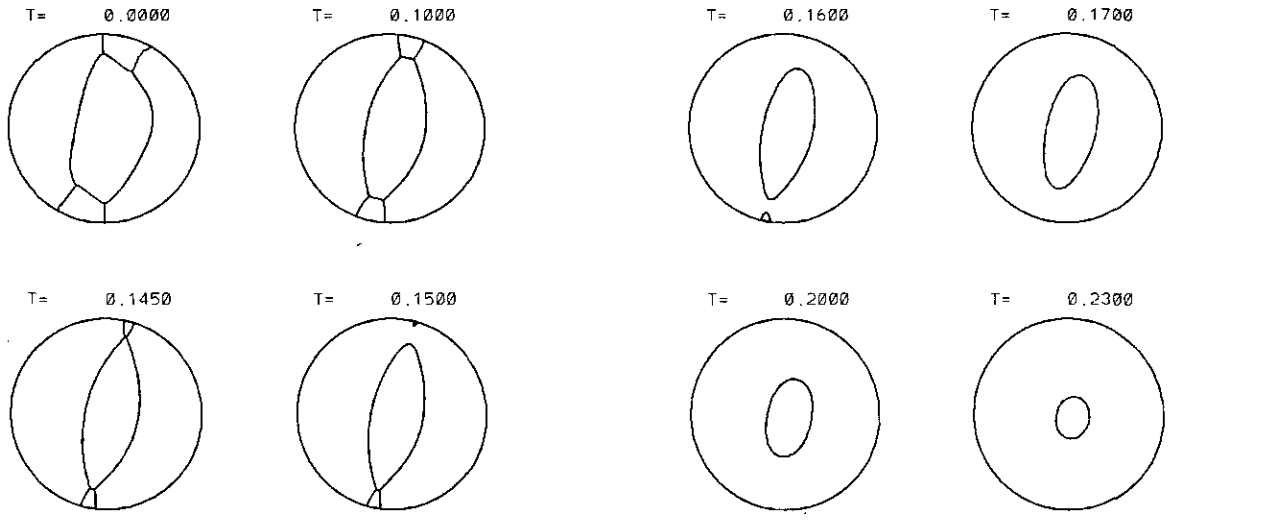

FIG. 11. A four-node network for the three-phase model.

tests we have done. The collapse sequence is reproducible using finer grid resolution so we have confidence in its structure.

In the absence of boundaries, we conjecture that the generic singularities in a network of grains are type 5 (the collapsing lens from Fig. 8) and type 4* (where the curve neighbors are switched as discussed above). This case was studied previously in [7] but they obtain different results. Indeed, a similar figure to Fig. 14 is shown in [7], where a symmetric collapse of a four-sided grain to a lens and two more junctions is shown. We believe this is a non-generic situation due to symmetric initial data.

4.3. Approximation of Area and Length Laws

In the symmetric case, a phase region surrounded completely by n curves that join at triple junctions (i.e. a region not adjacent to the domain boundary) has area A that obeys the Von Neumann–Mullins parabolic law:

$$\dot{A} = \frac{\pi}{3}(n - 6). \tag{18}$$

This law is derived for the curvature motion case in [18]. We compare the computed area (based on a polygonal approximation of the curves) for the enclosed region of Fig. 9 to that predicted from (18) (using the computed area as initial data and the singularity times from the calculation to change n). The two quantities are shown in Fig. 15 on the left graph and are almost identical. Here, a less accurate computation with $N = 8$ has been used to see any difference at all. The initial flat portion of the graph corresponds to the times when the region is enclosed by $n = 6$ curves and then the area decreases when it is enclosed by four and then two nodes. An area enclosed by seven or more curves will increase as seen in Fig. 10.

The total length of the curve network is also predicted to decrease (at a rate equal to the total integral of the square of the curvature over the network) and the computed values of total length shown on the right-hand side of Fig. 15 satisfy this property. Note that the relatively flat initial part of the graph corresponds to the portion of the computation in Fig. 9, where the lines are almost straight (i.e., have small curvature).

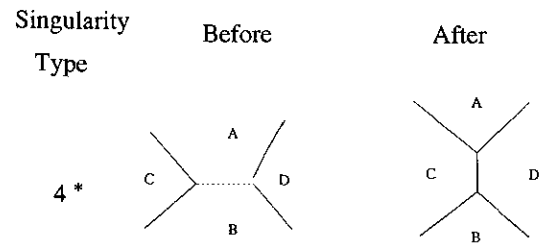

FIG. 12. Singularity 4* for the grain growth model.

4.4. Extensions

The numerical method is easily extended to the case where the angles at the triple junction are different from 120°. For example, a computation with three curves meeting at angles of 160°, 140°, and 60° is shown in Fig. 16. The method is also easily extensible to any convex domain. A computation with three curves meeting at 120° angles in a square domain is shown in Fig. 17. Finally, we note that the method is also easily extensible to include anisotropic effects in the curvature motion and non-normal incident angles at the domain boundary.

5. PERFORMANCE OF THE METHOD AFTER SINGULARITY

After singularity time, curves are joined together to form new continuous curves with discontinuous derivatives at isolated

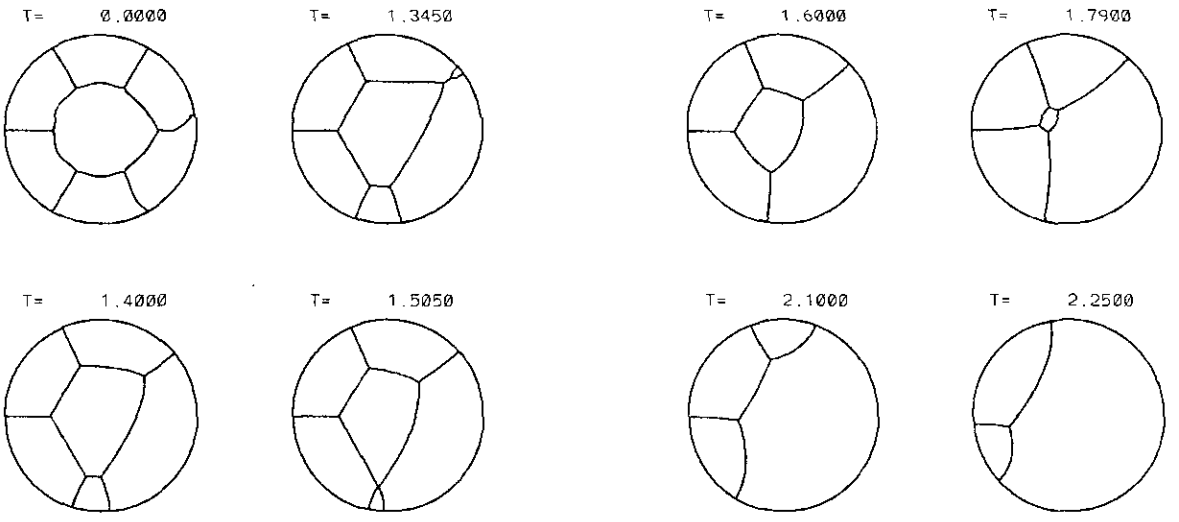

FIG. 13. A six-node network with crystal grain boundary model.

points. The performance of the method in this situation is shown below.

5.1. Convergence with Derivative Discontinuities

As a model for the behavior of the method after a singularity, we consider the problem of a single curve with initial data (16). Estimated maximum norm errors and convergence rates for this problem are shown in Table II. These estimates and those for similar computations with initial derivative discontinuities in the interior of the curve or at the boundary show convergence rates of "almost" second order. We speculate that the convergence is of a rate less than second order but not by a factor of h^α, i.e., that the error behaves something like $h^2|\log h|$. We are currently looking for a model for this type of behavior that can be analyzed exactly.

5.1.1. *Other Discontinuities*

Behavior at a singularity may introduce stronger discontinuities than those modeled above as the following example shows. The self-similar lens solution of the phase curve problem from [18] is shown in Fig. 18 (the arcs are not quite circular, they are rosette-like curves). Due to the area law described in Section 4.3, $r(t) = \sqrt{r^2(0) - Ct}$ for some constant C.

