Crystal Growth and Dendritic Solidification

JAMES A. SETHIAN*

Department of Mathematics, University of California, Berkeley, California 94720

AND

JOHN STRAIN[†]

*Courant Institute of Mathematical Sciences, New York University,
251 Mercer Street, New York, New York 10012*

We present a numerical method which computes the motion of complex solid/liquid boundaries in crystal growth. The model we solve includes physical effects such as crystalline anisotropy, surface tension, molecular kinetics, and undercooling. The method is based on two ideas. First, the equations of motion are recast as a single history-dependent boundary integral equation on the solid/liquid boundary. A fast algorithm is used to solve the integral equation efficiently. Second, the boundary is moved by solving a "Hamilton–Jacobi"-type equation (on a *fixed* domain) formulated by Osher and Sethian for a function in which the boundary is a particular level set. This equation is solved by finite difference schemes borrowed from the technology of hyperbolic conservation laws. The two ideas are combined by constructing a smooth extension of the normal velocity off the moving boundary, in a way suggested by the physics of the problem. Our numerical experiments show the evolution of complex crystalline shapes, development of large spikes and corners, dendrite formation and side-branching, and pieces of solid merging and breaking off freely.   © 1992 Academic Press, Inc.

The moving boundary problems of crystal growth and unstable solidification have attracted considerable interest over the last few years. An outstanding problem of current research is to find the time-dependent shape of a crystal of a pure substance, growing from an undercooled bath of its liquid phase. The goal is to understand the role played by such physical parameters as the crystalline anisotropy, surface tension and molecular kinetics of the material, the undercooling imposed at the container walls, and the initial temperature distribution, and the solid shape. A good

review of the physics may be found in Langer [19] and Chalmers [6], and detailed analysis may be found in many papers, including Caginalp and Fife [3], Gurtin [13], Mullins and Serkerka [25, 26], Nash and Glicksman [27, 28], Laxmanan [21], Verdi and Visintin [44], Wollkind and Notestine [45], Cahn and Hilliard [4], Duchon and Robert [8].

The physical motivation for this problem is as follows. Begin with a container of the liquid phase of the material under study, water for example. Suppose the box is smoothly and uniformly cooled to a temperature below its freezing point, so carefully that the liquid does not freeze. The system is now in a "metastable" state, where a small disturbance—such as dropping a tiny seed of the solid phase into the liquid—will initiate a rapid and unstable process known as *dendritic solidification*. The solid phase will grow from the seed by sending out branching fingers into the distant cooler liquid nearer the undercooled wall. This growth process is unstable in the sense that small perturbations of the initial data can produce large changes in the time-dependent solid/liquid boundary.

Mathematically, this phenomenon can be modeled by a moving boundary problem. The temperature field satisfies a heat equation in each phase, coupled through two boundary conditons on the unknown moving solid/liquid boundary, as well as initial and boundary conditions. Derivations are given by Langer [19], Gurtin [13], Caginalp and Fife [3], and much asymptotic analysis has been done (see Langer [19], Chadam and Ortoleva [5], Strain [40], Kessler and Levine [17], Ben Amar and Pomeau [1], Caginalp and Fife [3], Fix [11], Langer [18], Pinus and Taylor [31]), but no rigorous mathematical existence theory is availabe at present. This is because the moving boundary conditions explicitly involve geometric properties of the boundary itself, such as the local curvature and the normal direction, as well as the temperature field. Hence the

transformation into a fixed domain, which has been used to analyze the classical Stefan problem for melting, breaks down. Even asymptotic calculations are difficult, because surface tension constitutes a singular perturbation of the Stefan problem.

Models for crystal growth have also been studied through numerical calculations. One approach is to solve the heat equation numerically in each phase and to try to move the boundary so that the two boundary conditions are satisfied (see Chorin [7], Smith [37], Kelly and Ungar [16], Meyer [23], Sullivan et al. [43]). However, it is difficult to impose the boundary conditions accurately on a time-dependent and complicated boundary. Thus these calculations have been used mainly to study small perturbations of smooth crystal shapes. Another approach is to recast the equations of motion as a single integral equation on the moving boundary and solve the integral equation numerically, as is done in Meiron [22], Strain [38], Karma [15], Kessler and Levine [17], Langer [19]. This approach can yield more accurate results for smooth boundaries, as well as agreement with linear stability theory. On the other hand, a parametrization of the boundary must be computed at each time step. The curvature and normal vector are then derivatives of the parametrization, and these methods usually break down in the interesting cases where the boundary becomes complicated and loses smoothness, as discussed in Sethian [32, 33]. In particular, the calculations presented in Strain [38] indicate that corners and cusps may form and pieces of the boundary may intersect; in either case, the boundary cannot be parametrized by a single smooth function. Finally, we note that most known numerical methods for crystal growth are difficult to extend to problems in three space dimensions.

In this paper, we present a numerical method for crystal growth problems which avoids these difficulties and computes complex crystal shapes. Our method follows growth from an initial seed crystal or crystals (of arbitrary shape, size, and location) in a standard model which includes the crystalline anisotropy, surface tension and molecular kinetics of the material, the undercooling imposed on the container walls, and the initial state of the crystal/temperature system. Our numerical calculations exhibit complicated shapes with spikes and corners, topological changes in the solid-liquid boundary, dendrite formation, and sidebranching.

The method relies on two main ideas. First, we represent the solid/liquid boundary as the zero set of a function ϕ defined on the whole container. The boundary is then moved by solving a nonlinear pseudo-differential equation of suggestive of a Hamilton–Jacobi on the whole container. This level set "Hamilton–Jacobi" formulation of moving interfaces was introduced in Osher and Sethian [30] and allows us to compute geometric properties of highly complicated boundaries without relying on a parametrization.

Hence, the moving boundary can develop corners and cusps and undergo topological changes quite naturally. Second, we reformulate the equations of motion as a boundary integral equation for the normal velocity as is also done in Langer [19], Meiron [22], Strain [38], Kessler and Levine [17], Brush and Sekerka [2]. We then extend the normal velocity smoothly to the whole container, as required by the level set Osher–Sethian algorithm. These two ideas combine to yield a fixed-domain formulation of the equations of motion, which may be of analytical interest itself. Our numerical method is based on solving a regularization of this equation, with entropy-satisfying upwind differencing for the level set equation of motion and a fast algorithm for evaluating the normal velocity. The method generalizes immediately to higher order accurate schemes and, more importantly, to three-dimensional problems. This generalization will be presented elsewhere (Sethian and Strain [36]).

The paper is organized as follows. Section 1 reviews the equations of motion and some of the physical background. Section 2 gives an overview of the numerical method, and Section 3 gives a flow chart. Section 4 describes the level set equation for moving boundaries. Section 5 discusses a fast algorithm for applying the integral operator which occurs in the boundary integral equation. Section 6 contains some of the many details of the numerical method, and Section 7 presents the numerical examples. Our conclusions are discussed in Section 8.

In the numerical examples, we compute the motion of a growing crystal under a variety of conditions. We begin with calculations of crystal motion under purely geometric effects and demonstrate the effects of crystalline anisotropy on the evolving crystal. We then study a crystal growing from a perturbed circular seed. For one set of values for the kinetic coefficient, surface tension, and latent heat of solidification, the resulting motion generates large smooth limbs. With another set of physical parameters, our model produces intricate shapes which exhibit fingering, tip splitting, and side branching. In all cases, these crystalline boundaries, even when highly complex, remain virtually unchanged under refinement of the numerical parameters, demonstrating the robustness of our numerical algorithm. In addition, we change physical parameters smoothly to show the evolution of tip-splitting and fingering. Finally, we study the interaction of an initial array of crystals, showing the complex topological changes that result from various choices of the physical parameters.

1. THE MODEL EQUATIONS OF MOTION

We wish to model the growth of a solid crystal from a seed in a undercooled liquid bath. We consider a model which includes the following physical effects:

(1) *Undercooling.* The sub-freezing temperature of the walls and the initial liquid state. This is the driving force for the growth of the solid.

(2) *Crystalline anisotropy.* The tendency of the crystal to grow along certain axes of symmetry. The selection of a particular symmetry is due to the microscopic structure of the material; we treat it in a standard phenomenological way in this macroscopic model.

(3) *Surface tension.* A curvature-dependence of the boundary temperature which regularizes the problem by smoothing corners and preventing excessive stretching of the solid/liquid boundary.

(4) *Molecular kinetics.* A velocity-dependence of the boundary temperature, which imposes an energy penalty for excessively fast motion of the boundary.

(5) *Initial conditions.* The shape, size, location, and initial growth velocity of the initial seed crystal or crystals, and the initial temperature distribution in the liquid and solid.

We now state the equations of motion. Consider a square container, $B = [0, 1] \times [0, 1]$, filled with the liquid and solid phases of some pure substance. The unknowns are the temperature $u(x, t)$ for $x \in B$, and the solid-liquid boundary $\Gamma(t)$.

The temperature field u is taken to satisfy the heat equation in each phase (see Fig. 1), together with an initial condition in B and boundary conditions on the container walls. Thus

$$
\begin{aligned}
u_t &= \Delta u & &\text{in } B \text{ off } \Gamma(t) \\
u(x, t) &= u_0(x) & &\text{in } B \text{ at } t = 0 \qquad (1.1) \\
u(x, t) &= u_B(x) & &\text{for } x \in \partial B.
\end{aligned}
$$

Since the position of the moving boundary $\Gamma(t)$ is unknown, two boundary conditions on $\Gamma(t)$ are required to determine u and $\Gamma(t)$. Let n be the outward normal to the boundary, pointing from solid to liquid. The first boundary condition is the classical Stefan condition:

$$
[\partial u / \partial n] = -HV \qquad \text{on } \Gamma(t). \qquad (1.2)
$$

Here $[\partial u / \partial n]$ is the jump in the normal component of heat flux $\partial u / \partial n$ from solid to liquid across $\Gamma(t)$, V is the normal velocity of $\Gamma(t)$, taken positive if liquid is freezing, and H is the dimensionless latent heat of solidification, which is a constant. The signs of geometric quantities are chosen so that if $\partial u / \partial n < 0$ in the liquid phase and $\partial u / \partial n = 0$ in the solid phase, then $[\partial u / \partial n]$ is negative and $V > 0$, indicating that the solid phase is growing. Physically, this means that undercooling drives solid growth. The latent heat of solidification controls the balance between geometry and
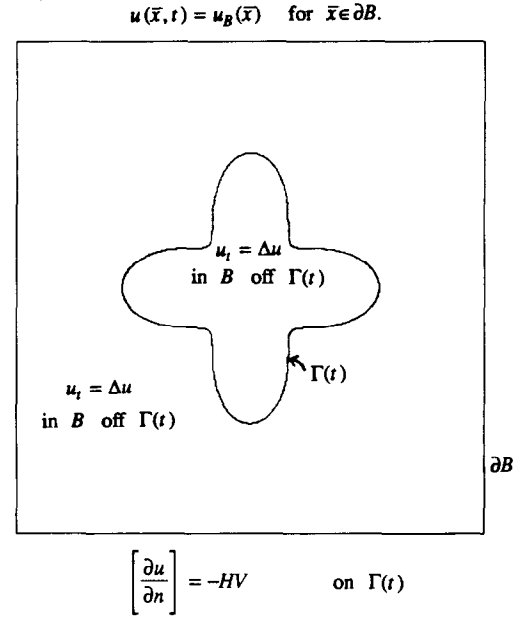
FIG. 1. The equations of motion and physical domain.

temperature effects; setting $H = 0$ reduces the crystal motion to pure geometry.

The second boundary condition on $\Gamma(t)$ is the classical Gibbs–Thomson relation, modified to include crystalline anisotropy and molecular kinetics as well as the surface tension:

$$
u(x, t) = -\varepsilon_C(n)C - \varepsilon_V(n)V
$$
$$
\text{for } x \text{ on } \Gamma(t). \qquad (1.3)
$$

Here C is the curvature at x on $\Gamma(t)$, taken positive if the center of the osculating circle lies in the solid. The anisotropy functions are modeled by

$$
\varepsilon_C(n) = \varepsilon_C(1 - A \cos(k_A \theta + \theta_0)) \qquad (1.4)
$$

$$
\varepsilon_V(n) = \varepsilon_V(1 - A \cos(k_A \theta + \theta_0)), \qquad (1.5)
$$

where θ is the angle between n and the x-axis, and ε_C, ε_V, A, k_A, and θ_0 are constants depending on the material and the experimental arrangement. For example, if $\varepsilon_C = 0$ ($\varepsilon_V = 0$), there are no surface tension (molecular kinetic) effects. For $A = 0$, the system is isotropic, while if $A > 0$, the solid is k_A-fold symmetric with a symmetry axis at angle θ_0 to the x-axis. Typically $A \le 1$. The equations of motion under study are (1.1), (1.2), and (1.3), with anisotropy functions given by (1.4) and (1.5).

For appropriate initial data u_0 and boundary data u_B, an

exact solution exists in which $\Gamma(t)$ is a line and u is an exponentially decaying traveling wave

$$\Gamma(t) = \{ y = Vt,\ x \in (0,\ 1) \} \tag{1.7}$$

$$u(x,\ y,\ t) = \begin{cases} 0, & y < Vt \\ e^{-V(y-Vt)} - 1 - \varepsilon_V V, & y \geq Vt \end{cases}, \tag{1.8}$$

where V is a positive constant and $\varepsilon_V = \varepsilon_V(0)$. Linear stability of this solution has been examined in Langer [19], Chadam and Ortoleva [5], Pinus and Taylor [33], Mullins and Sekerka [26], Strain [41]. If $\varepsilon_C = \varepsilon_V = 0$, then this solution is linearly unstable with a disturbance $\cos(kx)$ growing like $e^{V|k|t}$. If $\varepsilon_C > 0$, only a finite band of low wavenumbers are unstable; the high-k instability is damped by the introduction of surface tension. Making $\varepsilon_V > 0$ does not change the picture qualitatively.

More general models have been widely discussed in the literature. The assumption of equal heat diffusion coefficients in solid and liquid, for example, is roughly valid only for certain plastic crystals like succinonitrile, and certainly highly invalid for the ice/water system. It is also interesting to treat problems in which the domain is infinite; this removes the effects of the finite container size and makes it possible to compare numerical results with some classical similarity solutions of the Stefan problem, for example. The results presented in this paper are not directly comparable with known exact solutions like expanding circles, so our only check on the method will be convergence under mesh refinement. However, our method can be extended to treat these more general problems; we explain it in this case mainly to simplify the explanation of what is already a rather complicated algorithm.

2. OUTLINE OF THE NUMERICAL ALGORITHM

In this section, we sketch the numerical method we use to solve the equations of motion. We present the algorithm in four steps. In the first two steps, we transform the equations of motion into a boundary integral formulation. In the last two, we describe the level set formulation for moving the boundary and the necessary extension of V. The details of each step may be found in the sections that follow. However, the fundamental philosophy behind the algorithm is most easily conveyed through a series of figures.