A singularity (type 5) occurs at $t = r^2(0)/C$ and at this time, the speed of the triple junctions is infinite. Therefore the right-hand side of (1) must be infinite. Numerical evidence suggests that for the curves outside the lens (that remain after surgery) the parametrization does not fail; i.e., $|x_\sigma|$ stays bounded away from zero. Therefore, the singularity must involve the blow-up of the second derivative. Rather than examine in detail the structure of the solutions with our parametrization in such situations and then determine the expected performance of the

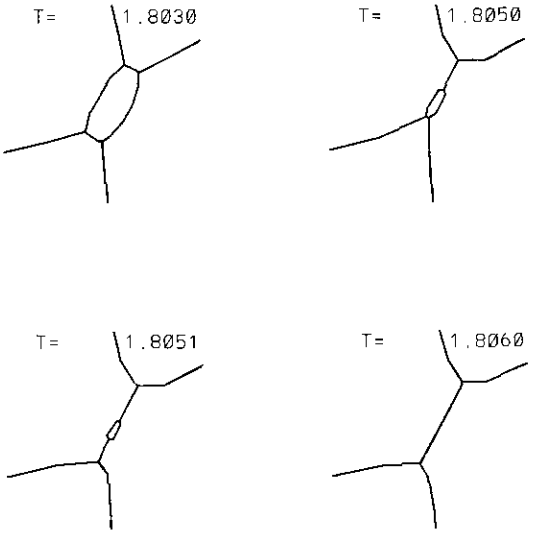

FIG. 14. A blow-up of the evolution of the network of Fig. 13.

TABLE II

Estimated Errors e_h (and Convergence Rates ρ_h) for the Single Curve with Initial Derivative Discontinuity

$N = 1/h$	$t = 0.04$	$t = 0.10$
16	0.4355e-2	0.3062e-2
32	0.1301e-2 (1.74)	0.8979e-3 (1.77)
64	0.3622e-3 (1.84)	0.2517e-3 (1.83)
128	0.1009e-3 (1.84)	0.6922e-4 (1.86)
256	0.2777e-4 (1.86)	0.1890e-4 (1.87)

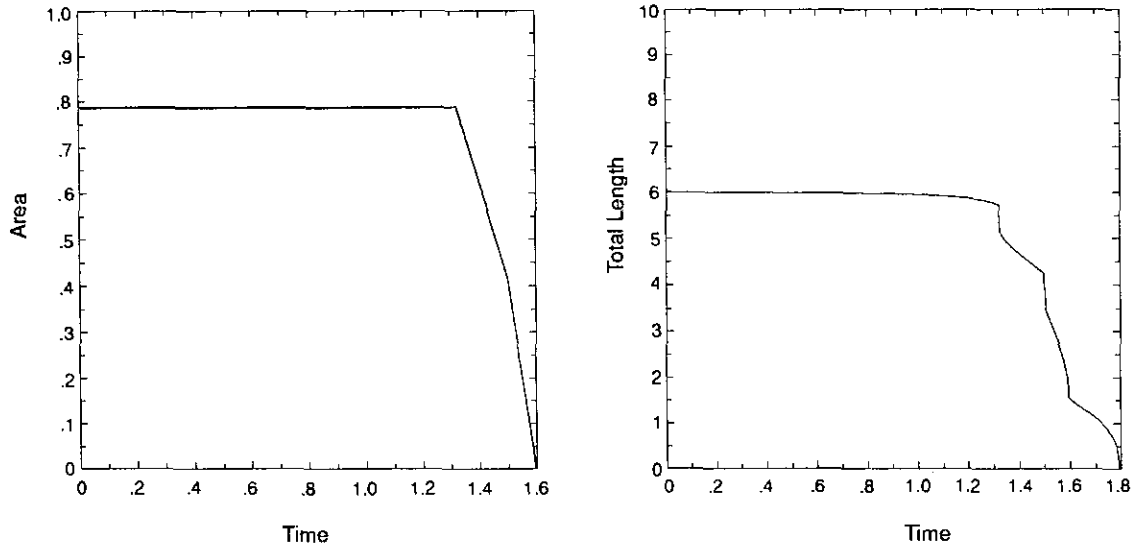

FIG. 15. Area and total length behavior of the six-node network.

method, we observe the numerical behavior directly. This is done in the next section.

5.2. Convergence of Singularity Times

In order to evaluate the performance of the method through multiple regridding and singularity times we consider the convergence of the numerically determined singularity times. The estimates for the six singularity times of the run shown in Fig. 9 are given in Table III. Convergence in singularity times is observed of "almost" second order as discussed above. This is also observed for many other runs, giving us confidence in the method.

6. OTHER DISCRETIZATIONS

In this section we construct other second order and higher order discretizations of junctions to compare with the one above. In addition we present other discretizations of the interior equations.

6.1. Third-Order Dirichlet Correction

Without widening the stencil for the staggered grid method presented above, we can improve the approximation at the

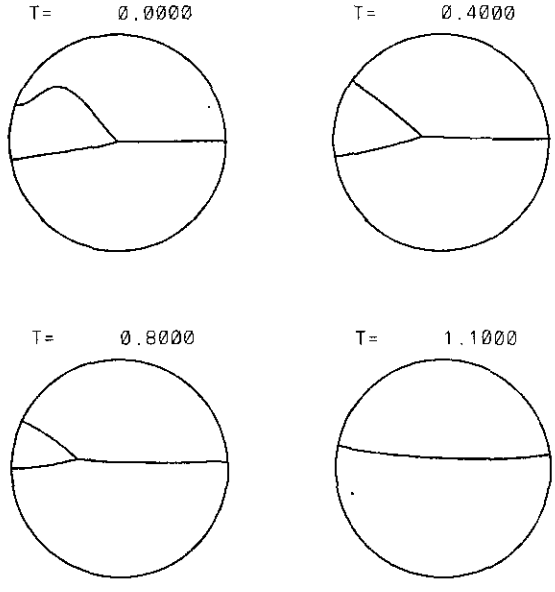

FIG. 16. The evolution of a curve network with angles 160°, 140°, and 60°.

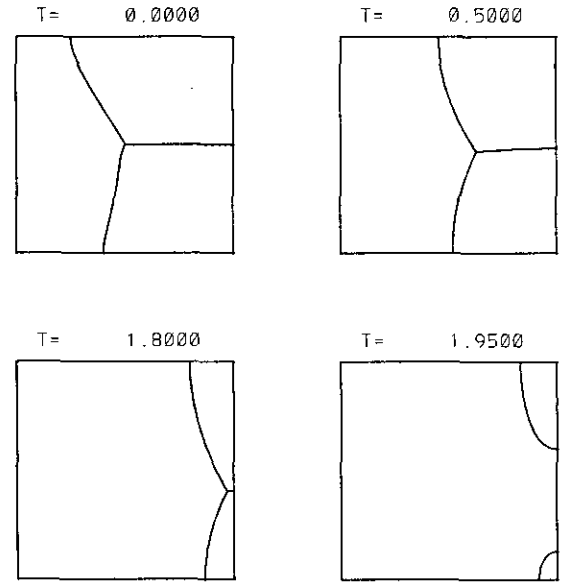

FIG. 17. The evolution of a curve network in a square domain.