In Fig. 2a, we show a typical solid/liquid boundary. According to Eqs. (1.1)–(1.6), the temperature field u must satisfy the heat equation in each phase, as well as two boundary conditions on the phase boundary. In Steps 1 and 2, we transform the equations of motion into a single boundary integral equation on the moving boundary (see Fig. 2b). Thus, we transform equations of motion which require computation of the temperature field u on the whole
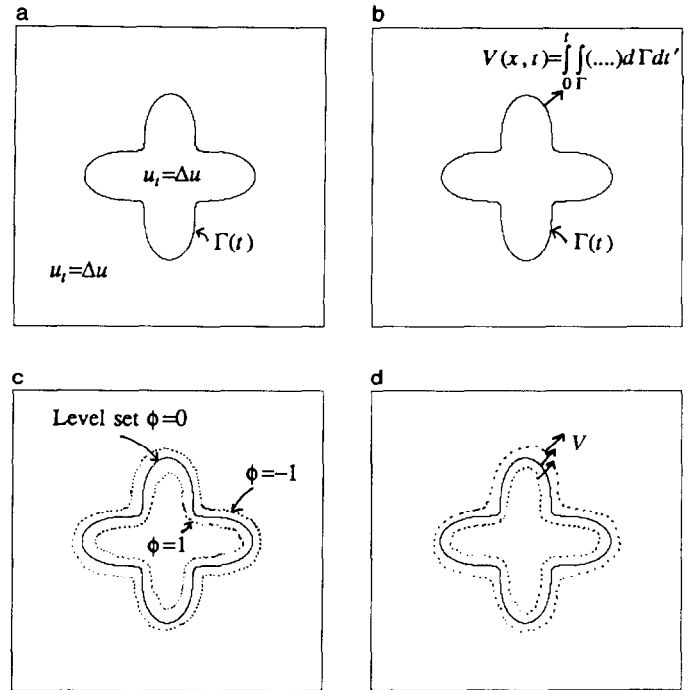

FIG. 2. The flow of the algorithm: (a) position of solid liquid boundary in B; (b) boundary integral formulation; (c) level set formulation; (d) extension of velocity field to level set formulation.

domain into an equation which involves only the moving boundary and its previous history.

In Step 3, we represent the boundary $\Gamma(t)$ as the level set $\{\phi = 0\}$ of a function ϕ defined on all of B (see Fig. 2c). We construct a nonlinear pseudodifferential equation which evolves ϕ in such a way that the zero set $\{\phi = 0\}$, at each time t, is the moving boundary $\Gamma(t)$ (see Fig. 2d). To do this, in Step 4 we construct a "speed function" F, which is equal to the normal velocity V on $\Gamma(t)$ and smoothly extends V to all to B. We then move all the level sets of ϕ with normal velocity F. The resulting level set equation for ϕ, which is reminiscent of a Hamilton–Jacobi equation, may be solved numerically by finite difference schemes borrowed from hyperbolic conservation laws. A major advantage of this formulation is that the equation for ϕ can be solved on a uniform mesh on the box B; the level sets are moved without constructing them explicitly.

We now describe the algorithm in more detail.

Step 1. Subtraction of Initial and Boundary Conditions

Recall the equations of motion (1.1)–(1.6). The temperature field u and moving solid/liquid boundary $\Gamma(t)$ satisfy

$$u_t = \Delta u \quad \text{in} \quad B - \Gamma(t)$$

$$u(x,\ t) = u_0(x) \quad \text{for} \quad t = 0, \tag{2.1}$$

$$u(x, t) = u_B(x) \qquad \text{for} \quad x \in \partial B$$

$$[\partial u/\partial n] = -HV \qquad \text{on} \quad \Gamma(t) \tag{2.2}$$

$$u(x, t) = -\varepsilon_C(n)C - \varepsilon_V(n)V$$

$$\text{for} \quad x \text{ on } \Gamma(t). \tag{2.3}$$

First, we subtract the temperature field due to the initial conditions u_0 and boundary conditions u_B. Let $U(x, t)$ be the solution to the heat equation

$$U_t = \Delta U \qquad \text{in } B$$

$$U(x, 0) = u_0(x) \qquad \text{at} \quad t = 0 \tag{2.4}$$

$$U(x, t) = u_B(x, t) \qquad \text{for} \quad x \in \partial B \text{ and } t > 0.$$

Let $W = u - U$. Then W satisfies

$$W_t = \Delta W \qquad \text{in } B - \Gamma(t)$$

$$W(x, t) = 0 \qquad \text{at} \quad t = 0, \tag{2.5}$$

$$W(x, t) = 0 \qquad \text{for} \quad x \in \partial B$$

$$[\partial W/\partial n] = -HV \qquad \text{on} \quad \Gamma(t) \tag{2.6}$$

$$W(x, t) = -\varepsilon_C(n)C - \varepsilon_V(n)V$$

$$- U(x, t) \qquad \text{for} \quad x \text{ on } \Gamma(t). \tag{2.7}$$

Equations (2.5)–(2.7) are equivalent to the original equations of motion. Given the solution U, at any time we can add W to U to produce the solution u to the original problem. The "free" temperature field $U(x, t)$, which is defined on a fixed domain, may be found analytically for simple initial and boundary conditions, or by numerical calculation in practical situations.

Step 2. Transformation to a Boundary Integral Equation

The next step is to transform Eq. (2.5)–(2.7) into a integral equation on the boundary $\Gamma(t)$. We use the kernel K of the heat equation to express the solution W to Eqs. (2.5)–(2.6) as a single layer heat potential. Given a function V on

$$\Gamma_T = \prod_0^T \Gamma(t) = \{(x, t) \mid x \in \Gamma(t), 0 \leqslant t \leqslant T\},$$

the single layer heat potential SV is defined for (x, t) in $B \times [0, T]$ by

$$SV(x, t) = \int_0^t \int_{\Gamma(t')} K(x, x', t - t') V(x', t') \, dx' \, dt'. \tag{2.8}$$

Here the x' integration is over the curves comprising $\Gamma(t')$, and the Green function K of the heat equation in the box

$B = [0, 1] \times [0, 1]$ with Dirichlet boundary conditions on the box walls is given by

$$K(x, x', t') = \sum_{k_1 = 1}^{\infty} \sum_{k_2 = 1}^{\infty} e^{-(k_1^2 + k_2^2)\pi^2 t'} \sin(k_1 \pi x_1)$$

$$\times \sin(k_2 \pi x_2) \sin(k_1 \pi x_1') \sin(k_2 \pi x_2') \tag{2.9}$$

$$= \frac{1}{4\pi t} \sum_{k_1 = -\infty}^{\infty} \sum_{k_2 = -\infty}^{\infty} \sum_{\sigma_1 = \pm 1} \sum_{\sigma_2 = \pm 1} \sigma_1 \sigma_2$$

$$\times e^{-[(x_1 - \sigma_1 x_1' - 2k_1)^2 + (x_2 - \sigma_2 x_2' - 2k_2)^2]/4t}, \tag{2.10}$$

where $x = (x_1, x_2)$ and $x' = (x_1', x_2')$. The first expression for K can be calculated by Fourier series, the second by the method of images. The equivalence of the two is a special case of the Poisson summation formula (Dym and McKean [9]).

The function $SV(x, t)$ defined by Eq. (2.8) is a continuous function on $B \times [0, T]$, vanishing for $t = 0$ or x on ∂B, and satisfying the heat equation everywhere off Γ_T. Across $\Gamma(t)$, however, $SV(x, t)$ has a jump in its normal derivative equal to V. Thus, $W(x, t) = H \cdot SV(x, t)$ is the solution to Eqs. (2.5)–(2.6). All that remains is to satisfy the second boundary condition Eq. (2.7). This is equivalent to the boundary integral equation

$$\varepsilon_C(n)C + \varepsilon_V(n)V + U$$

$$+ H \int_0^t \int_{\Gamma(t')} K(x, x', t - t')$$

$$\times V(x', t') \, dx' \, dt' = 0 \tag{2.11}$$

for $x \in \Gamma(t)$. Equation (2.11) is an integral equation for the normal velocity of the moving boundary. We note that the velocity V of a point x on $\Gamma(t)$ depends not only on the position of $\Gamma(t)$ but also on its previous history. Thus, we have stored information about the temperature off the moving boundary in the previous history of the boundary.

Step 3. Level Set Formulation of Moving Boundaries

Numerical approximations to the boundary integral equation based on marker particle techniques have been used to solve solidification problems similar to the one under investigation (see Meiron [22], Strain [38], Kessler and Levine [17], Karma [15], Brush and Sekerka [2]). In this approach, the boundary $\Gamma(t)$ is represented by a set of marker points. A discretized version of the integral equation is then solved for the velocity of each point, and this velocity is then used to update the positions of the markers. There are several difficulties with this approach. First, at each time step one must evaluate an integral over all previous history. Thus, given $1 \leqslant i \leqslant M$ points on the curve at each of $1 \leqslant n \leqslant N$ time levels $n \, \Delta t$, it costs $O(M^2 N^2)$ operations just

to evaluate SV at the given points. This becomes very expensive for large M and N, as documented in Strain [38] and Greengard and Strain [12]. Independent of the expense, fundamental problems with marker particle algorithms arise as the moving boundary $\Gamma(t)$ becomes more complex (see Sethian [32, 33]). Sharp corners and cusps can develop which are difficult to resolve with marker particles, and instabilities often result when the boundary becomes interesting. Furthermore, it is difficult to use marker particles when the boundary changes topology, for example, when two boundaries collide or a piece of solid breaks off. Finally, the extension to three dimensional problems is challenging.

An alternative to marker particles for moving boundaries is provided by the level set technology introduced in Osher and Sethian [30]. With this approach, a complex boundary can be advanced. Sharp corners and cusps are handled naturally, and changes of topology in the moving boundary require no additional effort. Furthermore, these methods work in any number of space dimensions. Recently, this technique has been applied to interface problems in mean curvature flow [33], singularity formation and minimal surface construction [34], and compressible gas dynamics [24]. In addition, theoretical analysis of mean curvature flow based on the level set model presented in Osher and Sethian [30] has recently been developed in [10].

We now describe the level set algorithm in the general case when a curve or union of curves $\Gamma(t)$ moves with speed V normal to itself. The essential idea is to construct a function $\phi(x, t)$ defined everywhere on B, such that the level set $\{\phi = 0\}$ is the set $\Gamma(t)$; that is,

$$\Gamma(t) = \{x : \phi(x, t) = 0\}. \tag{2.12}$$

We now derive a partial differential equation for ϕ, which holds on $B \times [0, T]$. Suppose we can construct a smooth function $F(x, t)$ defined on all of B such that

$$V(x, t) = F(x, t) \quad \text{for} \quad x \in \Gamma(t). \tag{2.13}$$

Then $V = F$ on Γ_T, and we call F a *smooth extension of V off* $\Gamma(t)$. We shall postpone until the next section the extension for the case of crystal growth.

What is the equation of motion for ϕ? Obviously, this equation must reduce to normal propagation by speed V on the level set $\phi = 0$. Suppose we initialize $\phi(x, 0)$ such that

$$\phi(x, 0) = \pm \text{distance from } x \text{ to } \Gamma(t), \tag{2.14}$$

where the minus (plus) sign is chosen if x is inside (outside) the initial boundary $\Gamma(t = 0)$. Now consider the motion of some level set $\{\phi(x, t) = C\}$. Here, we follow the derivation

given in [24]. Let $x(t)$ be the trajectory of a particle located on this level set, so

$$\phi(x(t), t) = C. \tag{2.15}$$

The particle speed $\partial x / \partial t$ in the direction n normal to $\Gamma(t)$ is given by the speed function F. Thus,

$$\frac{\partial x}{\partial t} \cdot n = F, \tag{2.16}$$

where the normal vector n is given by $n = \nabla \phi / |\nabla \phi|$. By the chain rule,

$$\phi_t + \frac{\partial x}{\partial t} \cdot \nabla \phi = 0 \tag{2.17}$$

and substitution yields

$$\phi_t + F |\nabla \phi| = 0$$
$$\phi(x, t = 0) = \text{given}. \tag{2.18}$$

Equation (2.18) yields the motion of $\Gamma(t)$ with normal velocity V on the level set $\phi = 0$. We refer to Eq. (2.18) as the level set "Hamilton–Jacobi" formulation. It is not strictly a Hamilton–Jacobi equation except for certain speed functions F, but the flavor of Hamilton–Jacobi equations is present.

The initial value problem (2.18) can be solved numerically by finite difference schemes. Because ϕ can develop corners and sharp gradients, numerical techniques borrowed from hyperbolic conservation laws are used to produce upwind schemes for ϕ which track sharp corners accurately and employ the correct boundary conditions at the edge of the computational box. These schemes are presented in Section 4.

Step 4. Construction of the Speed Function F

We now describe how to extend the velocity V to a globally defined speed function F. Such an extension is necessary in order to use the level set formulation.

The most natural extension makes direct use of the integral equation (2.11), namely,

$$\varepsilon_C(n) C + \varepsilon_V(n) V + U$$
$$+ H \int_0^t \int_{\Gamma(t')} K(x, x', t - t')$$
$$\times V(x', t') \, dx' \, dt' = 0 \tag{2.19}$$

for $x \in \Gamma(t)$. Each term in Eq. (2.19) can be evaluated

anywhere in B, once V is known on $\Gamma(t')$ for $0 \leqslant t' \leqslant t$ and ϕ is known on B. Thus, given the set $\Gamma(t)$, plus all its previous positions and velocities for $0 \leqslant t' \leqslant t$, one could first solve an integral equation to find the velocity V for all points on $\Gamma(t)$ and then find $F(x, t)$ by solving the equation

$$\varepsilon_C(n)C + \varepsilon_V(n)\,F(x, t) + U(x, t)$$
$$+ H\int_0^t \int_{\Gamma(t')} K(x, x', t - t')$$
$$\times V(x', t')\,dx'\,dt' = 0 \qquad (2.20)$$

for F throughout B. Here, x is a point in B, while $C = C(x, t)$ is the curvature, and n is the outward normal vector to the level set passing through x;

$$C = \nabla \cdot \left(\frac{\nabla\phi}{|\nabla\phi|}\right) = \nabla \cdot n$$
$$n = \frac{+\nabla\phi}{|\nabla\phi|} \qquad (2.21)$$

and these expressions make sense everywhere in B. The speed funcion F given by Eq. (2.20) is defined throughout B and equal to V on the solid/liquid boundary $\Gamma(t)$.

For our purposes, it is easier to work with a regularization of the speed function. We split the single layer heat potential into a history part $S_\delta V$ and a local part $S_L V$ as follows:

$$SV(x, t) = \int_0^{t-\delta} \int_{\Gamma(t')} K(x, x', t - t')\,V(x', t')\,dx'\,dt'$$
$$+ \int_{t-\delta}^t \int_{\Gamma(t')} K(x, x', t - t')\,V(x', t')\,dx'\,dt'$$
$$\equiv S_\delta V + S_L V. \qquad (2.22)$$

Here δ is a small regularization parameter. Heuristically, we try to separate the local part, which is causing the jump in the normal derivative of the potential, from the history part which is smooth and independent of the current velocity. In Section 5, we show that the local part $S_L V$ can be approximated by

$$S_L V(x, t) = \sqrt{\delta/\pi}\,V(x, t) + O(\delta^{3/2}) \qquad (2.23)$$

at points x on $\Gamma(t)$. The history part $S_\delta V$ depends only on values of V at times t' bounded away from the current time.

expression for the local part $S_L V$ varies smoothly in the direction normal to the curve if V does, this suggests the

following regularized extension of V. Let F be defined everywhere in B by

$$\varepsilon_C(n)C + \varepsilon_V(n)\,F(x, t) + U(x, t)$$
$$+ H\sqrt{\delta/\pi}\,F(x, t) + HS_\delta V = 0. \qquad (2.24)$$

We can then solve this equation for F to produce

$$F = \frac{-1}{\varepsilon_V(n) + H\sqrt{\delta/\pi}}\,[\varepsilon_C(n)C + U + HS_\delta V]. \qquad (2.25)$$

Thus, F is an *explicit* function of ϕ, the velocity at previous times $t' \leqslant t - \delta$, and the free temperature field U.