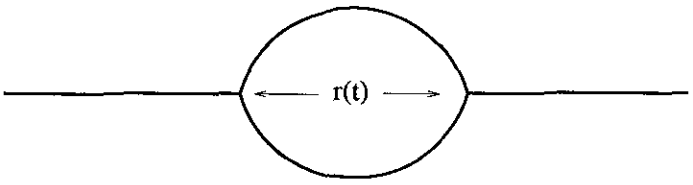

FIG. 18. A self-similar solution that goes to singularity.

domain boundary by replacing the second-order approximation (8) by the third-order approximation

$$B = \tfrac{3}{8} X_0 + \tfrac{3}{4} X_1 - \tfrac{1}{8} X_2. \tag{19}$$

Using the same approximation to the Neumann condition (9) and reasoning as in Section 3.1 we discover that B must be the closest point to

$$\tfrac{9}{8} X_1 - \tfrac{1}{8} X_2$$

and X_0 can be found from B using (19). This amounts to a more accurate "reflection" at the boundary. A similar approach can be taken at the triple junction. The resulting method is applied to the smooth test problem described in Section 3.4. The approximate errors and convergence rates are shown in Table IV.

As expected, the method still converges with second order but the error size is much reduced (by a factor of 10 from the original method—compare Tables IV and I). This shows that a significant part of the error comes from the boundary discretization. Unfortunately, the method above with the third-order Dirichlet correction performs very badly near a singularity so it cannot be used for anything but smooth solutions. It is not surprising that a higher order method would fail when derivatives in the solution become singular.

TABLE III

Estimated Singularity Times (and Estimated Convergence Rate in Singularity Times) for the Run Shown in Fig. 9

N	First	Second	Third
8	1.3224571	1.3263752	1.4992721
16	1.3428338 (1.04)	1.3467354 (1.49)	1.5074870 (3.19)
32	1.3527337 (1.75)	1.3566085 (1.73)	1.5083897 (1.67)
64	1.3556852 (1.92)	1.3596308 (1.91)	1.50867366 (1.88)
128	1.3564617 (1.99)	1.3604571 (1.97)	1.50875070 (1.96)
256	1.3566571	1.3606710	1.50877046

N	Fourth	Fifth	Sixth
8	1.5077970	1.5954315	1.7919395
16	1.5146238 (2.90)	1.6141944 (1.05)	1.8202979 (1.39)
32	1.5155387 (1.32)	1.6208964 (1.71)	1.8311081 (1.81)
64	1.5159041 (1.62)	1.6229145 (1.87)	1.8341904 (1.83)
128	1.5160233 (1.73)	1.6234531 (1.95)	1.8350557 (1.84)
256	1.51605915	1.6235906	1.8352974

TABLE IV

Estimated Errors and Convergence Rates for Smooth Solutions Using Third-Order Dirichlet Correction

$N = 1/h$	e_h	ρ_h
8	0.3632e-3	
16	0.9991e-4	1.86
32	0.2471e-4	2.02

6.2. A Regular Grid Method

We now consider a regular grid approximation of the curve x, i.e., $X_j \approx x(jh)$. At the domain boundary we can now approximate (2) directly by

$$X_0 = B,$$

where B is as usual an unknown point on the boundary. We approximate (3) using a second-order one-sided approximation

$$(-\tfrac{1}{2} X_2 + 2X_1 - \tfrac{3}{2} B) \cdot T = 0,$$

where T is the boundary tangent at B. Using the same procedure as in Section 3.1 we combine the two equations above to conclude that B must be the closest point on the boundary to

$$-\tfrac{1}{3} X_2 + \tfrac{4}{3} X_1.$$

A similar approach can be taken for triple junctions. This discretization was first considered in [8]. We apply this boundary discretization to the smooth calculation of Section 3.4. The results are shown in Table V.

Again, the convergence is second order as expected. However, the error sizes are much larger than for the staggered grid method originally described. We could have predicted this since the local truncation error for short centered difference approximations of the Neumann condition for the staggered grid is smaller by a factor of 8 than that for second-order one-sided differencing.

6.3. Interior Discretization

Other authors [8, 7] do not discretize a term like the right-hand side of (1) but rather use a geometrical approach to curvature motion: Considering a point and its two neighbors, a unique circular arc can be constructed, giving an approximation to the curvature as well as the normal direction. It can be shown that this discretization is a second order accurate approximation of

TABLE V

Estimated Errors and Convergence Rates for Smooth Solutions Using the Regular Grid Discretization

$N = 1/h$	e_h	ρ_h
8	0.2496e-1	
16	0.4293e-2	2.54
32	0.8922e-3	2.27

a curvature motion law but one like (17) which is only partially parabolic. As discussed in Section 3.5.1, the partial parabolicity leads to poor redistribution of grid points. The method in [7] introduces "artificial" tangential motion or point redistribution to overcome this problem. Such an approach is computationally intensive and can lead to excessive smoothing from the process of discrete redistribution [24].

7. IMPLICIT TIME STEPPING

In order to avoid the excessively small time steps (15) of explicit time-stepping methods we now consider implicit techniques. For simplicity we explore a backward Euler time-stepping scheme, although the structure of the resulting system of equations is identical for schemes like Crank–Nicholson or higher order multi-step methods. The time step k is taken to be constant and a second superscript n denotes the time level, i.e., X_j^n approximates $x^j((j-\frac{1}{2})h, nk)$. We consider an introductory problem with three curves and symmetric angles in a disk. Given the position of the three curves at time n we find the positions of the curves at time level $n+1$ by solving the system

$$\mathcal{G}(X^{n+1}) = 0, \tag{20}$$

where

$$\mathcal{G}(X) = X - k\frac{D_2X}{|D_1X|^2} - X^n.$$

The extrapolated points at the triple junction at time level $n+1$ satisfy (11)–(13) and the extrapolated points at the domain boundary satisfy

$$X_0 + X_1 - 2\frac{X_1}{|X_1|} = 0$$

for all curves. We consider three techniques for solving this system of equations and predict the performance for large networks. The methods are applied to a first step of the example network described in Section 3.4.

7.1. Newton Iteration with Sparse Matrix Solver

We perform simple Newton iterations beginning with the curve positions from the previous step:

$$X^{(j+1)} = X^{(j)} - [\nabla_X \mathcal{G}(X^{(j)})]^{-1}\mathcal{G}(X^{(j)}). \tag{21}$$

The single superscript (j) on X denotes the iteration level now. The matrix $\nabla_X\mathcal{G}(X)$ can be written symbolically as

$$I - k\frac{D_2}{|D_1X|^2} + \frac{(2kD_2X)D_1X \cdot D_1}{|D_1X|^4},$$

where I is the identity matrix. This is equivalent to the implicit discretization of a convection diffusion equation as the linearization of (1) suggests. This matrix is (2×2) block tridiagonal. The domain junction terms do not widen the diagonal structure but the triple junction boundary terms link all the unknowns at the ends of the curves. We use an extension of the Thomas algorithm (described in, e.g., [26]) to reduce the problem to three block tridiagonal solves per curve followed by a 6×6 system to determine the update to the extrapolated points at the triple junction for the three curves. Convergence is quadratic and requires only a few iterations ($\approx$4 for an error less than 10^{-5}) for the example described in Section 3.4 for a wide range of values of h and k.

For a network with K junctions the same technique can be used. One, three, or five block tridiagonal solves must be done per curve (depending on whether it joins zero, one, or two triple junctions, respectively) followed by the solution of a $6K \times 6K$ system representing the coupling effects of the updates to the curve positions through the junctions. While this particular technique (the extension of the Thomas algorithm) may not always be the most efficient way to solve the system resulting from a given network, it is a convenient way to show that the sparse solution process will be a combination of local block tridiagonal solves with coupling through the junctions. The part of the system corresponding to junction coupling is sparse but not banded and can be large since large networks are of interest in statistical studies of grain size [8]. Since the cost of solving this part of the problem may dominate the total solution time, we consider two alternative methods for solving the system (20) that update the positions of single curves locally and then update junction positions.