Note that F is defined even if $\varepsilon_V = 0$. Furthermore, F is equal to $V + O(\delta^{3/2})$ on $\Gamma(t)$, and does not have a boundary layer as we move off the curve, as long as δ is not too small. Finally, in the limit $\delta \to 0$, Eq. (2.25) reduces to Eq. (2.20) as it should.

Remark. We have reduced the equations of motion, with an $O(\delta^{3/2})$ error, to a pair of equations on a *fixed* domain B:

$$\varepsilon_C(n)C + \varepsilon_V(n)F + U(x, t)$$
$$+ H\sqrt{\delta/\pi}\,F + HS_\delta V = 0 \qquad (2.26)$$
$$\phi_t + F\,|\nabla\phi| = 0. \qquad (2.27)$$

These equations constitute a fixed-domain approximation of the crystal growth equations of motion. No analysis of the original equations of motion has been carried out, mostly because there is no fixed-domain formulation. An approximate fixed-domain formulation may be a viable substitute.

3. OUTLINE OF THE ALGORITHM

In this section, we describe the general flow of the algorithm from one time step to the next. We have a pair of equations for the speed function F and the level set function ϕ defined throughout B, namely

$$\varepsilon_C(n)C + \varepsilon_V(n)F + U(x, t)$$
$$+ H\sqrt{\delta/\pi}\,F + HS_\delta V = 0 \qquad (3.1)$$
$$\phi_t + F\,|\nabla\phi| = 0. \qquad (3.2)$$

Here, the curvature C and the normal vector n are functionals of ϕ. U is computed by solving the heat equation on B. $S_\delta V$ is computed from the previous history of $\Gamma(t)$.

To describe the algorithm, we imagine that at time step

(1) the level set function ϕ_{ij}^n defined at discrete grid points x_{ij}

(2) the free temperature field U_{ij}^n defined on the same discrete grid

(3) the previous positions of the boundary $\Gamma(m\,\Delta t)$, $m = n - 1, n - 2, ..., n - d$ (where $d = \delta/\Delta t$), stored as points on each curve.

Given this information, we proceed from one time step to the next as follows:

ALGORITHM.

Step 1. At each grid point x_{ij}, compute the extended speed function F_{ij}^n from (3.1). This is done as follows:

(a) Expressions for the discrete curvature C_{ij} and normal vector n_{ij} may be computed from the discrete level set function ϕ_{ij}^n without explicit construction of the particular level set passing through the grid point x_{ij} (see Section 4).

(b) The history part $S_\delta V$ is updated by using the stored boundary at previous time levels. This part of the calculation is described in Section 5.

Step 2. Calculate ϕ_{ij}^{n+1} from ϕ_{ij}^n and F_{ij}^n using the upwind, finite difference scheme described in Section 4.

Step 3. Calculate U_{ij}^{n+1} from U_{ij}^n by solving the finite difference approximation to the heat equation discussed in Section 6.A.

Step 4. Find point on the new boundary $\Gamma((n+1)\,\Delta t)$ by constructing the level set $\phi = 0$ from ϕ_{ij}^{n+1}, (see Section 6.C). Store the position x and velocity V of each point, found by interpolating F from the values on the grid. Note that these points on the boundary *do not move*. They serve only as quadrature points for updating the history integral $S_\delta V$.

Step 5. Replace n by $n+1$ and return to Step 1.

In Section 4, we present the details of the level set algorithm and in Section 5 we give the details of the evaluation of the history part of the single layer potential. Section 6 presents other numerical details.

4. SOLVING THE LEVEL SET EQUATION

We have reformulated the problem of moving $\Gamma(t)$ with normal velocity V by representing $\Gamma(t)$ as the zero set

$$\Gamma(t) = \{x \mid \phi(x, t) = 0\}. \tag{4.1}$$

Here ϕ solves the initial value problem

$$\phi_t + F\,|\nabla\phi| = 0$$

$$\phi(x, t = 0) = \pm\,\text{distance from } x \text{ to } \Gamma(t=0) \tag{4.2}$$

and F is a smooth extension of V off $\Gamma(t)$. There are at least two advantages of this level set formulation compared to

methods based on parametrizing $\Gamma(t)$. The first is that $\phi(x, t)$ always remains a function, even if the level surface $\phi = 0$ corresponding to the boundary $\Gamma(t)$ changes topology, breaks, merges, or forms sharp corners. Parametrizations of the boundary become multiple-valued or singular in these cases. As an example, consider two circles in R^2 expanding outward with normal velocity $V = 1$ (Fig. 3a). The initial function ϕ is a double-humped function which is continuous but not everywhere differentiable (Fig. 3b). As ϕ evolves under Eq. (4.2), the topology of the level set $\phi = 0$ changes. When the two circles expand, they meet and merge into a single closed curve with two corners (Fig. 3c). This is reflected in the change of topology of the level set $\phi = 0$ (Fig. 3d).

Another advantage of this formulation concerns numerical approximation. Because $\phi(x, t)$ remains a function as it evolves, finite difference approximations may be used to approximate the spatial and temporal derivatives. For example, given a uniform mesh with spacing h, with grid nodes x_{ij} and time step Δt, let ϕ_{ij}^n approximate $\phi(x_{ij}, n\,\Delta t)$. An approximation to Eq. (4.2) is given by

$$\frac{\phi_{ij}^{n+1} - \phi_{ij}^n}{\Delta t} + F_{ij}^n\,|\nabla_h\phi_{ij}^n| = 0. \tag{4.3}$$

Here, we have used forward differences in time, and some appropriate finite difference operator $\nabla_h\phi_{ij}^n$ to approximate the spatial gradient. In this section, we shall discuss how accurate and efficient approximations $\nabla_h\phi_{ij}^n$ are obtained by exploiting the close link between Hamilton–Jacobi equations and hyperbolic conservation laws. Details may be found in [30].
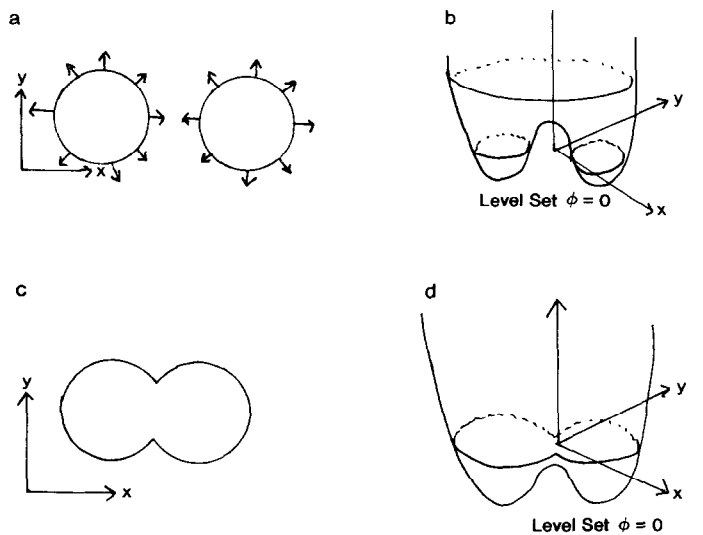

FIG. 3. Formulation of level set approach: (a) propagation of two circles; (b) initial value of level set function; (c) merger of propagating circles; (d) change of topology of level set $\phi = 0$.

A. *Corner Formation and the Entropy Condition*

To illustrate, suppose we wish to follow an initial cosine curve $\Gamma(t=0)$ propagating with normal velocity $V = 1 - \varepsilon C$, where C is the local curvature of the boundary. The curvature of the level curve passing through a point (x, y, t) is given by

$$
\begin{aligned}
C &= \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \\
&= - \frac{\phi_y^2 \phi_{xx} - 2\phi_x \phi_y \phi_{xy} + \phi_x^2 \phi_{yy}}{(\phi_x^2 + \phi_y^2)^{3/2}}.
\end{aligned}
\tag{4.4}
$$

Once again, the minus sign occurs because we have initialized the surface so that $\nabla \phi$ points inwards, and we want C to be positive for a circle. The smooth extension of V to F is straightforward, and Eq. (4.2) becomes

$$
\phi_t + (\phi_x^2 + \phi_y^2)^{1/2} = \varepsilon \nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right)
\tag{4.5}
$$

$$
\phi(x, y, t=0) = \pm \text{distance from } (x, y) \text{ to } \Gamma(t=0).
$$

As shown in [32], for $\varepsilon > 0$, the parabolic right-hand side diffuses sharp gradients and forces ϕ to stay smooth for all time. This is not true for $\varepsilon = 0$ and $F = 1$. In Fig. 4, we show the level set $\phi = 0$ corresponding to the propagating boundary $\Gamma(t)$. For $\varepsilon = 0$, the boundary moves with unit speed, and a corner must develop, since the two sides come together at a constant rate (see Fig. 4a). Thus, singularities in the curvature C develop from smooth initial data.

What happens once a corner develops? It is not clear how to propagate in the normal direction, since the derivative is not defined at the corner. One possibility is the "swallowtail" solution found by letting the boundary pass through itself, see Fig. 4b. However, the boundary consists only of those points located a distance t from the initial curve, which is the Huygens' principle construction. Roughly speaking, we may obtain this solution by removing the "tail" from the "swallowtail." In Fig. 4c, we show this alternative "weak" solution. Another way to characterize this weak solution is through the following "entropy condition," see [32]: If the boundary is viewed as a burning flame, then *once a particle is burnt it stays burnt.* Careful adherence to this stipulation produces the same weak solution as the Huygens' principle construction. Furthermore, this physically reasonable weak solution has an equally appealing mathematical quality. It is the formal limit of the smooth solutions for $\varepsilon > 0$ as $\varepsilon \to 0$ and the curvature term vanishes [32].

Thus, our goal is to produce approximations to the spatial derivative that (1) do not artificially smooth sharp corners and (2) pick out the correct entropy solution when singularities develop. Our schemes are motivated by
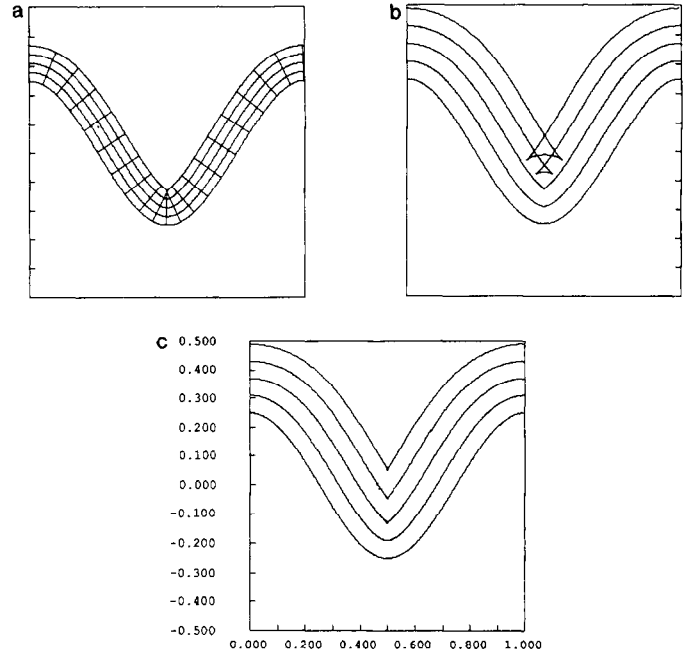
FIG. 4. Corner formation and the entropy condition: (a) propagating curve until singularity forms; (b) entropy-violating swallowtail solution; (c) entropy-satisfying solution from Huyghen's construction.

the fact [32] that our entropy condition for propagating boundaries is identical to the one for hyperbolic conservation laws, where stable, consistent, entropy-satisfying algorithms have a rich history. Our next step is to exploit this link.

B. *Hamilton–Jacobi Equations and Hyperbolic Conservation Laws in One Dimension*

As motivation, consider the one-dimensional version of the level set equation, with constant normal velocity $V = 1$. Then Eq. (4.2) becomes a standard Hamilton–Jacobi equation

$$
\phi_t + H(\phi_x) = 0, \qquad \text{where} \quad H(\phi_x) = -(\phi_x)^{1/2}
$$
$$
\phi(x, 0) = \text{given}.
\tag{4.6}
$$

Let $u = \phi_x$. Taking the derivative of Eq. (4.6) with respect to x, we have

$$
u_t + [H(u)]_x = 0, \qquad \text{where} \quad H(u) = -(u^2)^{1/2}
$$
$$
u(x, 0) = \text{given}.
\tag{4.7}
$$

Equation (4.7) is a single scalar hyperbolic conservation law in one space variable. Solutions can develop discontinuous jumps, known as *shocks*, even with smooth initial data [20]. Jumps in the slope $u = \phi_x$ are equivalent to corners in ϕ [32]. In order to make sense of the solution after shocks

form, we study an integral version of the conservation law (4.7) which admits discontinuous solutions. Thus, consider an arbitrary interval $[a, b]$. We integrate both sides of Eq. (4.7) to produce

$$\frac{d}{dt} \int_a^b u(x, t)\, dx = H[u(a, t)] - H[u(b, t)]. \qquad (4.8)$$

We say that u is a *weak solution* of the conservation law if it satisfies the above integral equation. Note that u need not be differentiable to satisfy the integral form of the conservation law.

How can we be sure that a numerical algorithm will approximate the correct, entropy-satisfying solution to Eq. (4.8)? We begin with a definition, which is simply a numerical version of Eq. (4.8):

DEFINITION. Let u_i^n be the value of u at mesh point $i\,\Delta x$ at time $n\,\Delta t$. A three-point difference scheme is said to be in *conservation form* if there exists a function $g(u_1, u_2)$ such that the scheme can be written in the form

$$\frac{u_i^{n+1}\,\Delta x - u_i^n\,\Delta x}{\Delta t} = g(u_{i-1}^n, u_i^n) - g(u_i^n, u_{i+1}^n),$$

$$\text{where} \quad g(u, u) = H(u). \qquad (4.9)$$

This definition is natural; the scheme must approximate the hyperbolic conservation law, subject to the consistency requirement $g(u, u) = H(u)$. In order to guarantee that the

tion, we require *monotonicity*, that is, that u_i^{n+1} be an increasing function of the arguments u_{i-1}^n, u_i^n, u_{i+1}^n. We can now state the main fact: A conservative, monotone scheme produces a solution that satisfies the entropy condition.