7.2. "Junction" Iteration

The linear system of equations above would be easy to solve if the junction positions were known. We modify the method above to use Dirichlet conditions at triple junctions based on the junction centers at the previous iteration; i.e., we use

$$\frac{X_N^{i,(j+1)} + X_{N+1}^{i,(j+1)}}{2} = C^{(j)}$$

to eliminate the extrapolated value $X_{N+1}^{i,(j+1)}$ from the stencil, where $C^{(j)}$ is the junction center predicted from the previous

TABLE VI

Number of Iterations to Convergence for a Single Time Step k Using Junction Iteration

$N = 1/h$	$k = 0.01$	$k = 0.04$	$k = 0.16$
8	12	28	61
16	25	58	124
32	53	120	252
64	108	243	
128	219		

step. Computing $X^{(j+1)}$ is now a simple block tridiagonal solve for each curve. The new values $X_N^{i,(j+1)}$ for $i = 1, 2, 3$ can be used to find a new curve center $C^{(j+1)}$ as described in Section 3.2. This method is essentially the solution technique used in [8], except that they use the regular grid discretization described in Section 6.2.

This technique is applied to the example from Section 3.4. Convergence is linear and the number of iterations required to get an error of less than 10^{-5} (to an accurate solution from the sparse Newton technique) is shown in Table VI. Note that the number of iterations required grows like $k^{1/2}N$. In order to understand this behavior we study a simple model below.

7.2.1. *Model of Junction Iteration*

Junction iteration essentially involves replacing discrete Neumann conditions (12)–(13) by discrete Dirichlet conditions ((11) with C known) and then updating the Dirichlet conditions using the extrapolation formula. The convergence is limited by the linear convergence of this procedure not the quadratic convergence of the nonlinearity. Also, we expect that the method's performance should only depend on the nature of the solution near the boundary. Therefore we consider the following simple linear constant coefficient model in a half plane:

Consider a regular grid finite difference discretization of

$$u_t = u_{xx}; \qquad x \in [0, \infty); \qquad u_x(0, t) = 0$$

with backward Euler time stepping. Using the notation of the previous sections, the discrete problem is

$$(I - kD_2)U^{n+1} = U^n. \tag{22}$$

We use a second-order one-sided approximation of the Neumann condition

$$(-\tfrac{1}{2}U_2 + 2U_1 - \tfrac{3}{2}U_0)/h = 0. \tag{23}$$

Normally, the expression (23) would be incorporated into the boundary stencil of (22) but to model the junction iteration we compute iterated approximations $U^{(j)}$ to U^{n+1} as follows:

1. Set $U_0^{(0)} = U_0^n$.
2. Compute $U_l^{(j+1)}$ for $l > 0$ by solving (22) with Dirichlet data $U_0^{(j)}$.
3. Update U_0 using the extrapolation formula (23), i.e.,

$$U_0^{(j+1)} = \tfrac{4}{3}U_1^{(j+1)} - \tfrac{1}{3}U_2^{(j+1)}.$$

4. Continue 2–3 until the desired accuracy is reached.

We will show that the iterations converge for all h and k. The real question is whether the map $U_0^{(j)} \mapsto U_0^{(j+1)}$ is contracting and at what rate. To investigate the contraction rate we can ignore the data term U^n in (22) by considering differences of successive iterations. Suppose now $U_0^{(j)} = a$. We solve (22) with zero data and Dirichlet data $U_0 = a$ to obtain

$$U_l = ae^{-\beta l},$$

where $\cosh \beta = 1 + h^2/2k$ and $\beta > 0$ (verify by direct substitution). By extrapolation (step 3 above) we find that

$$U_0^{(j+1)} = U_0^{(j)} f(e^{-\beta}),$$

where $f(x) = \frac{4}{3}x - \frac{1}{3}x^2$ is the contraction rate. Since $x \in (0, 1)$ we have $0 < f < 1$ so the method converges for all choices of h and k. For h^2/k small, $\beta \approx h/\sqrt{k}$ and $f \approx 1 - \frac{2}{3}h/\sqrt{k}$. This expression confirms the numerical prediction of the performance of the junction method above: convergence to a fixed accuracy requires $O(N\sqrt{k})$ iterations (recall $N = 1/h$).

The use of a staggered discretization leads to $f(x) = x$ and $f \approx 1 - h/\sqrt{k}$ (similar asymptotic character but with a better constant). With Crank–Nicholson time stepping this becomes $f \approx 1 - h/(2\sqrt{k})$ again with the same asymptotic character. A similar analysis can be performed on junction-type iterations on vector parabolic problems with mixed Neumann and Dirichlet conditions under suitable assumptions on the well-posedness of the boundary conditions and the stability of the discrete schemes used.

7.3. Multi-Grid Iteration

The junction iterations above can be thought of as a block iteration with blocks consisting of the interior points of each curve and the extrapolated points for each junction. As seen in the model above, the performance of the iterations deteriorates as the grid is refined. The typical multi-grid (MG) philosophy is to reduce the errors left after iteration on fine grids with corrections on coarser grids. See [12] for a general discussion of MG and a description of the terms used below.

A MG approach can be taken with the implicit curve network problem. It is actually more efficient to break up the curve blocks and perform nonlinear Gauss–Seidel iteration on the interior equations separately, followed by an update of the extrapolated points as in the junction iteration above. This is followed by a recursive correction on successively coarser grids

TABLE VII

Number of Iterations to Convergence for a Single Time Step k Using MG Iteration

$N = 1/h$	$k = 0.01$	$k = 0.04$	$k = 0.16$
8	2	4	15
16	3	4	13
32	3	4	13
64	3	4	13
128	3	4	13

using the FAS technique with V-cycles. The performance of the method is shown in Table VII and is much superior to the junction iteration method. Convergence level is 10^{-5} as before. Note that for fixed k the number of iterations is independent of N as expected from MG theory. The deterioration of the method for large time step size can be rectified by the use of W-cycles (more appropriate when the initial guess is far from the discrete solution) although for reasonable time steps the use of V-cycles is more efficient. A technical detail that makes the Gauss–Seidel iterations very efficient is that the two components of X are decoupled at each grid point since the "diffusion matrix" is diagonal and centered differences are used for first derivatives (a point similar to that discussed [12, p. 184]).

7.4. Comparison of Solution Techniques

For $N = 16$ and $k = 0.04$ ("reasonable" values) the computational times (in SPARC-2 seconds) to reach 10^{-5} accuracy were 0.20, 2.41, and 0.05 for the sparse Newton, junction iteration and MG methods, respectively. For larger networks the computational time will grow linearly with the number of curves for all three methods (with some additional overhead in the sparse Newton technique to handle the coupling between the many junctions), assuming the same discretization level for each curve. It appears that the MG method gives the best performance for the implicit network problem. It is rather unusual to use a MG method on a problem with dependence on only one space variable: in this case the use of a MG method allows us to efficiently handle the coupling of the curves at the junctions.

8. OTHER TYPES OF JUNCTIONS

We describe other types of junctions for curve networks that are suitable for our tracking techniques.

8.1. Eutectic Media Model

In lamellar eutectics two solid phases grow into a liquid phase [28, 13]. As in [1] we model this process by considering two curves (the solid–liquid interfaces) meeting at a point with a prescribed angle, moving normally with given speed equal to a constant a plus the curvature term from before. As the two curves evolve, the locus of the meeting point traces out a third curve (the solid–solid interface) that should maintain a fixed angle with the solid–liquid interfaces at the meeting point (see Fig. 19). We assume that the solid curves meet the domain boundary normally and that the junction angles are symmetric for simplicity. We consider two curves $x^i(\sigma, t)$ parametrized as before on a fixed interval $[0, 1]$ in σ but here we run the parametrization from left to right for both curves. The governing equations are

$$x_t^i = \frac{x_{\sigma\sigma}^i}{|x_\sigma^i|^2} + a\frac{x_\sigma^{i\perp}}{|x_\sigma^i|}, \tag{24}$$

where $(a, b)^\perp = (-b, a)$. At the junction we have

$$x^1(1, t) = x^2(0, t) := c(t) \tag{25}$$

$$\frac{x_\sigma^1(1, t)}{|x_\sigma^1(1, t)|} \cdot \frac{x_\sigma^2(0, t)}{|x_\sigma^2(0, t)|} = -\frac{1}{2} \tag{26}$$

$$\frac{\dot{c}(t)}{|\dot{c}(t)|} \cdot \frac{x_\sigma^2(0, t)}{|x_\sigma^2(0, t)|} = -\frac{1}{2}. \tag{27}$$

This is a simplified model since it ignores important nonlocal effects on the interface motion. However, the implementation of the junction condition is of interest here and an appropriate method is presented below. A short-time existence result for this problem was proved in [1]. We proceed as in Section 3 and discretize the interior equations with second-order differences and handle the junction by second-order extrapolation as before. The details are interesting enough that we present them below.

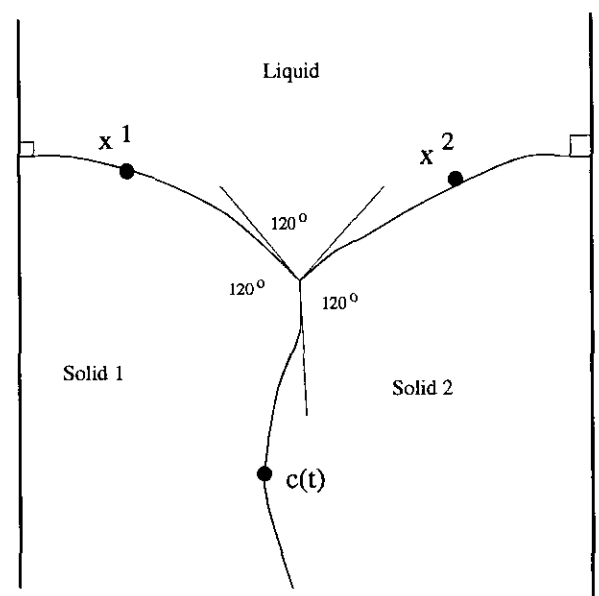

FIG. 19. Eutectic interfaces separating solid 1, solid 2, and liquid phases.

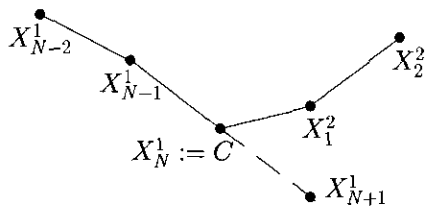

FIG. 20. Details of the discretization near the eutectic junction.

In this case, a regular (i.e., non-staggered) grid is used as in Section 6.2 with N points on each curve with spacing $h = 1/N$. We take $X^1_N := C$ to be an unknown, not to be determined by extrapolation. The value can be used to complete the stencil for the evolution of X^1_1 to satisfy (25). To evolve the point C we need the extrapolated value X^1_{N+1}. This situation is shown in Fig. 20. We can approximate x^2_σ at the junction with one-sided differencing of known points

$$(-\tfrac{1}{2}X^2_2 + 2X^2_1 - \tfrac{3}{2}C)/h.$$

This determines an approximation of direction $\hat{e}_2$ (see Fig. 21) which using (26) and (27) determines $\hat{e}_1$, $\hat{p}_1$, $\hat{p}_3$ and $\hat{e}_3$. Now (26) is approximated using

$$D_1X^1_N = \frac{X^1_{N+1} - X^1_{N-1}}{2h} = \alpha\hat{e}_1$$

or

$$X^1_{N+1} = X^1_{N-1} + 2h\alpha\hat{e}_1, \tag{28}$$

where $\alpha = |D_1X^1_N|$ must be determined. Using (27) we see that

$$\dot{C} = \beta\hat{e}_3, \tag{29}$$

where β must also be determined. Since C is a point on curve 1 its motion can be described by (24). We use a second-order discretization of (24) to approximate $\dot{C}$ in (29) to obtain

$$\frac{D_2X^1_N}{|D_1X^1_N|^2} + a\frac{(D_1X^1_N)^\perp}{|D_1X^1_N|} = \beta\hat{e}_3.$$

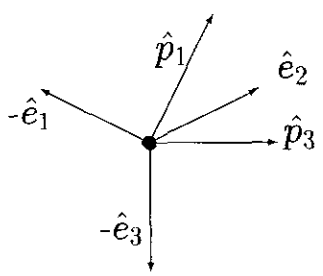

FIG. 21. Unit directions near the eutectic junction: $\hat{e}_1$ is the tangent of curve 1, $\hat{p}_1$ is the normal of curve 1, $\hat{e}_2$ is the tangent of curve 2, $\hat{e}_3$ is the direction of $\dot{C}$, and $\hat{p}_3$ is orthogonal to $\hat{e}_3$.

This can be written

$$\frac{X^1_{N-1} - 2C + X^1_{N+1}}{\alpha^2h^2} + a\hat{p}_1 = \beta\hat{e}_3.$$

We substitute (28) into the above equation and take the inner product with $\hat{p}_3$ to obtain the following quadratic equation for α:

$$\frac{2(X^1_{N-1} - C)\cdot\hat{p}_3}{h^2} + \frac{2}{h}\hat{e}_1\cdot\hat{p}_3\alpha + a\hat{p}_1\cdot\hat{p}_3\alpha^2 = 0.$$

The appropriate root of the above equation (consider a as a perturbation term) can be used in (28) to complete the extrapolation.

The discretization is tested on the problem with initial data

$$x^1(\sigma, 0) = \left(\sigma - 1, \frac{1}{2\sqrt{3}}(1 - (\sigma - 1)^2) + \frac{1}{4}\sin^2(\pi\sigma)\right)$$

$$x^2(\sigma, 0) = \left(\sigma, \frac{1}{2\sqrt{3}}(1 - (1 - \sigma)^2)\right).$$

The initial data and solution at time 1 is shown in Fig. 22 (with the time history of the junction, representing the division between the solids 1 and 2). In "real" eutectics, the interest is in parameter regions where the interface oscillates.

A numerical convergence study is carried out and the results are shown in Table VIII. Second-order convergence is clearly seen. It should be noted that the discretization used for this problem is "lopsided"; i.e., it handles the left and right curves differently at the junction, even though they appear in the same manner in the description of the problem. The authors believe this is necessary to make sense of junction extrapolation for this problem. Naturally, when the roles of the curves are exchanged, the scheme converges to the same solution.

8.2. Nonlocal Model

Consider a collection of disjoint closed curves Γ_i in the plane. Under curvature motion they will tend to circles that will shrink to points in finite time [9, 11]. However, if the normal velocity

TABLE VIII

Estimated Errors and Convergence Rates for Eutectic Computations

$N = 1/h$	e_h	ρ_h
8	0.6542e-3	
16	0.2471e-3	1.38
32	0.7749e-4	1.67
64	0.2157e-4	1.85
128	0.5705e-5	1.92

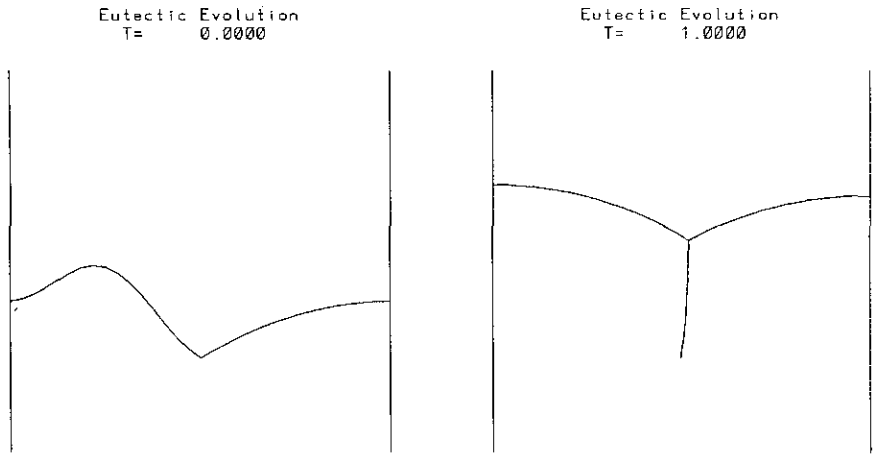

FIG. 22. Eutectic media example computation at $t = 0$ and $t = 1$.