How do we convert Eq. (4.9), which is a scheme for the slope u, into a scheme for ϕ itself? We begin by writing Eq. (4.6) with a forward difference in time, namely,

$$\phi_i^{n+1} = \phi_i - \Delta t\, H(u). \qquad (4.10)$$

Since the numerical flux function g approximates H, the solution to Eq. (4.10) may be approximated by

$$\phi_i^{n+1} = \phi_i^n - \Delta t\, g(u_{i-1/2}, u_{i+1/2}) \qquad (4.11)$$

$$\approx \phi_i^n - \Delta t\, g\left(\frac{\phi_i^n - \phi_{i-1}^n}{\Delta x}, \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}\right), \qquad (4.12)$$

$$= \phi_i^n - \Delta t\, g(D_x^- \phi_i, D_x^+ \phi_i), \qquad (4.13)$$

where g is an appropriate numerical flux function and we

have used the standard definitions of the forward and backward difference operators, namely,

$$D_x^- \phi_i = \frac{\phi_i^n - \phi_{i-1}^n}{\Delta x},$$

$$D_x^+ \phi_i = \frac{\phi_{i+1}^n - \phi_i^n}{\Delta x}. \qquad (4.14)$$

Finally, we need an appropriate numerical flux function g. In the special case where $H(u)$ may be written as a function of u^2, i.e., $H(u) = f(u^2)$ for some function f, we may use the Hamilton–Jacobi flux function given in [30]:

$$g(u_{i-1/2}, u_{i+1/2}) = g(D_x^- \phi_i, D_x^+ \phi_i)$$

$$= f((\max(D_x^- \phi, 0))^2 + (\min(D_x^+ \phi, 0))^2). \qquad (4.15)$$

This conservative monotone scheme is an *upwind* method in that it differences in the direction of propagating characteristics. This is important, since we must impose boundary conditions on the walls of a finite-sized computational box. An upwind scheme automatically differences in the outward-flowing direction at the box walls if the boundary is expanding, thus information flows out. In the special case when $V = 1$, so that $f(u^2) = -(u^2)^{1/2}$, we have

$$\phi_i^{n+1} = \phi_i^n - \Delta t\, ((\max(D_x^- \phi, 0))^2 + (\min(D_x^+ \phi, 0))^2)^{1/2}. \qquad (4.16)$$

weak solution to the moving boundary problem. More details about shock schemes and conservation laws may be found in [29].

C. *Multimensional Problems and More Complicated Speed Functions*

We now extend the previous discussion to problems in more than one space dimension. Recall that we are solving the "Hamilton–Jacobi"-type equation

$$\phi_t + F\,|\nabla\phi| = 0. \qquad (4.17)$$

We begin by decomposing F into two components:

$$F = F_A + F_G. \qquad (4.18)$$

Here, F_A is an advection term containing that part of the velocity which is independent of the moving boundary, and F_G contains those terms which depend on geometric properties of the boundary, such as the curvature and normal. We

begin by splitting the influence of F, and rewrite Eq. (4.17) as

$$\phi_t = -(F_A \, |\nabla\phi| + F_G \, |\nabla\phi|). \qquad (4.19)$$

In two space dimensions, one can easily extend the scheme given in Eq. (4.12) by differencing in each direction to produce

$$\phi_{ij}^{n+1} = \phi_{ij}^n - F_A \, \Delta t \, ((\max(D_x^- \, \phi_{ij}, 0))^2$$
$$+ (\min(D_x^+ \, \phi_{ij}, 0))^2 + (\max(D_y^- \, \phi_{ij}, 0))^2$$
$$+ (\min(D_y^+ \, \phi_{ij}, 0))^2)^{1/2} - \Delta t \, F_G \, |\nabla\phi|. \qquad (4.20)$$

Here, we have not approximated the final term $F_G \, |\nabla\phi|$; one may use a straightforward central difference approximation to this term. This is the first-order multidimensional algorithm described in [30].

In Fig. 5, we show this technique applied to the case of a star propagating outwards with speed $F = 1 - \varepsilon C$, where C is the curvature as in Eq. (4.5). Here, we take $\Delta t = 0.00025$ and a mesh size of 160 points in each direction on a unit box ($\Delta x = \Delta y = 1/159$). The curve evolves smoothly and becomes circular as it evolves. The orientation of the underlying grid used to advect the circle is difficult to detect.

F. "Grid-Free" Extension to Multi-Dimensions

Unfortunately, when the speed function F is more complicated, the above extension to multi-dimensions is inadequate. In particular, for the anisotropic speed function

$$F = 1 - \varepsilon_C(1 - A \, \cos(k_A \theta + \theta_0))C, \qquad (4.21)$$

which is the crystal growth speed function given in Eq. (2.25) in the limit $H \to 0$, $\delta \to 0$, the above method
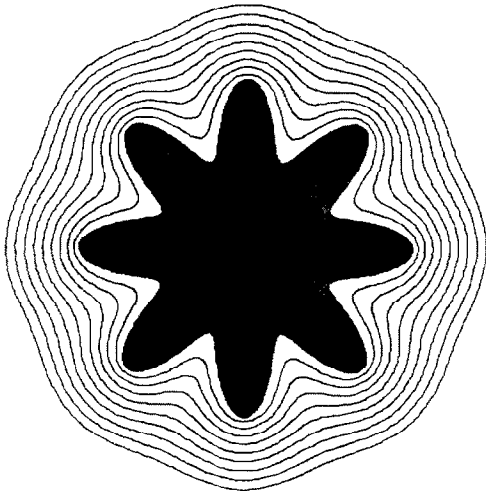
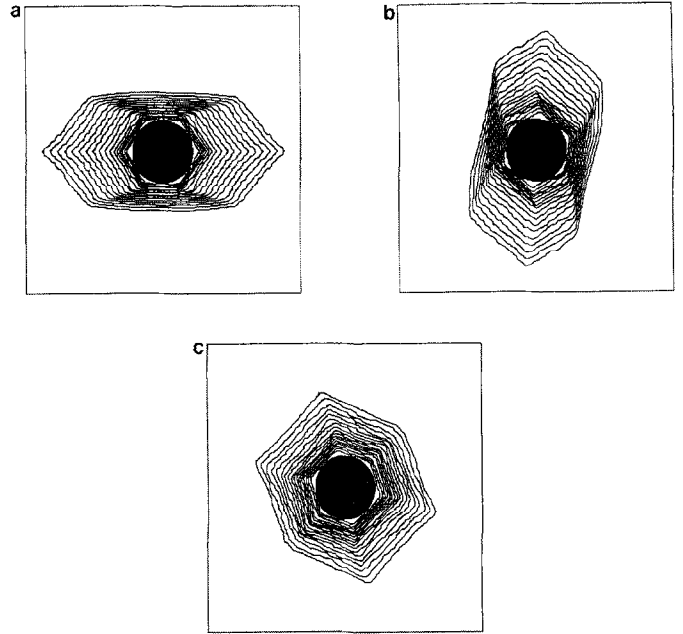**FIG. 5.** Expanding star, $F(K) = 1 - 0.01 \, K$.

FIG. 6. Grid effects in anisotropic motion: (a) grid effects: sixfold symmetry, phase angle = 0.0; (b) grid effects: sixfold symmetry, phase angle = 22.5°; (c) grid effects: sixfold symmetry, phase angle = 45.0°.

suffers from grid effects. In Fig. 6, we show the effects of the algorithm applied to a circle moving under a sixfold anisotropy ($k_A = 6$) speed function. Here, we have taken $\varepsilon_C = 0.01$, $\varepsilon_V = 0.00$. In Fig. 6a, the phase angle $\theta_0 = 0$, thus the preferred modes of growth are along the axes at 0°, 60°, and 120°. In Fig. 6b, the phase angle is shifted to $\theta_0 = \pi/16$; thus the preferred growth direction axes are shifted accordingly. In Fig. 6c, the phase angle is shifted to $\theta_0 = \pi/8$. It is clear that grid effects are pronounced, as shown in the asymmetry of the evolving shapes.

This problem arises because of our construction of the length of the gradient. We have used the approximation

$$|\nabla\phi| = (\phi_x^2 + \phi_y^2)^{1/2}$$
$$\approx ((\max(D_x^- \, \phi, 0))^2 + (\min(D_x^+ \, \phi, 0))^2$$
$$+ (\max(D_y^- \, \phi, 0))^2 + (\min(D_y^+ \, \phi, 0))^2)^{1/2}. \qquad (4.22)$$

As the family of level sets sharpen along the lines of anisotropy, they develop sharp corners. At those corners, the approximation of $|\nabla\phi|$ is biased with respect to the grid lines. As can be seen from Fig. 6, when the preferred lines of growth do not lie along the x and y axis, the problem is substantial.

Grid effects may be significantly diminished by the following alternative multi-dimensional scheme. At each grid point, we find the normal vector to the level curve ($\phi = \text{Const}$) passing through that grid point by averaging the one-sided limits of the normal vectors. We then perform

one-dimensional upwind differencing in the calculated normal direction.

We now explain this strategy in detail. If ϕ is smooth, then the unit normal at point (x_0, y_0) to the level curve passing through (x_0, y_0) is $\nabla\phi(x_0)/|\nabla\phi(x_0)|$. However, when ϕ develops corners, care must be taken in the construction of this normal, since the speed function F is highly sensitive to the exact direction of the normal in the anisotropic case given by Eq. (4.21). At a corner, the direction of the normal undergoes a jump. This suggests the following technique. We form the one-sided difference approximation to the unit normal in each possible direction. We then average all four limiting normals to produce the approximate normal at the corner, see Fig. 7:

$$n_{ij} \equiv (\hat{x}_{ij}, \hat{y}_{ij}) = \left[\frac{\phi_x}{(\phi_x^2 + \phi_y^2)^{1/2}}, \frac{\phi_y}{(\phi_x^2 + \phi_y^2)^{1/2}} \right]$$

$$\approx \frac{1}{4} \left[\frac{(D_x^+ \phi_{ij}, D_y^+ \phi_{ij})}{((D_x^+ \phi_{ij})^2 + (D_y^+ \phi_{ij})^2)^{1/2})} \right.$$

$$+ \frac{(D_x^+ \phi_{ij}, D_y^- \phi_{ij})}{((D_x^+ \phi_{ij})^2 + (D_y^- \phi_{ij})^2)^{1/2})}$$

$$+ \frac{(D_x^- \phi_{ij}, D_y^+ \phi_{ij})}{((D_x^- \phi_{ij})^2 + (D_y^+ \phi_{ij})^2)^{1/2})}$$

$$\left. + \frac{(D_x^- \phi_{ij}, D_y^- \phi_{ij})}{((D_x^- \phi_{ij})^2 + (D_y^- \phi_{ij})^2)^{1/2})} \right]. \qquad (4.23)$$

If any of the one-sided approximations to $|\nabla\phi|$ in the denominator vanish, we ignore that component and re-weight accordingly. This discrete unit normal vector $n_{ij} = (\hat{x}_{ij}, \hat{y}_{ij})$ is then used to compute θ_{ij} in the anisotropy-based speed function F_{ij} given in Eq. (4.21).
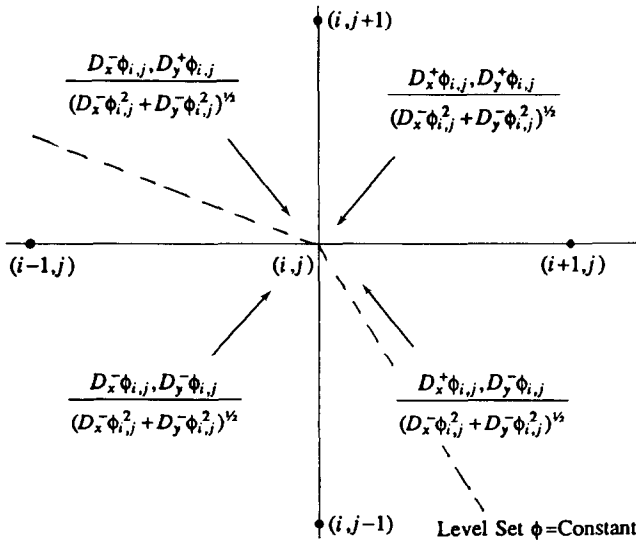
The discrete unit normal n_{ij} calculated above can then be used to construct a relatively "grid-free" multi-dimensional scheme for moving boundaries. We begin by constructing the forward and backward differences, approximating the directional derivative in the direction n_{ij} at the grid point $(i\,\Delta x, j\,\Delta y)$, namely,

$$D^+ \phi_{ij} = \frac{\phi(i\,\Delta x + \Delta s\,\hat{x}_{ij}, j\,\Delta y + \Delta s\,\hat{y}_{ij}) - \phi_{ij}}{\Delta s} \qquad (4.24)$$

$$D^- \phi_{ij} = \frac{\phi_{ij} - \phi(i\,\Delta x - \Delta s\,\hat{x}_{ij}, j\,\Delta y - \Delta s\,\hat{y}_{ij})}{\Delta s} \qquad (4.25)$$

with step size $\Delta s = \min(\Delta x, \Delta y)$. The values of ϕ in between grid points are found by interpolation from neighboring grid values. The above forward and backward difference operators are used to construct an upwind approximation to the gradient based on the one-dimensional scheme given in Eq. (4.12), namely,

$$\phi_{ij}^{n+1} = \phi_{ij}^n - F_{Aij}\,\Delta t$$

$$\times (\max(D^- \phi_{ij}, 0)^2 + \min(D^+ \phi_{ij}, 0)^2))^{1/2}$$

$$- \Delta t \,[F_{Gij} \,|\nabla\phi|_{ij}]. \qquad (4.26)$$

To summarize, we first construct the normal vector and then use one-dimensional upwind differencing in the calculated normal direction. Note that this methodology may be extended to any number of space dimensions. Equation (4.26) is the scheme used in this paper.
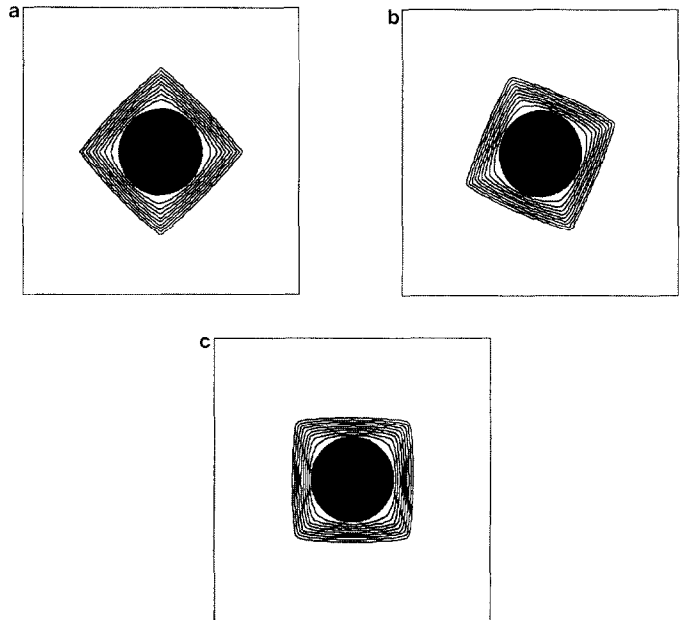

FIG. 8. Modified approach: (a) Modified method: sixfold symmetry, phase angle = 0.0°; (b) Modified method: sixfold symmetry, phase angle = 22.5°; (c) Modified method: sixfold symmetry, phase angle = 45.0°.
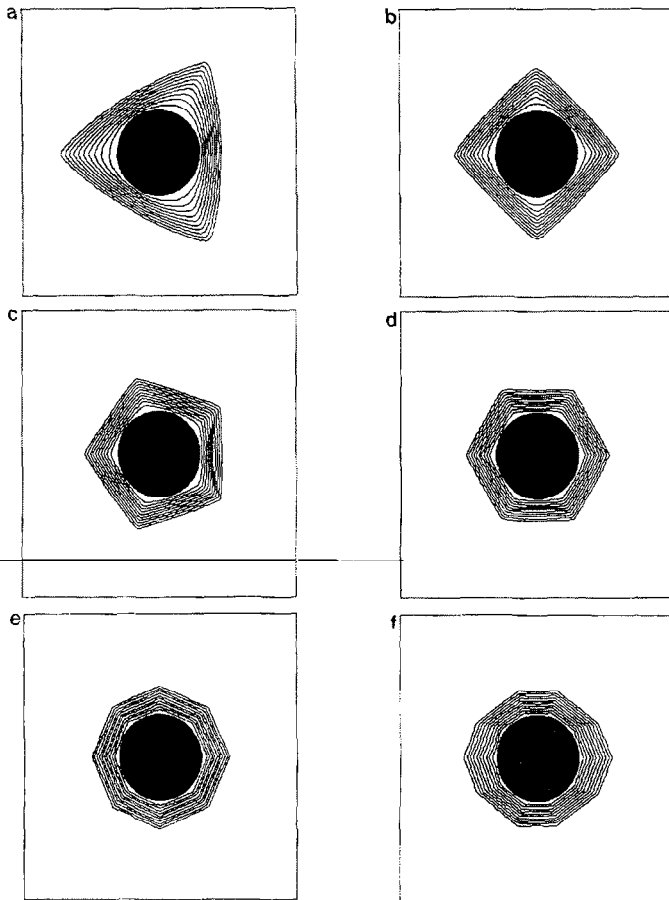

FIG. 7. Construction of modified method.

FIG. 9. Calculation of motion under various crystalline anistropies: (a) Threefold symmetry; (b) Fourfold Symmetry; (c) Fivefold symmetry; (d) Sixfold symmetry; (e) eightfold symmetry; (f) tenfold symmetry.

In Fig. 8, we show the results of this new algorithm applied to the previous example of a circle using the fourfold anisotropy speed function. In Fig. 8a, the phase angle $\theta = 0$; thus the preferred modes of growth are along the x and y axes. In Fig. 8b, the phase angle $\theta = \pi/8$; thus the preferred modes of growth are along the diagonals. In both cases, the circle is transformed into a square, and the grid effects are considerably diminished. In Fig. 9, we show this modified algorithm applied to problems with threefold, fourfold, fivefold, sixfold, eightfold, and tenfold symmetries. In each case, grid effects have been essentially eliminated by using the modified approach.

5. RAPID EVALUATION OF THE SINGLE LAYER HEAT POTENTIAL

The boundary integral formulation of the problem required the evaluation of the single layer heat potential

$$SV(x, t) = \int_0^t \int_{\Gamma(t')} K(x, x', t - t') \, V(x', t') \, dx' \, dt', \qquad (5.1)$$

on a $M \times M$ grid in B. The heat kernel K is given (see Dym and McKean [9] and Strain [39]), by

$$K(x, x', t) = \sum_{k_1 = 1}^{\infty} \sum_{k_2 = 1}^{\infty} e^{-(k_1^2 + k_2^2)\pi^2 t} \sin(k_1 \pi x_1)$$
$$\times \sin(k_2 \pi x_2) \sin(k_1 \pi x_1') \sin(k_2 \pi x_2') \qquad (5.2)$$

$$= \frac{1}{4\pi t} \sum_{k_1 = -\infty}^{\infty} \sum_{k_2 = -\infty}^{\infty} \sum_{\sigma_1 = \pm 1} \sum_{\sigma_2 = \pm 1} \sigma_1 \sigma_2$$
$$\times e^{-[(x_1 - \sigma_1 x_1' - 2k_1)^2 + (x_2 - \sigma_2 x_2' - 2k_2)^2]/4t}. \qquad (5.3)$$

A straightforward numerical approach to calculating $SV(x, t)$ is to use a quadrature formula to approximate the integral. To compute the values of SV at M^2 points at N times $n \, \Delta t$ would then require $O(M^3 N^2)$ work if there are $O(M)$ points on $\Gamma(t)$ at each time t. Since complicated moving boundaries can develop in solidification problems, accurate approximations require large values for M and N and straightforward evaluation schemes become prohibitively expensive, as shown in [38, 12] for evidence of this.

As an alternative, we can use a fast algorithm which evaluates $SV(x, t)$ on a $M \times M$ grid in $O(M^2)$ work per time step, to any specified precision. The constant in $O(M^2)$ depends on the precision and the smoothness of the data $\Gamma(t)$ and V. The basic idea behind this algorithm is as follows. Suppose we split the time integral in $SV(x, t)$ at a cutoff time δ:

$$SV(x, t) = \int_0^{t - \delta} \int_{\Gamma(t')} K(x, x', t - t') \, V(x', t') \, dx' \, dt'$$
$$+ \int_{t - \delta}^t \int_{\Gamma(t')} K(x, x', t - t') \, V(x', t') \, dx' \, dt'$$
$$\equiv S_\delta V + S_L V. \qquad (5.4)$$

The key to evaluating Eq. (5.4) is to use the two different expressions, (5.2) and (5.3), for the kernel K. Substitution of Eq. (5.2) into the history part $S_\delta V$ yields a Fourier series which can be approximated by a small number of terms. In addition, the Fourier coefficients of $S_\delta V$ can be updated in $O(M^2)$ work per time step, using Rokhlin's non-equidistant fast Fourier transform (Rokhlin, unpublished manuscript; Strain [39]). On the other hand, substitution of the method of images sum (5.3) into the local part $S_L V$ produces a rapidly convergent sum, which we can approximate by Taylor series expansion of the first term.

A less robust fast algorithm which evaluates $SV(x, t)$ for x in $\Gamma(t)$ using M quadrature points on $\Gamma(t)$ and requiring only $O(MN)$ work for N time steps has been constructed in [12]. However, we do not need this algorithm here, because we are evaluating only $S_\delta V$ on a $M \times M$ grid and we are evaluating it with $O(M)$ quadrature points given in $\Gamma(t')$. This simplifies the problem considerably.

A. *The History Part $S_\delta V$*

We begin with the history part $S_\delta V$. We have

$$S_\delta V(x, t) = \int_0^{t-\delta} \int_{\Gamma(t')} K(x, x', t - t') V(x', t') \, dx' \, dt'.$$

(5.5)

The Fourier series for K is given by

$$K(x, x', t - t')$$

$$= 4 \sum_{k_1 = 1}^\infty \sum_{k_2 = 1}^\infty e^{-\pi^2 |k|^2 (t - t')} \sin(k_1 \pi x_1)$$

$$\times \sin(k_2 \pi x_2) \sin(k_2 \pi x_2) \sin(k_1 \pi x_1') \sin(k_2 \pi x_2'), \quad (5.6)$$

where $|k|^2 = k_1^2 + k_2^2$. Now $\delta < t - t'$ by construction, and therefore we may bound the summands in Eq. (5.6) by $e^{-\pi^2 |k|^2 \delta}$. Thus, the Fourier series converges quickly for $\delta > 0$ and can be used to provide a good approximation to the history part. Inserting Eq. (5.6) into the history part Eq. (5.5) gives

$$S_\delta V = 4 \int_0^{t-\delta} \int_{\Gamma(t')} \sum_{k_1 = 1}^\infty \sum_{k_2 = 1}^\infty e^{-\pi^2 |k|^2 (t - t')}$$

$$\times \sin(k_1 \pi x_1) \sin(k_2 \pi x_2) \sin(k_1 \pi x_1')$$

$$\times \sin(k_2 \pi x_2') V(x, t') \, dx' \, dt'. \quad (5.7)$$

Interchanging summation and integration gives a Fourier series representation

$$S_\delta V(x, t)$$

(5.8)

$$= \sum_{k_1 = 1}^\infty \sum_{k_2 = 1}^\infty C_k(t, \delta) \sin(\pi k_1 x_1) \sin(\pi k_2 x_2)$$

(5.9)

with coefficients $C_k(t, \delta)$ given by

$$C_k(t, \delta) = 4 \int_0^{t-\delta} \int_{\Gamma(t')} e^{-\pi^2 |k|^2 (t - t')} \sin(\pi k_1 x_1')$$

$$\times \sin(\pi k_2 x_2') V(x', t') \, dx' \, dt'. \quad (5.10)$$

Here, $x = (x_1, x_2)$, and $x' = (x_1', x_2')$. For $\delta > 0$, this series converges rapidly. We first note that since $\delta < t - t'$, the coefficients $C_k(t, \delta)$ decay like Gaussian functions of $|k|$,

$$C_k(t, \delta) \leqslant 4 e^{-\pi^2 |k|^2 \delta} |V|_1,$$

(5.11)

where

$$|V|_1 = \int_0^{t-\delta} \int_{\Gamma(t')} |V(x', t')| \, dx' \, dt'.$$

(5.12)

We can now use the integral test to bound the error E_p incurred by truncating the series (5.9) after the first p^2 terms:

$$E_p \leqslant 4 |V|_1 \left[\sum_{k_1 = 1}^\infty \sum_{k_2 = p + 1}^\infty + \sum_{k_1 = p + 1}^\infty \sum_{k_2 = 1}^\infty \right]$$

$$\times |C_k(t, \delta) \sin(\pi k_1 x_1) \sin(\pi k_2 x_2)|$$

$$\leqslant 8 |V|_1 \left[\int_0^\infty e^{-\pi^2 k^2 \delta} \, dk \int_p^\infty e^{-\pi^2 k^2 \delta} \, dk \right]$$

$$\leqslant 8 |V|_1 \left[\frac{1}{\sqrt{4\pi\delta}} \frac{e^{-\pi^2 p^2 \delta}}{\pi^2 p \delta} \right] = O\left(\frac{e^{-\pi^2 p^2 \delta}}{\delta} \right) \quad (5.13)$$

if $\pi^2 p^2 \delta \geqslant 1$. The final bound is derived by observing that

$$\int_p^\infty e^{-\pi^2 k^2 \delta} \, dk \leqslant \frac{e^{-\pi^2 p^2 \delta}}{\pi^2 p \delta}$$

(5.14)

if $\pi^2 p^2 \delta \geqslant 1$. This error bound corrects an erroneous statement in [12].

The bound given in Eq. (5.14) shows that the series converges exponentially fast as $(p + 1)^2 \delta \to \infty$, even if $\delta \to 0$. Thus the Fourier series truncation error requires that $\delta p^2 \to \infty$ as $\delta \to 0$ and $p \to \infty$.

At this point, it is not obvious that evaluating the series given in Eq. (5.9) is any less expensive than the original $O(N^2)$ cost of history-dependence. This is because at each time t every coefficient $C_k(t, \delta)$ must be computed by integrating from 0 to $t - \delta$. Fortunately, the coefficients can be updated one step at a time by recursion, rather than recomputed all the way from $t = 0$. To simplify notation, define $M_k(t')$ by

$$M_k(t') = \int_{\Gamma(t')} \sin(k_1 \pi x_1')(k_2 \pi x_2') V(x', t') \, dx'.$$

(5.15)

Then we have the recursion relation

$$C_k(t, \delta) = 4 \int_0^{t - \delta - \Delta t} e^{-\pi^2 |k|^2 (t - \Delta t - t')} M_k(t') \, dt'$$

$$+ 4 \int_{t - \delta - \Delta t}^{t - \delta} e^{-\pi^2 |k|^2 (t - t')} M_k(t') \, dt'$$

$$= 4 e^{-\pi^2 |k|^2 \Delta t} \int_0^{t - \Delta t - \delta} e^{-\pi^2 |k|^2 (t - t')} M_k(t') \, dt'$$

$$+ 4 \int_{t - \delta - \Delta t}^{t - \delta} e^{-\pi^2 |k|^2 (t - t')} M_k(t') \, dt'$$

$$= e^{-\pi^2 |k|^2 \Delta t} C_k(t - \Delta t, \delta)$$

$$+ 4 \int_{t - \delta - \Delta t}^{t - \delta} e^{-\pi^2 |k|^2 (t - t')} M_k(t') \, dt'. \quad (5.16)$$

Thus $C_k(t, \delta)$ can be computed from $C_k(t - \Delta t, \delta)$, given the

values of the moment $M_k(t')$ over the time step. Since this requires constant work per time step, we can evaluate $C_k(n\, \Delta t, \delta)$ for $n = 1, 2, \dots$ in $O(NW_I)$ work, where W_I is the work involved in computing the moment integrals $M_k(t')$. This eliminates the $O(N^2)$ cost of the history-dependence.

To achieve our goal of $O(M^2)$ work per time step, we choose $p = O(M)$ so we have $O(M^2)$ coefficients C_k to evaluate. Then we can evaluate each one by quadrature with $O(M)$ quadrature points, doing $O(M^2 \log M)$ work altogether, with Rokhlin's non-equidistant fast Fourier transform. Since the Fourier series truncation error requires $\delta p^2 \to \infty$ as $\delta \to 0$ and $p \to \infty$, this means $\delta M^2 \to \infty$ so $\delta > 0$ (Δt) where Δt is the time step. This means that δ will be, in practice, a fixed small number of time steps, and the storage requirements of our method are therefore small. This algorithm uses all the points given on $\Gamma(t)$ and can therefore be expected to be more robust than the algorithm given in [12].

After evaluating the coefficients C_k, we must evaluate a p^2-term Fourier series for $S_H V$ at the M^2 grid points in B. The coefficients are zero-padded and an $M \times M$ FFT applied, for a total of $O(M^2 \log M)$ work.

Finally, we need an accurate quadrature rule to compute ~~three integrals over space and time. Recall that we wish to~~ compute the time integral (Eq. (5.16)), given by

$$4 \int_{t-\delta-\Delta t}^{t-\delta} e^{-\pi^2 |k|^2 (t-t')} M_k(t')\, dt'. \tag{5.17}$$

The exponential can vary with moderate rapidly over the range of integration, when $|k|^2$ is large. On the other hand, M_k is relatively smooth. Thus we interpolate $M_k(t')$ by a linear function of t' passing through the endpoints $t - \delta - \Delta t$ and $t - \delta$. We can evaluate the resulting integral exactly.

Gauss–Legendre quadrature over each line segment is used for the spatial integration over $\Gamma(t)$ required to evaluate the moments M_k, with typically six points being ample. This reduces the integral over $\Gamma(t)$ to a weighted sum of sines, to which the non-equidistant fast Fourier transform applies. A more detailed description of this calculation can be found in Strain [39].