is modified from the curvature κ to $\kappa - \kappa_a$, where κ_a is the average curvature defined below, then the total area enclosed by the curves remains constant. This type of motion is obtained as a limit of a nonlocal model of binary alloys [19, 2]. The length of a curve Γ_i is denoted $|\Gamma_i|$. Let $|\Gamma|$ be the total length. The average curvature is defined as

$$\kappa_a = \frac{1}{|\Gamma|} \sum_i \int_{\Gamma_i} \kappa \, ds.$$

A discrete scheme to model this problem is set up as before, using approximations to $x^i(\sigma, t) = (u^i, v^i)$ that parametrize the curve Γ^i for $\sigma \in [0, 1]$. We use the approximations

$$|\Gamma_i| \approx h \sum_{j=1}^{N} |D_1 X_j^i| \tag{30}$$

$$\int_{\Gamma_i} \kappa \, ds \approx h \sum_{j=1}^{N} \frac{D_2 X_j^i}{|D_1 X_j^i|^2} \cdot (D_1 X_j^i)^{\perp} \tag{31}$$

$$A_i \approx -h \sum_{j=1}^{N} D_+ U_j^i \mathcal{F} V_j^i, \tag{32}$$

where $h = 1/N$ and A_i is the area enclosed by the curve Γ_i. Note that the above formulas are trapezoidal rule approximations (using finite difference approximations of derivatives) of the corresponding integrals transformed to a σ parametrization. *Equation (32) also represents the polygonal area of the discrete curve.*

When the approximations (30) and (31) are used to approximate the average curvature κ_a the discrete equations become

$$\dot{X}_j^i = \frac{D_2 X_j^i}{|D_1 X_j^i|^2} - \kappa_a \frac{(D_1 X_j^i)^{\perp}}{|D_1 X_j^i|}.$$

An example calculation with two curves is shown in Fig. 23. One of the areas grows at the expense of the other. The approximate total area for this computation is shown in Fig. 24 we see that discrete area is kept approximately constant. In fact, at the semi-discrete level, the above discretization exactly preserves discrete area as defined above (this can be shown by direct substitution and some summation by parts). The jumps in Fig. 24 are due to regridding.

Unlike the case of simple curvature motion, curves moving with the nonlocal term can cross or self-intersect (see [2] for an explicit example). Tracking methods in this case become less useful, because constant checking for changing topology is needed and when this is detected some nontrivial surgery must be done. Other authors are working on extending the more appropriate level set methods to this case [21]. However, questions about the existence and location of steady-state solutions of a single curve in bounded domains (where the area conservation is preserved if the curves meet the boundary nor-

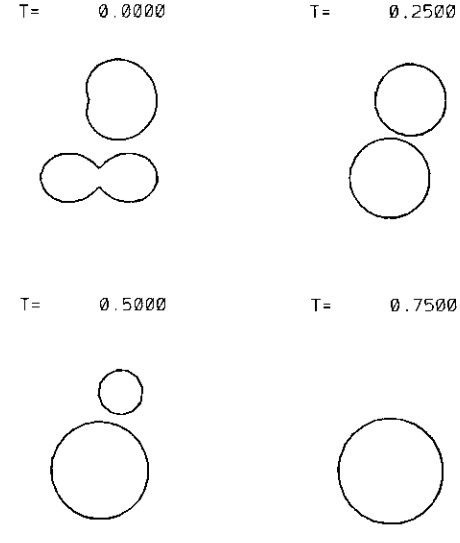

FIG. 23. Nonlocal model calculations.

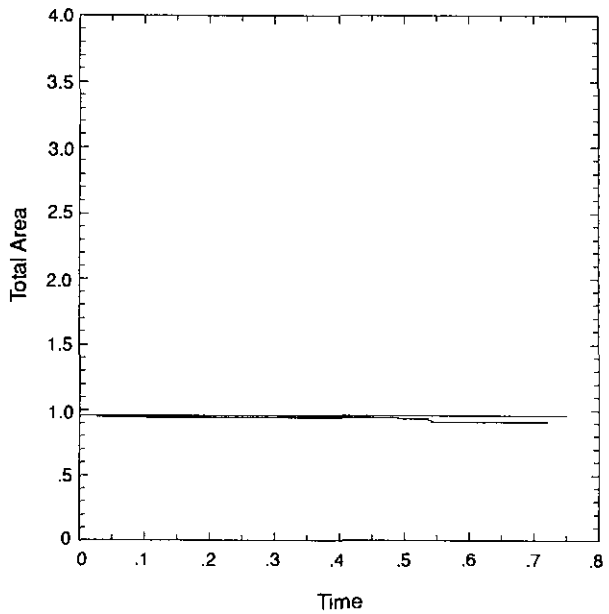

FIG. 24. Discrete area for nonlocal calculations.

mally) can be more easily addressed with this tracking technique. This is an area of current research.

9. SUMMARY

A numerical method is presented for tracking curve networks moving with curvature motion and meeting at several types of junctions, the principal example being triple junctions with prescribed angles. The evolution is uniquely determined until one or more curves shrink to zero length (singularity). At this time, surgery of the curve network can be performed based on physical ideas from the particular phase model under consideration to continue the computation. We present lists of generic singularities that we conjecture are complete.

The method is shown to be linearly stable and second-order accurate, with some minimal loss of accuracy at times of singularity and surgery. We compare our spatial discretization to others and find that it is more accurate. We also present two methods to implement implicit time stepping that are more efficient than those currently used.

Finally we extend our numerical method to curve networks connected with eutectic junctions as well as curves that interact through nonlocal area preserving effects.

Further work suggested by the results of this paper includes an investigation of the singularities in curve networks. What is the structure of the solution at singularity in our parametrization? Can one prove that our list of generic singularities is complete? Does the curve reconstruction proposed here correspond to the asymptotic limit of the continuous reaction–diffusion model? These are some of the questions that should be answered. In addition, the more efficient time-stepping techniques proposed here may make statistical studies of the type undertaken in [8] with a larger number of grains feasible.

APPENDIX: LINEAR STABILITY

We consider the frozen coefficient linear stability of our semi-discretization method applied to a network of three curves with a triple junction (for simplicity we consider symmetric angles of 120° at the junction). The linearized boundary conditions for the continuous problem satisfy the appropriate determinant condition that guarantees a unique, smooth solution [23, 1]. For our type of discretization the boundary conditions must also satisfy another determinant condition that we describe below. General sufficient conditions for stability of fully discrete finite difference methods applied to constant coefficient linear problems of this type are given in [27] and these results can be used to show the stability of our discretization. We present an alternate technique below.

Structure of the Problem

The equation corresponding to linear perturbations $\tilde{x}^i$ of a solution x^i of curvature motion (1) is

$$\tilde{x}^i_t = \frac{\tilde{x}^i_{\sigma\sigma}}{|x^i_\sigma|^2} - \frac{2x^i_{\sigma\sigma}x^i_s \cdot \tilde{x}^i_\sigma}{|x^i_\sigma|^4}.$$

This equation is of convection–diffusion type and matches the linearized discrete implicit equations from Section 7.1. We consider a six-tuple vector u to describe the two components of the three curves of the linearized version of (1),

$$u_t = A(\sigma, t)u_{\sigma\sigma} + C(\sigma, t)u_\sigma + f, \tag{33}$$

where A is a diagonal matrix with positive entries

$$|x^1_\sigma|^{-2}, |x^1_\sigma|^{-2}, |x^2_\sigma|^{-2}, |x^2_\sigma|^{-2}, |x^3_\sigma|^{-2}, |x^3_\sigma|^{-2}.$$

At the junction, which we take to be at $\sigma = 0$ now, the linearized versions of (4)–(6) are $B_0u(0) = g^{\mathrm{II}}$ and $B_1u_\sigma(0) = g^{\mathrm{I}}$, where the notation of Varah [27] is used. The g's are data for the boundary conditions. B_0 has rank 4 and corresponds to the Dirichlet part of the boundary conditions and B_1 has rank 2 and corresponds to the Neumann part of the boundary conditions. The explicit forms are

$$B_0 := \begin{pmatrix} I & -I & 0 \\ 0 & I & -I \end{pmatrix}$$

$$B_1 := \begin{pmatrix} 0 & 2|x^2_\sigma|\sqrt{3} & 3|x^1_\sigma| & \sqrt{3}|x^1_\sigma| & 0 & 0 \\ 0 & 0 & -3|x^3_\sigma| & -\sqrt{3}|x^3_\sigma| & -3|x^2_\sigma| & -\sqrt{3}|x^2_\sigma| \end{pmatrix},$$

where I above denotes the 2 by 2 identity matrix and the form of B_1 is that obtained after suitable coordinate rotation at the triple junction.

As is typical when considering this type of problem or a discretization of it, we consider first only the highest order term in (33) with frozen coefficients from the junction at a fixed time; we further simplify the problem by neglecting the other (domain) boundary and let $\sigma \in [0, \infty)$ (the analysis of the linearized domain boundary conditions are considerably easier than those at the junction). The discretization of this reduced linear problem is

$$\dot{U}_j = AD_2U_j + f_j \tag{34}$$

for $j > 0$ on a staggered grid and with A a constant. The discrete boundary conditions are

$$B_0 \mathcal{F} U_0 = g^{II} \tag{35}$$

and

$$B_1 D^+ U_0 = g^{I}, \tag{36}$$

where B_0 and B_1 are also constant in time. The discrete boundary conditions are used to eliminate U_0 in terms of U_1 (extrapolation). We show the stability of this frozen coefficient linear discrete problem below.

Two Determinant Conditions

In order for the continuous linear problem (33) above to be well-posed, the following determinant condition must be satisfied [23]:

$$\det \begin{pmatrix} B_1 A^{-1/2}(0) \\ B_0 \end{pmatrix} \neq 0. \tag{37}$$

This condition is shown to be satisfied in [1] for our problem, provided the parametrization of the curves is not singular at the junction (i.e., $|x_\sigma| \neq 0$ for any curve) during an existence proof for the nonlinear problem.

A second determinant condition arises from the discretization. If we want to consider the boundary conditions (35), (36) as a way to derive an extrapolation of U_1 to U_0 then

$$\begin{pmatrix} B_0 \\ B_1 \end{pmatrix} U_0 = \begin{pmatrix} -B_0 \\ B_1 \end{pmatrix} U_1 + \begin{pmatrix} 2g^{I} \\ -hg^{II} \end{pmatrix}$$

and the following additional determinant condition must be met:

$$\det \begin{pmatrix} B_0 \\ B_1 \end{pmatrix} \neq 0. \tag{38}$$

An explicit calculation shows that our problem also satisfies this condition (which we suspected since the extrapolation for the nonlinear problem at the triple junction is always well defined). In cases where this condition is not met but the more fundamental previous condition is satisfied, other discretizations of the boundary conditions can be performed [27].

Stability Result

We consider now the stability of the reduced linear model problem above. The stability results are obtained using Laplace transforms in time and l_2 estimates in space (with norm $\|\cdot\|_2$) on suitably "rotated" variables. This is essentially a discrete version of the estimates found in [15]. The "trick" in the discrete setting is to choose the correct combination of time derivatives and discrete space derivatives (see (46)–(49) below). We denote the Laplace transform of $U(t)$ by $\hat{U}(s)$. Note that since U is an "error," $U(0) \equiv 0$ and so $\dot{\hat{U}} = s\hat{U}$.

We consider $s = \eta + i\xi$ with $\eta > 0$ and sufficiently large and h sufficiently small. We will derive the energy estimate on the transform (the proof will be given in a separate section below).

LEMMA 1 (Constant coefficient, linear stability). *The Laplace transform of the discrete solution of* (34)–(36) *with* $U(0) = 0$ *obeys the estimate*

$$\|\hat{U}\|_2^2 \leq K(|\hat{g}|^2 + \|\hat{f}\|_2^2), \tag{39}$$

where K is independent of h and η.

Using a version of the Plancherel theorem and a "future does not affect past" argument in [15] the result of this lemma gives

$$\int_0^T \|U\|_2^2 \leq C(T) \int_0^T (|g|^2 + \|f\|_2^2). \tag{40}$$

This is a weak bound on the solution in terms of the data similar to that proved in [27]. When we consider f and g to be the second-order truncation errors from our discretization, it shows the second-order convergence of our method to the reduced linear problem in this integral sense.

The stability result can be extended to the following additional cases:

1. To $A(\sigma)$ (not necessarily diagonal) where A satisfies $A + A^* \geq \delta I$, where $\delta > 0$. Both determinant conditions (37) and (38) must be satisfied.
2. To the case where lower order terms $C(\sigma)u_\sigma$ and $D(\sigma)u$ are included. The same determinant conditions are required in this case.
3. To problems on a finite spatial domain where there is a second boundary condition which also satisfies the determinant condition (this is true of the linearized domain boundary condition).
4. To the case where lower order Dirichlet terms are included in the Neumann boundary conditions.

However, the extension to time-dependent coefficients and the full nonlinear problem has not been done (this is also not done

in [27]). A straightforward technique of dominating the problem with time-dependent coefficients locally with the time-independent problem (as is done in [23]) is not possible because the estimates here and in [27] are too weak. We remark that with more work the estimates in (40) can be improved to estimates on differences of the error U. In future work we plan to prove a short-time convergence result for the nonlinear problem using these estimates combined with the asymptotic error analysis of Strang [25].

Proof of Lemma 1. To ease the details of the proof we take $A = I$. The more general case can be obtained by modifying the arguments below in essentially the same manner as done in [15]. We define

$$\rho = \sqrt{s}$$

and

$$\gamma = \sqrt{1 + \rho^2 h^2/4}$$

so that the argument of the above quantities has absolute value less than $\pi/4$. The following lemma of simple relationships between γ and ρ is stated without proof:

LEMMA 2. *The following estimates are valid for all s with* $\operatorname{Re} s > 1$ *and* $h < 1$:

$$\frac{h}{2} < \left|\frac{\gamma}{\rho}\right| < \frac{h}{2} + \frac{1}{|\rho|} < \frac{3}{2} \tag{41}$$

$$\operatorname{Re}(\gamma^*\rho) > \frac{|\gamma|\,|\rho|}{2}. \tag{42}$$

After Laplace transformation, the discrete equations (34) become

$$\rho^2 \hat{U} = D_2 \hat{U} + \hat{f}. \tag{43}$$

Some useful facts relating difference operators are listed below for reference,

$$\begin{aligned} \mathcal{B}\mathcal{F}\hat{U}_i &= \hat{U}_i + \frac{h^2}{4} D_2 \hat{U}_i \\ &= \gamma^2 \hat{U}_i - \frac{h^2}{4}\hat{f}_i \end{aligned} \tag{44}$$

$$\mathcal{B} D_+ \hat{U}_i = D_1 \hat{U}_i, \tag{45}$$

where in (44) the discrete equation (43) has been used. Here, $\mathcal{B}$ and D_- denote backward averaging and differencing, respectively.