B. The Local Part $S_L V$

Now we derive the approximation to the local part $S_L V$. From the method of images expression (5.3) for the heat kernel K in B, we have

$K(x, x', t - t')$

$$= \sum_{k_1 = -\infty}^{\infty} \sum_{k_2 = -\infty}^{\infty} \sum_{\sigma_1 = \pm 1} \sum_{\sigma_2 = \pm 1} \sigma_1 \sigma_2$$

$$\times \frac{1}{4\pi(t - t')} e^{-[(x_1 - \sigma_1 x'_1 - 2k_1)^2 + (x_2 - \sigma_2 x'_2 - 2k_2)^2]/(4(t - t'))}. \tag{5.19}$$

Since $t - t' < \delta$, the sum in Eq. (5.19) converges very rapidly. Suppose we truncate Eq. (5.19) after one term. The error in this approximation depends on the distance d of the boundary $\Gamma(t')$ from the box walls, since this measures how close the nearest contributing images can lie. Note that this distance d is independent of the numerical parameters. All but the first term (in which $k_1 = k_2 = 0$, $\sigma_1 = \sigma_2 = 1$) are then bounded by $O((1/\delta) e^{-d^2/\delta})$. Then as the numerical parameters are refined and $\delta \to 0$, we can approximate $S_L V$ by the corresponding free-space heat potential

$$S_L V(x, t) \approx \int_{t-\delta}^{t} \frac{1}{4\pi(t - t')} \int_{\Gamma(t')} \\ \times e^{-[(x_1 - x'_1)^2 + (x_2 - x'_2)^2]/4(t - t')} \\ \times V(x', t')\, dx'\, dt'. \tag{5.20}$$

(This is just Kac's "principle of not feeling the boundary" [14].) Since the Gaussian decays so fast for t' small, we can assume that $S_L V(x, t)$ is mostly determined by (x', t') near (x, t). Thus, we can perform a Taylor expansion, and, to lowest order, replace $V(x', t')$ by $V(x, t)$ and $\Gamma(t')$ by its tangent plane at (x, t). A brief calculation, carried out in [12], shows then that

$$S_L V(x, t) = \sqrt{\delta/\pi}\, V(x, t) + O(\delta^{3/2}) \tag{5.21}$$

for $x \in \Gamma(t)$. A higher order Taylor expansion can be carried out, if desired, as in [12], but (5.21) will be accurate enough for our purposes.

Thus, we may summarize the fast algorithm described in this section as

$$SV(x, t) = S_\delta(x, t) + S_L(x, t), \tag{5.22}$$

where

$$S_L V(x, t) = \sqrt{\delta/\pi}\, V(x, t) + O(\delta^{3/2}) \tag{5.23}$$

$$S_\delta V(x, t) = \sum_{k_1 = 1}^{p} \sum_{k_1 = 1}^{p} C_k(t, \delta) \sin \pi k_1 x_1 \\ \times \sin \pi k_2 x_2 + O(e^{-\pi^2 p^2 \delta}/\delta) \tag{5.24}$$

$$C_k(t, \delta) = e^{-\pi^2 |k|^2 \Delta t} C_k(t - \Delta t, \delta) \\ + 4 \int_{t-\delta-\Delta t}^{t-\delta} e^{-\pi^2 |k|^2 (t-t')} M_k(t')\, dt' \tag{5.25}$$

$$M_k = \int_{\Gamma(t')} \sin(k_1 \pi x'_1)(k_2 \pi x'_2)\, V(x', t')\, dx'. \tag{5.26}$$

This decomposition allows us to evaluate $S_\delta V$ in $O(M^2)$ work per time step at M^2 points with $O(M)$ quadrature points on $\Gamma(t)$ at each time $t = n\, \Delta t$.

6. NUMERICAL DETAILS

In this section, we discuss some of the numerical details that arise in the course of the algorithm.

A. The "Free" Temperature Field U_{ij}^n

Computing the speed function F_{ij}^n using Eq. (2.20) requires evaluating the free temperature field U_{ij}^n which is the solution to Eq. (2.4). To update U_{ij}^n, we use an finite difference approximation, for example,

$$U_{ij}^{n+1} = U_{ij}^n + \Delta t$$

$$\times \left[\frac{U_{i,j+1}^n + U_{i,j-1}^n + U_{i+1,j}^n + U_{i-1,j}^n - 4U_{i,j}^n}{h^2} \right],$$

$$(6.1)$$

where $h = \Delta x = \Delta y$. (For ease of notation, we shall write all formulae assuming a uniform grid). The initial conditions $U_{ij}^{n=0}$ are found by solving the consistency conditions implied by the original moving boundary problem, which requires that, at $t = 0$, the two boundary conditions (Eqs. (1.2)–(1.3)) are satisfied on Γ, as well as the boundary conditions on the box wall for the initial temperature field (Eq. (1.1)). Thus we must have

$$u_0 = -\varepsilon_C(n)C - \varepsilon_V(n)V,$$

$$[\partial u/\partial n] = -HV \quad \text{on} \quad \Gamma,$$

and

$$u_0 = u_B.$$

on the box walls. Note that the normal velocity V is also unknown at time $t = 0$. This problem is underdeterminated, since u_0 is arbitrary away from Γ. It is physically and mathematically reasonable to complete the determination of u_0 by imposing the Laplace equation $\Delta u_0 = 0$, subject to the given boundary conditions on Γ and ∂B. Physically, it is reasonable to require u_0 to solve the steady-state heat equation, while mathematically this imposes smoothness and requires u_0 to satisfy the maximum principle. This equation may be solved efficiently by a potential-theoretic method. First, u_0 is represented as the sum of a harmonic function in B with the right boundary values on ∂B and a classical single layer potential with density V spread over Γ, resulting in an integral equation for V on Γ. Once this integral equation is solved for V, u_0 may be evaluated everywhere by an appropriate fast algorithm. We refer the reader to [39] for the details and efficiencies involved.

B. Splitting the Velocity Field and Computing the Curvature

As discussed in Section 4.C., we split the speed function F into two components, namely,

$$F = F_A + F_G, \quad (6.2)$$

where F_A is advection and F_G depends on the geometry of $\Gamma(t)$. For crystal growth, the speed function is given in Eq. (2.20), and we let

$$F_A = \frac{+1}{\varepsilon_V(n) + H\sqrt{\delta/\pi}} [U + S_\delta V] \quad (6.3)$$

$$F_G = \frac{-1}{\varepsilon_V(n) + H\sqrt{\delta/\pi}} \varepsilon_C(n)C. \quad (6.4)$$

We then use Eq. (4.26) to update ϕ. The curvature term in Eq. (6.4) is approximated using central differences applied to the expression for the curvature of the level curve of ϕ passing through x, namely,

$$C = -\frac{\phi_y^2\phi_{xx} - 2\phi_x\phi_y\phi_{xy} + \phi_x^2\phi_{yy}}{(\phi_x^2 + \phi_y^2)^{3/2}}. \quad (6.5)$$

Since C is approximated by central differences, we maintain consistency by choosing a central difference approximation to $|\nabla\phi|$ in the last term of Eq. (4.26). Following the philosophy of Section 4, we could construct a grid-free approximation to the curvature similar to our approximation to the normal, but have not done so here.

C. Finding the Front $\Gamma(t)$

As outlined in Step 4 of Section 3, at each time step, we must find the level set $\phi = 0$. We construct a piecewise linear approximation to $\Gamma(t)$ as follows. Given a cell ij, if $\max(\phi_{i,j}, \phi_{i+1,j}, \phi_{i,j+1}, \phi_{i+1,j+1}) < 0$ or $\min(\phi_{i,j}, \phi_{i+1,j}, \phi_{i,j+1}, \phi_{i+1,j+1}) > 0$, then the cell cannot contain $\Gamma(t)$, so we ignore that cell. Otherwise, find the entrance and exit points where $\phi = 0$ by linear interpolation; this provides two nodes or "marker points" on $\Gamma(t)$ and thus one of the line segments which form our approximation to $\Gamma(t)$. The collection of all such line segments constitutes our approximation to $\Gamma(t)$, which is then stored for future evaluation of the history integrals M_k. The line segments need not be ordered, nor is there need for any further information about the curve. Note that this construction finds $O(M)$ points on $\Gamma(t)$ on a $M \times M$ grid, as assumed in the fast evaluation scheme. Note also that it generalizes, like the rest of our method, to the three-dimensional case.

D. Differing Time Steps

The stability requirement for the explicit method for solving the our level set equation (4.26) is $\Delta t = O(\Delta x^2)$, due

to the parabolic-like curvature dependence. This forces a small time step for fine grids. However, the boundary propagation algorithm is relatively inexpensive compared to the evaluation of the heat potential required in computing F_{ij}^n. Because $\Gamma(t)$ moves little during each time step Δt, the change in F is small. More precisely, there is no stability requirement for the velocity evaluation. Thus, we provide the option of employing two time steps, Δt_{Front} and Δt_{Vel}, with $k \, \Delta t_{Front} = \Delta t_{Vel}$. That is, we take k steps of size Δt_{Front} in the level set algorithm before we recompute the velocity field F_{ij}^n and update the stored boundary by time step Δt_{Vel}. This option both speeds up the computations, since it allows us to increase δ, keep fewer Fourier series terms, and requires less storage, since fewer past time steps need be stored.

7. NUMERICAL RESULTS

In this section, we show a series of calculations with the numerical method presented in this paper.

A. Input Parameters

The numerical input parameters are

h = grid size in the unit box

Δt = time step

δ = size of history cutoff (≥ 0 (Δt) for consistency)

p = number of Fourier modes ($p \geq 0$ ($\Delta t^{-1/2}$) for consistency)

k = number of times boundary moves per velocity evaluation.

The physical input parameters are

$\varepsilon_C, \varepsilon_V$ = anisotropy constants

k_A = crystalline symmetry

θ_0 = angle of crystalline symmetry

u_0 = initial values of temperature field

u_B = boundary values of temperature field on box walls

Size, shape, and position of initial seed.

B. Results

In this section, we present the results of our numerical calculations using our solidification algorithm. We begin with a series of calculations to check numerical consistency; that is, to verify that the computed solution converges as the numerical parameters are refined. (A further verification comparing the computed solution using the level set approach with an accurate boundary integral calculation in their common range of validity is in progress, see [42].) After demonstrating the robustness of the algorithm, we

perform a study of the relative influence of the physical parameters, analyzing the effect of the surface tension coefficient ε_C, the kinetic coefficient ε_V, coefficient of anisotropy A, crystalline symmetry k_A, latent heat of solidification H, and initial seed on the evolving crystal boundary.

1. *Smooth growth.* We begin with a calculation performed with surface tension coefficient $\varepsilon_C = 0.001$, kinetic coefficient $\varepsilon_V = 0.01$, no crystalline anisotropy (coefficient of anisotropy $A = 0$), and latent heat of solidification $H = 1$. The calculation were performed in a unit box, with a constant undercooling on the side walls of $u_B = -1$.

The initial shape was a perturbed circle with average radius $R = 0.15$, perturbation size $P = 0.08$, and $L = 4$ limbs. That is, the parametrized curve $(x(s), y(s))$, $0 \leq s \leq 1$, describing the initial position of the crystal is given by

$$(x(s), y(s)) = [(R + P \cos 2\pi L s)](\cos 2\pi s, \sin 2\pi s) \quad (7.1)$$

with $R = 0.15$, $P = 0.08$, and $L = 4$.

In Fig. 10, we show a series of calculations performed to study the effect of refining the grid size and time step on the computed solution. We begin in Fig. 10a with mesh size 48×48, and time step $\Delta t = 0.005$. The boundary grows smoothly outward from the initial limbs. Each limb is drawn towards its wall by the effect of the undercooling. The value of the kinetic coefficient ε_V is large enough to keep the evolving boundary smooth (in contrast to those calcula-
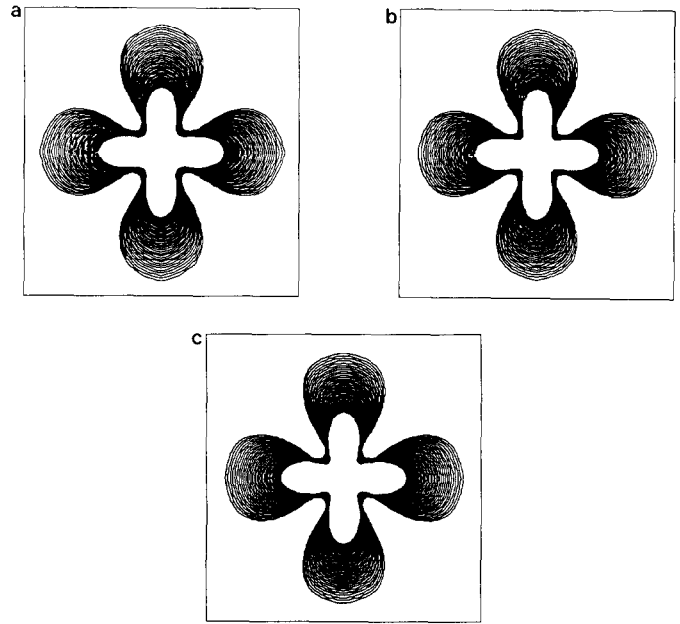

FIG. 10. Smooth crystal: Effect of refining both grid size and time step: (a) 48×48 mesh, $\Delta t = 0.005$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.01$, $\varepsilon_V = 0.01$, $k_A = 0$; (b) 64×64 mesh, $\Delta t = 0.0025$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.01$, $k_A = 0$; (c) 96×96 mesh, $\Delta t = 0.001$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$.

tions discussed below). In Fig. 10b we repeat the above

Fig. 10c, we take mesh size 96×96 and $\Delta t = 0.001$. The figures are unchanged, indicating the robustness of the algorithm.

Although the shape remains smooth as it evolves, several effects can be seen. First, while the undercooling pulls the limbs towards the walls, the other walls act to thicken the limb, creating a highly smoothed version of side-branching. Second, the tips of the limbs, where the curvature is positive, move very fast compared to the indented pockets between limbs, where the curvature is negative.

2. *Fingered growth.* Next, we perform a similar calculation, but change the kinetic coefficient to $\varepsilon_V = 0.001$. Once again, we take surface tension coefficient $\varepsilon_C = 0.001$, no crystalline anisotropy (coefficient of anisotropy $A = 0$), and latent heat of solidification $H = 1$. The constant undercooling on the side walls is -1, and the initial shape is again a perturbed circle with average radius 0.15, perturbation size 0.08, and four limbs, as in Eq. (7.1).

In Fig. 11, we show the results using these values for the physical parameters. We begin with a set of calculations performed with decreasing time step and fixed mesh size. In Fig. 11a, we show the evolution of $\Gamma(t)$ on a 48×48 mesh with time step $\Delta t = 0.005$. Starting from the smooth perturbed circle, the evolving boundary changes dramatically.

First, each limb flattens out. Then, tip splitting occurs as begins as each multi-tipped arm is both pulled toward the closest wall and also drawn by the walls parallel to the limb. In Fig. 11b, we perform the same calculation, but decrease the time step to $\Delta t = 0.0025$, maintaining the 48×48 mesh. In Fig. 11c, we perform the same calculation, but decrease the time step to $\Delta t = 0.001$, maintaining the 48×48 mesh. The position of the evolving boundary is unchanged under these time refinements, indicating the robustness of the numerical algorithm.