We define the following quantities:

$$w_i^0 = \mathcal{F}\hat{U}_i - \frac{\gamma}{\rho} D_+ \hat{U}_i \tag{46}$$

$$w_i^1 = \mathcal{F}\hat{U}_i + \frac{\gamma}{\rho} D_+ \hat{U}_i \tag{47}$$

$$\alpha_i^0 = \hat{U}_i - \frac{\gamma}{\rho} D_1 \hat{U}_i \tag{48}$$

$$\alpha_i^1 = \hat{U}_i + \frac{\gamma}{\rho} D_1 \hat{U}_i. \tag{49}$$

The following results follow from the definitions of w and α and (44), (45):

$$\mathcal{B} w_i^0 = \gamma^2 \alpha_i^0 - \frac{h^2}{4}\hat{f}_i \tag{50}$$

$$D_- w_i^0 = -\gamma\rho\alpha_i^0 + \frac{\gamma}{\rho}\hat{f}_i \tag{51}$$

$$\mathcal{B} w_i^1 = \gamma^2 \alpha_i^1 - \frac{h^2}{4}\hat{f}_i \tag{52}$$

$$D_- w_i^1 = \rho\alpha_i^1 - \frac{\gamma}{\rho}\hat{f}_i. \tag{53}$$

We define the following sums $\mathcal{I}$ and $\mathcal{J}$ of inner products in l_2:

$$\mathcal{I} := (\mathcal{B}w^0, D_- w^0) + (D_- w^0, \mathcal{B}w^0)$$

$$\mathcal{J} := -(\mathcal{B}w^1, D_- w^1) - (D_- w^1, \mathcal{B}w^1).$$

The first estimates from these sums will be derived from the four relationships (50)–(53),

$$\mathcal{I} \le -\operatorname{Re}(\gamma^*\rho)|\gamma|^2\|\alpha^0\|^2 + \left(\frac{|\gamma|^3}{|\rho|} + \frac{|\gamma\rho|h^2}{4}\right)\|\hat{f}\|\,\|\alpha^0\|$$

which after the use of (41) and (42) and then appropriate estimates of the cross terms ($2ab \le \varepsilon a^2 + \varepsilon^{-1} b^2$) becomes

$$\mathcal{I} \le -C_1\gamma^3\rho\|\alpha^0\|^2 + C_2\frac{\gamma^3}{\rho^3}\|\hat{f}\|^2. \tag{54}$$

Here we use shorthand ρ for $|\rho|$ and γ for $|\gamma|$. The constants C_1 and C_2 are independent of h and s, assuming $\operatorname{Re} s > 1$ and $h < 1$. Similarly,

$$\mathcal{J} \le -C_1\gamma^3\rho\|\alpha^1\|^2 + C_2\frac{\gamma^3}{\rho^3}\|\hat{f}\|^2. \tag{55}$$

We now use simple summation by parts to get estimates of $\mathcal{I}$ and $\mathcal{J}$ from the boundaries:

$$(\mathcal{B}w, D-w) = \tfrac{1}{2}\sum_{i=1}(|w_i|^2 - |w_{i-1}|^2) + \tfrac{1}{2}\sum_{i=1}(w_i^* w_{i-1} - w_{i-1}^* w_i). \tag{56}$$

Assuming a semi-infinite spatial domain with appropriate decay in the solution as x goes to ∞, this formula can be combined with the complex conjugate term as in $\mathcal{J}$ to give

$$\mathcal{J} = -|w_0^0|^2 \tag{57}$$

and, similarly,

$$\mathcal{J} = |w_0^1|^2. \tag{58}$$

In terms of w the boundary conditions (35) and (36) become

$$\begin{pmatrix} -B_1 \\ B_0 \end{pmatrix} w_0^0 + \begin{pmatrix} B_1 \\ B_0 \end{pmatrix} w_0^1 = \begin{pmatrix} \frac{2\gamma}{\rho} g^I \\ 2g^{II} \end{pmatrix} \tag{59}$$

or

$$w_0^0 = \begin{pmatrix} B_1 \\ -B_0 \end{pmatrix}^{-1} \left\{ \begin{pmatrix} B_1 \\ B_0 \end{pmatrix}^{-1} w_0^1 - \begin{pmatrix} \frac{2\gamma}{\rho} g^I \\ 2g^{II} \end{pmatrix} \right\}.$$

It is at this step that the determinant condition (37) which in this case of $A = I$ coincides with (38) is needed. We use (41) to show that the factor γ/ρ can be neglected in (59) and so we obtain

$$|w_0^0|^2 \le c|w_0^1|^2 + C|\hat{g}|^2. \tag{60}$$

We combine (54), (55), (57), (58), and (60) to get

$$|w_0|^2 - C|\hat{g}|^2 \le \mathcal{J} + d\mathcal{J} \le -C_1\gamma^3\rho\|\alpha\|^2 + C_2\frac{\gamma^3}{\rho^3}\|\hat{f}\|^2, \tag{61}$$

where $d > 1 + 2c$ and $\|\alpha\|^2 = \|\alpha^0\|^2 + \|\alpha^1\|^2$. We note that $2\hat{U} = \alpha^0 + \alpha^1$ and we can use the estimate (61) to obtain

$$\|\hat{U}\|^2 \le \frac{K}{\rho\gamma^3}|\hat{g}|^2 + \frac{K}{\gamma^4}\|\hat{f}\|^2.$$

Since $\rho > 1$ and $\gamma > 1$ by assumption on s we obtain the result of the stability lemma (39)

ACKNOWLEDGMENTS

We thank Chris Anderson, David Kirkpatrick, Stan Osher, and Fernando Reitich for helpful discussions.

REFERENCES

1. L. Bronsard and F. Reitich, *Arch. Rat. Mech.* **124,** 355 (1993).
2. L. Bronsard and B. Stoth, Centre for Nonlinear Analysis Research Report No. 94-NA-008, Carnegie Mellon University, 1994 (unpublished).
3. J. Cahn, *J. Chem. Phys.* **66,** 3667 (1977).
4. J. Cahn, C. Handwerder, and J. Taylor, preprint.
5. E. Ceppi and O. Nasello, *Scripta Metall.* **18,** 1221 (1984).
6. V. Fradkov, L. Shvindlerman, and D. Udler, *Phil. Mag. Lett.* **55,** 289 (1987).
7. V. E. Fradkov, M. E. Glicksman, M. Palmer, J. Nordberg, and K. Rajan, *Physica D* **66,** 50 (1993).
8. H. Frost, C. Thompson, C. Howe, and J. Whang, *Scripta Metall.* **22,** 65 (1988).
9. M. Gage and R. S. Hamilton, *J. Differential Geom.,* **23,** 69 (1986).
10. J. Glazier, M. Anderson, and G. Grest, *Philos. Mag. B* **62** (6), 615 (1990).
11. M. Grayson, *J. Differential Geom.* **26,** 285 (1987).
12. W. Hackbusch, *Multi-Grid Methods and Applications,* (Springer Verlag, Berlin, 1985).
13. A. Karma, "Computer Modeling of Eutectic Growth," in *Solidification Processing of Eutectic Alloys* (D. Stefanescu, G. Abbaschian, and R. Bayuzick, Eds.), The Metallurgical Society, 1988.
14. K. Kawasaki, T. Nagai and K. Nakashima, *Phil. Mag. B* **60,** 399 (1989).
15. H.-O. Kreiss and J. Lorenz, *Initial-Boundary Value Problems and the Navier–Stokes Equations* (Academic Press, San Diego, CA, 1989), Chap. 7.
16. D. Michaelson, *Math. Comput.* **49,** 445 (1987).
17. B. Merriman, J. Bence, and S. Osher, preprint.
18. W. Mullins, *J. Appl. Phys.* **27** (8), 900 (1956).
19. J. Rubinstein and P. Sternberg, *IMA J. Appl. Math.* **48,** 248 (1992).
20. J. Sethian and J. Strain, *J. Comput. Phys.* **98,** 231 (1992).
21. P. Smereka, private communication.
22. C. Smith, *Grains, Trans. Am. Inst. Mining Metall. Eng.* **175,** 15 (1948).
23. Solonnikov, *Proc. Steklov Inst.* **83** (1965).
24. J. Strain, *J. Comput. Phys.* **85,** 342 (1989).
25. G. Strang, *Numer. Math.* **6,** 37 (1964).
26. J. Strikwerde, *Finite Difference Schemes and Partial Differential Equations* (Wadsworth, Pacific Grove, 1989), p. 78.
27. J. M. Varah, *SINUM* **8,** 598 (1971).
28. D. Woodruff, *The Solid–Liquid Interface* (Cambridge Univ. Press, Cambridge, 1973).