While the evolving boundary is complex, it should remain unchanged as both the space and time step are refined. In Fig. 12, we study the effects of such refinement, using the same physical parameters, that is, kinetic coefficient $\varepsilon_V = 0.001$, surface tension coefficient $\varepsilon_C = 0.001$, no crystalline anisotropy $(k_A = 0, A = 0)$, and latent heat of solidification $H = 1$, with constant undercooling -1. In Fig. 12a, we take a 32×32 mesh with $\Delta t = 0.005$. In Fig. 12b, we take a 48×48 mesh with $\Delta t = 0.005$. In Fig. 12c, we take a 96×96 mesh with $\Delta t = 0.00125$. In Fig. 12d, we take a 128×128 mesh with $\Delta t = 0.00125$. On the coarsest mesh (32×32), only the gross features of the fingering and tip splitting process are seen. As the numerical parameters are refined, the basic pattern emerges. It is clear that the resulting shapes are qualitatively the same, and there is little quantitative difference between Fig. 12c and
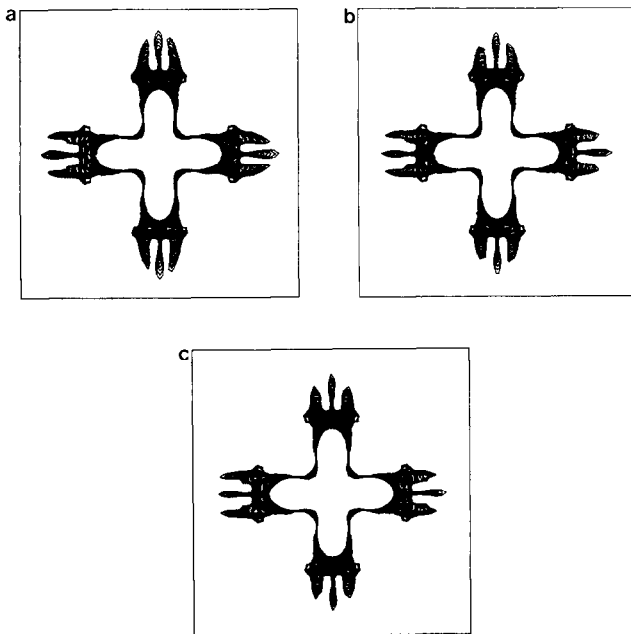
FIG. 11. Fingered crystal: Effect of refining time step: (a) 48×48 mesh, $\Delta t = 0.005$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$; (b) 48×48 mesh, $\Delta t = 0.0025$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$; (c) 48×48 mesh, $\Delta t = 0.001$, $H = 1.0$, $A - 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$.
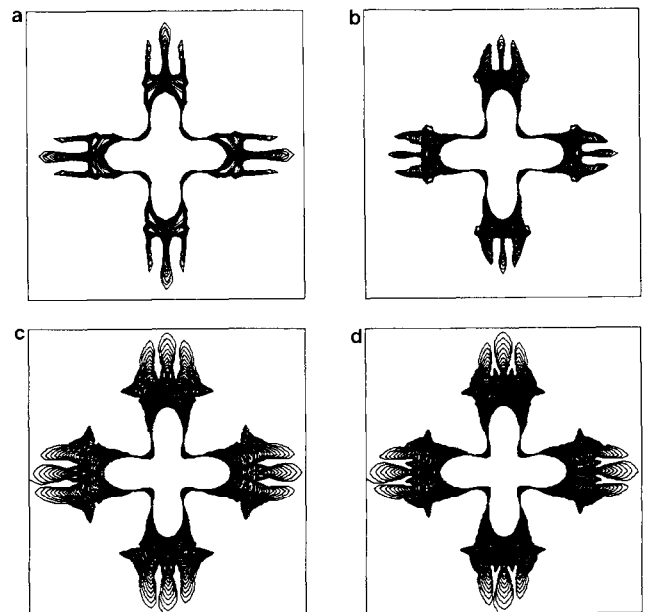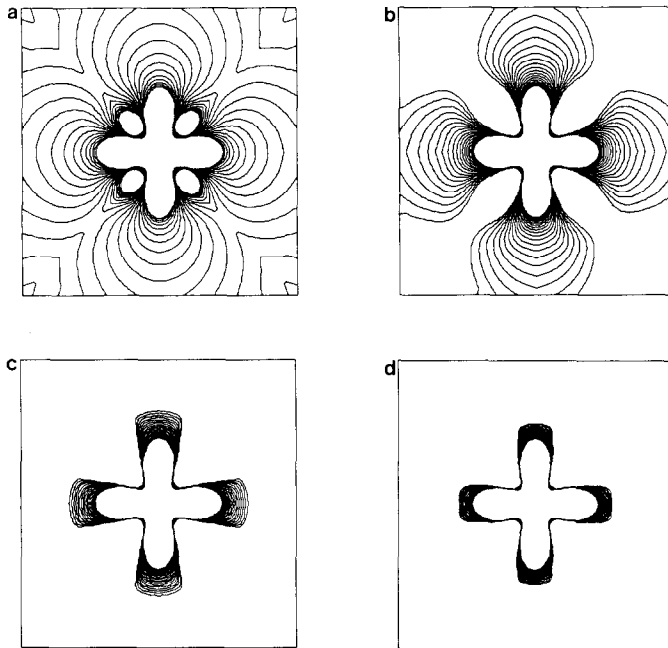
FIG. 12. Fingered crystal: Effect of refining both grid size and time step: (a) 32×32 mesh, $\Delta t = 0.005$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$; (b) 48×48 mesh, $\Delta t = 0.005$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$; (c) 96×96 mesh, $\Delta t = 0.00125$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$; (d) 128×128 mesh, $\Delta t = 0.00125$, $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$.
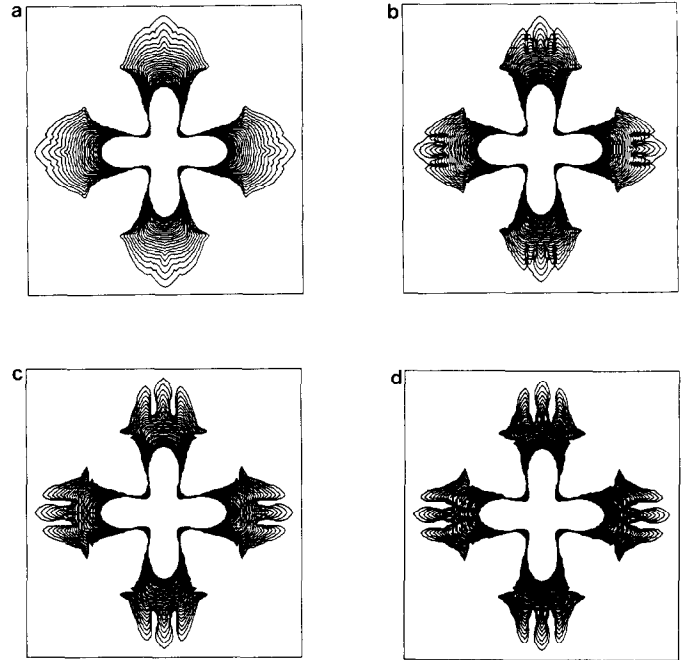
12d. Even when computing the highly complex boundaries seen in these figures, the algorithm remains robust.

3. *Latent heat of solidification.* Next, we study the effect of changing the latent heat of solidification H. Recall that H controls the balance between the pure geometric effects and the solution of the history-dependent heat integral. We consider the same initial shape and physical parameters as in Fig. 12 (again, $\varepsilon_V = 0.001$, $\varepsilon_C = 0.001$, $k_A = 0$, $A = 0$, with constant undercooling -1). In all calculations, we use a 96×96 mesh with time step $\Delta t = 0.00125$. However, we vary H smoothly: in Fig. 13a, $H = 0.25$, in Fig. 13b, $H = 0.5$, in Fig. 13c, $H = 0.75$, and in Fig. 13d, $H = 1.0$.

The evolving boundaries are all plotted at the same time, and the differences are substantial. With $H = 0.25$ (Fig. 13a), the dominance of geometric motion serves to create a rapidly evolving boundary that is mostly smooth. As the latent heat of solidification is increased, the growing limbs expand outwards less smoothly, and instead develop flat ends. As seen in Fig. 12, these flat ends are unstable and serve as precursors to tip splitting. We also note that the influence of the heat integral slows down the evolving boundary, as witnessed by the fact that all the plots are given at the same time. The mechanism operating here is presumably that increasing latent heat decreases the most unstable wavelength, as described by linear stability theory [19].

FIG. 14. Small-scale refinement of latent heat of solidification H: (a) $H = 0.75$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$; (b) $H = 0.833$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$; (c) $H = 0.916$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$; (d) $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$.

FIG. 13. Effect of smoothly varying latent heat of solidification H: (a) $H = 0.25$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$; (b) $H = 0.50$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$; (c) $H = 0.75$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$; (d) $H = 1.0$, $A = 0.0$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0$, 96×96, $\Delta t = 0.00125$.

To further understand the transition from smooth crystals to complex ones, in Fig. 14 we perform a small-scale refinement of the latent heat of solidification for values $H = 0.75$ (Fig. 14a), $H = 0.833$ (Fig. 14b), $H = 0.916$ (Fig. 14c), and $H = 1.0$ (Fig. 14d). In this set of figures, the evolving boundaries are *not* given at the same time. Instead, the final shape is plotted when $\Gamma(t)$ has reached to within 0.02 of the box walls (recall that the box has width 1.0). In these figures, we can see how fingering and tip splitting emerge from the essentially smooth shape given in Fig. 14a to the complex shape of Fig. 14d.
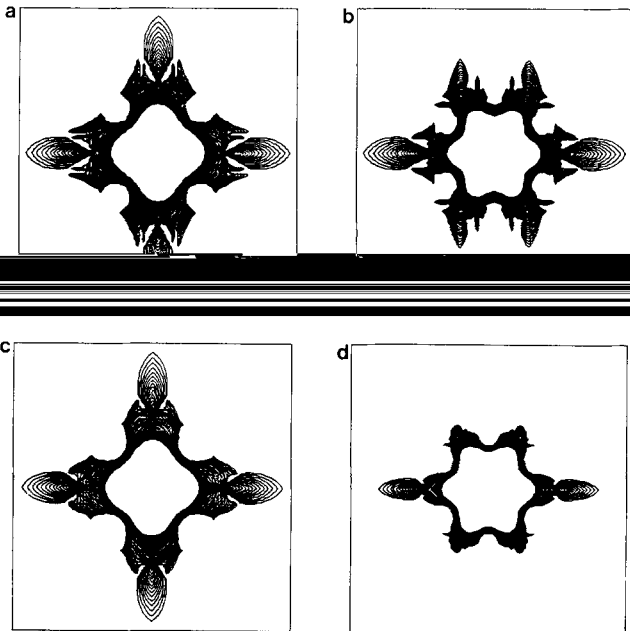
4. *Coefficient of anisotropy.* Next, we study the effect of changing the coefficient of anisotropy. We again start with the basic crystal shape given in (Eq. (7.1)), with the same choices for radius and perturbation. We take $\varepsilon_V = 0.001$, $\varepsilon_C = 0.001$, $H = 1.0$, a 96×96 mesh with time step $\Delta t = 0.00125$ and constant undercooling -1. In these calculations, we assume a fourfold anisotropy ($k_A = 4$). In Fig. 15a, we take the coefficient of anisotropy $A = 0.2$, in Fig. 15b, $A = 0.4$, in Fig. 15c, $A = 0.6$, and in Fig. 15b, $A = 0.08$. Our results show that increasing the coefficient of anisotropy has several effects. First, it increases the speed of propagating interface. Second, it controls the shape of the evolving finger from each limb. Third, it influences the amount of side branching that occurs.
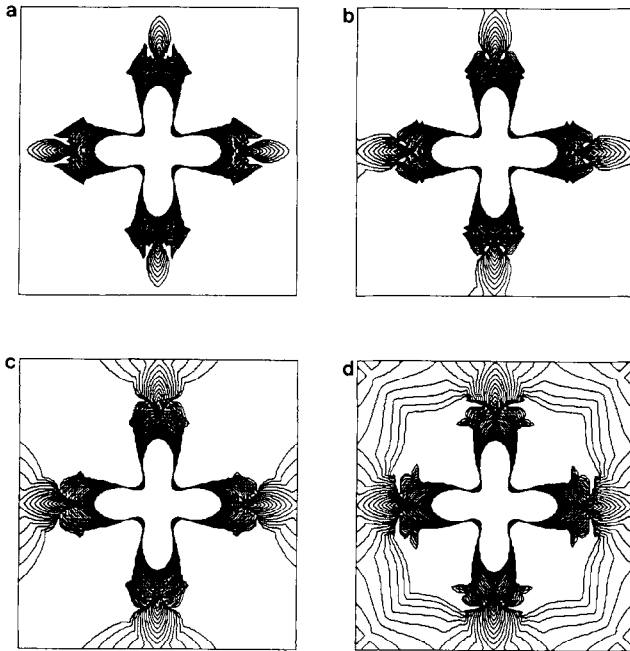
FIG. 15. Effect of changing coefficient of anisotropy A: (a) $A = 0.2$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (b) $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (c) $A = 0.6$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (d) $A = 0.8$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$, $H = 1.0$, 96×96, $\Delta t = 0.00125$.
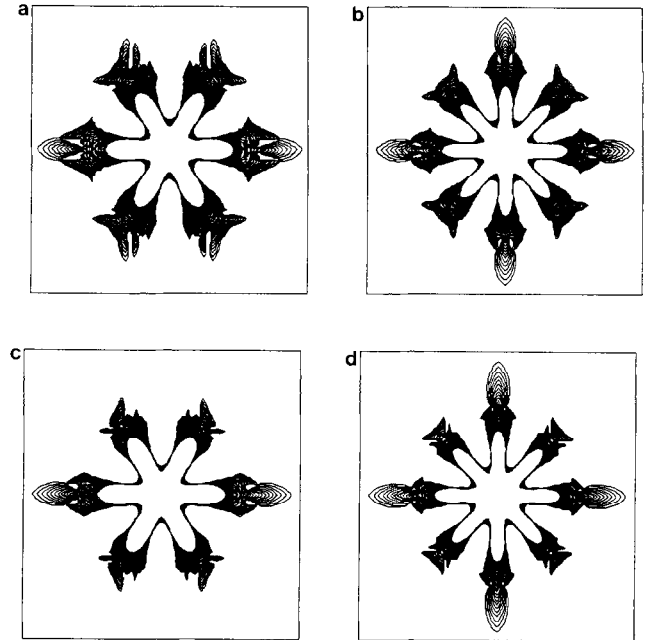
FIG. 17. Changing initial perturbation, anisotropy coefficient A, and crystalline symmetry k_A: (a) $L = 6$, $A = 0.0$, $k_A = 6$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (b) $L = 8$, $A = 0.0$, $k_A = 8$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (c) $L = 6$, $A = 0.4$, $k_A = 6$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (d) $L = 8$, $A = 0.4$, $k_A = 8$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$.
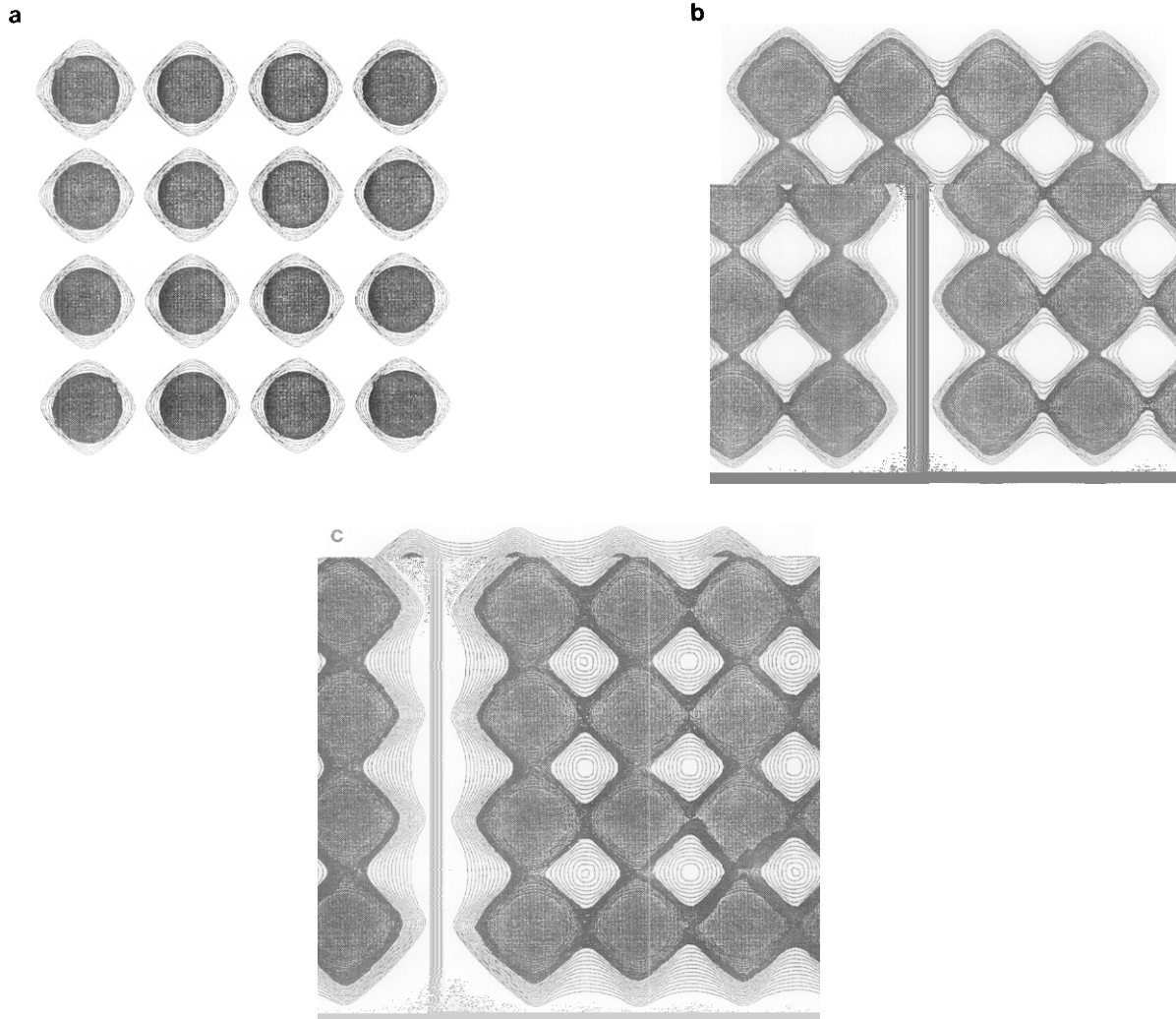
5. *Changing initial conditions, anisotropy coefficient A, and crystalline symmetry k_A.* In Fig. 16 and 17, we compute a collection of shapes by altering the initial seed, anisotropy coefficient, and degree of crystalline anisotropy. All calculations are performed using a 96×96 grid, with time step $\Delta t = 0.00125$, and $\varepsilon_V = 0.001$, $\varepsilon_C = 0.001$, $H = 1.0$, and constant undercooling -1.

take an initial shape with four limbs, fourfold crystalline anisotropy, and $A = 0.4$. In Fig. 16b, we take an initial shape with four limbs, sixfold crystalline anisotropy, and $A = 0.4$. In Fig. 16c, we take an initial shape with four limbs, fourfold crystalline anisotropy, and $A = 0.8$. In Fig. 16b, we take an initial shape with four limbs, sixfold crystalline anisotropy, and $A = 0.8$. The results depend dramatically on the choice of these values.

In Fig. 17, we concentrate on changing the anisotropy coefficient and the initial shape of the crystal. In Fig. 17a, we take an initial shape with six limbs, sixfold crystalline anisotropy, and $A = 0$. In Fig. 17b, we take an initial shape with eight limbs, eightfold crystalline anisotropy, and $A = 0$. In Fig. 17c, we take an initial shape with six limbs, sixfold crystalline anisotropy, and $A = 0.4$. In Fig. 17d, we take an initial shape with eight limbs, eightfold crystalline

FIG. 16. Changing initial perturbation, anisotropy coefficient A, and crystalline symmetry k_A: (a) $L = 4$, $A = 0.4$, $k_A = 4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (b) $L = 4$, $A = 0.4$, $k_A = 6$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (c) $L = 4$, $A = 0.8$, $k_A = 4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$; (d) $L = 6$, $A = 0.8$, $k_A = 6$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $H = 1.0$, 96×96, $\Delta t = 0.00125$.

FIG. 18. Array of crystals: $H = 0$, purely geometric motion: (a) Time = 0.0–0.025, 96 × 96 mesh, $\Delta t = 0.00125$, $H = 0.0$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$; (b) Time = 0.025–0.05, 96 × 96 mesh, $\Delta t = 0.00125$, $H = 0.0$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$; (c) Time = 0.05–1.0, 96 × 96 mesh, $\Delta t = 0.00125$, $H = 0.0$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$.

anisotropy, and $A = 0.4$. The resulting shapes display a variety of intricate behavior.

6. Arrays of Crystals. To demonstrate the versatility of our method, in Fig. 18 and 19 we consider an array of crystals. In both cases, we consider a 96 × 96 mesh, with time step $\Delta t = 0.00125$. In Fig. 18, we show the case of purely geometric motion ($H = 0$, and motion according to Eq. (4.1)), with $\varepsilon_C = 0.01$, $\varepsilon_V = 0.00$. We consider a 4 × 4 array of crystals, and take fourfold anisotropy, with anisotropy coefficient $A = 0.4$. We plot the figure at various times, as shown in Figs. 18a, 18b, and 18c. In each of these figures, we show several time levels of the moving boundary, as well as the solid crystal at a early time, depicted by shading in those points for which $\phi > 0$.

To begin with, the crystals evolve under the influence of the fourfold anisotropy. Since the motion of each crystal is

determined by purely geometric effects, the presence of neighboring crystals is not known until collision occurs. Once neighboring boundaries meet, they all join together, leaving only a collection of pockets which soon collapse. The complexity of this problem underscores the ability of the level set approach to track involved topological changes.

In Fig. 19, we perform a similar calculation, but this time we turn on the full effects of the equations of motion. We take $H = 0.75$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, with fourfold anisotropy and coefficient of anisotropy $A = 0.4$. Again, we start with a 4 × 4 array of initially circular crystals. In this case, the evolving crystals are quite different. Neighboring crystals "hear" each other due to the temperature boundary conditions on each interface. Thus, we see that the exterior edge of crystals grow towards the walls, under the influence of the undercooling and the anisotropic effects. However, growth towards the center, and in particular for the interior
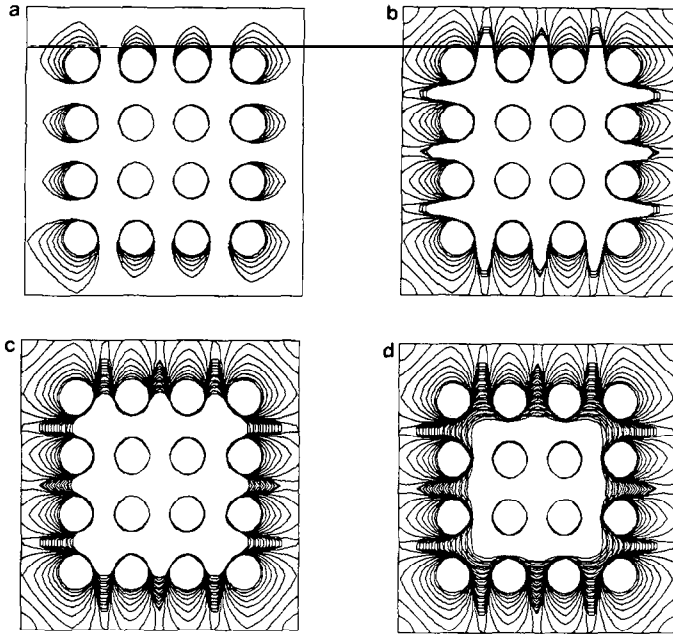
FIG. 19. Array of crystals: $H = 0.75$, full effects: (a) Time = 0.0025, 48×48 mesh, $\Delta t = 0.005$, $H = 0.75$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$; (b) Time = 0.005, 48×48 mesh, $\Delta t = 0.005$, $H = 0.75$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 0.4$; (c) Time = 0.01, 48×48 mesh, $\Delta t = 0.005$, $H = 0.75$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$; (d) Time = 0.015, 48×48 mesh, $\Delta t = 0.005$, $H = 0.75$, $A = 0.4$, $\varepsilon_C = 0.001$, $\varepsilon_V = 0.001$, $k_A = 4$.

crystals, is severely limited, as the release of latent heat by the neighbors warms the liquid.

The calculations presented above are all performed in a square container. The influence of the box walls is significant on the growing front, and it would be interesting to extend our work to other bounding containers, as well as infinite domains. The extension to other bounding regions is not overly difficult; our focus in this paper has been to explain and test the fundamental algorithm. The extension of our approach to problems in infinite domains, while not as straightforward, would be valuable.

8. CONCLUSION

In this paper, we have presented a numerical algorithm for computing the motion of complex solid/liquid interfaces in crystal growth. Our model includes such physical effects as crystalline anisotropy, surface tension, molecular kinetics, and undercooling. Our algorithm relies on a boundary integral formulation of the equations of motion coupled to a level set method for advancing the propagating interface. The method is able to follow the evolution of extremely intricate shapes, exhibiting complex behavior such as fingering, tip splitting, and side branching, as well as profound topological changes. In addition, our numerical technique is shown to be highly robust, in that refinement of

the numerical parameters produces a converged solution, even for the most complex interfaces. In [35], we use our algorithm to generate crystal boundaries under a wide range of conditions. Finally, the method presented here generalizes in an obvious manner to three space dimensions. We shall report on that work elsewhere [36].

ACKNOWLEDGMENTS

REFERENCES

1. M. Ben Amar and Y. Pomeau, *Europhys. Lett.* **2**, 3307 (1986).
2. L. N. Brush and R. F. Sekerka, *J. Cryst. Growth*, to appear.
3. G. Caginalp and P. C. Fife, *SIAM J. Appl. Math.* **48**, 506 (1988).
4. J. W. Cahn and J. E. Hilliard, *J. Chem. Phys.* **28**, 258 (1958).
5. J. Chadam and P. Ortoleva, *IMA J. Appl. Math.* **30**, 57 (1983).
6. B. Chalmers, *Principles of Solidification* (Wiley, New York, 1964).
7. A. J. Chorin, *J. Comput. Phys.* **57**, 472 (1985).
8. J. Duchon and R. Robert, Evolution d'une Interface par Capillarite et, Diffusion de Volume I. Existence Locale en Temps, *Ann. Inst. Henri Poincare* **1**, 361 (1984).
9. H. Dym and H. P. McKean, *Fourier Series and Integrals* (Academic Press, New York, 1972).
10. L. C. Evans and J. Spruck, *J. Diff. Geom.* **33**, 635 (1991).
11. G. J. Fix, "Numerical simulation of free boundary problems using phase field methods," in *The Mathematics of Finite Elements and Applications IV (Uxbridge, 1981)*, p. 265 (Academic Press, London/New York, 1982).
12. L. Greengard and J. Strain, *Commun. Pure Appl. Math* **43**, 949 (1990).
13. M. E. Gurtin, *Arch. Ration Mech. Anal.* **96**, 199 (1986).
14. M. Kac, *Am. Math. Mon.* **73**, No. 4, 1 (1966).
15. A. Karma, *Phys. Rev. Lett.* **57**, 858 (1986).
16. F. X. Kelly and L. H. Ungar, *Phys. Rev. B* **34**, 1746 (1986).
17. D. A. Kessler and H. Levine, *Phys. Rev. Lett.* **57**, 3069 (1986).
18. J. S. Langer, *Phys. Rev. A*, **33**, p. 435–441, 1986.
19. J. S. Langer, *Rev. Mod. Phys.* **52**, (1980).
20. P. D. Lax, *Commun. Pure Appl. Math.* **10**, 537 (1957).
21. V. Laxmanan, *Acta Metall.* **33**, 1023 (1985).
22. D. I. Meiron, *Physica D* **23**, 329 (1986).
23. G. H. Meyer, *SIAM J. Numer. Anal.* **10**, 552 (1973).
24. W. Mulder, S. J. Osher, and J. A. Sethian, *J. Comput. Phys.*, to appear.
25. W. W. Mullins and R. F. Sekerka, *Appl. Phys.* **34**, 323 (1963).
26. W. W. Mullins and R. F. Sekerka, *J. Appl. Phys.* **35**, 444 (1964).
27. G. E. Nash and M. E. Glicksman, *Acta Metall.* **22**, (1974).
28. G. E. Nash and M. E. Glicksman, *Acta Metall.* **22**, 1291 (1974).
29. S. J. Osher, *SIAM J. Numer. Anal.* **21**, 27 (1984).
30. S. J. Osher and J. A. Sethian, *J. Comput. Phys.* **79**, 12 (1988).
31. V. K. Pinus and P. L. Taylor, *Phys. Rev. B* **32**, 5362 (1985).
32. J. A. Sethian, *Commun. Math. Phys.* **101**, 4 (1985).
33. J. A. Sethian, *J. Diff. Geom.* **31**, 131 (1989).

34. J. A. Sethian and D. L. Chopp, "Singularity Formation and Minimal Surfaces," *J. Diff. Atom.*, to appear.

34. J. A. Sethian and J. Strain, "Numerical Simulation of Complex Crystal Growth," *J. Diff. Geom.*, to appear.

35. J. A. Sethian and J. Strain, "Three-Dimensional Crystal Growth" (unpublished).

36. J. B. Smith, *J. Comput. Phys.* **39**, 112 (1981).

37. J. Strain, *J. Comput. Phys.* **85**, 342 (1989).

38. J. Strain, Berkeley Center for Pure and Applied Mathematics Report, PAM-494, April 1990; *J. Comput. Phys.*, submitted.

39. J. Strain, *Physica D* **30**, 297 (1988).

40. J. Strain, *SIAM J. Appl. Math.* **50**, 1 (1990).

41. J. Strain, "A Fast Boundary Integral Method for Crystal Growth" (unpublished).

42. J. M. Sullivan, D. R. Lynch, and K. O. O'Neill, *J. Comput. Phys.* **69**, 81 (1987).

43. C. Verdi and A. Visintin, *Boll. U.M.I.* (7) **I-B**, 795 (1987).

44. D. J. Wollkind and R. D. Notestine, *IMA J. Appl. Math.* **27**, 85 (1981).