Journal of Computational Physics **173,** 765–796 (2001)
doi:10.1006/jcph.2001.6911, available online at http://www.idealibrary.com on IDEAL®

Front-Tracking Finite Element Method for Dendritic Solidification

P. Zhao and J. C. Heinrich

Department of Aerospace and Mechanical Engineering, The University of Arizona, Tucson, Arizona 85721
E-mail: pzhao@u.arizona.edu; heinrich@zeus.ame.arizona.edu

Received February 9, 2001; revised July 16, 2001

A two-dimensional finite element method capable of tracking sharp interfaces is developed. The method is based on a fixed mesh of bilinear isoparametric elements and is extremely simple and easy to use. The interfaces are tracked with a set of marker points that define their position at all times. Several different approaches to finding the interface position and velocity are discussed, and their effect on the convergence rate is examined. It is shown through numerical examples that the approximation to the interface position converges with second-order accuracy in two-dimensional problems, as opposed to previously developed finite difference algorithms that are only first-order accurate. The approximation to the interface velocity is shown to be of first order. A number of examples are examined, including several that provide a careful comparison with previously published results.

Key Words: interface tracking; dendritic solidification; finite element method.

1. INTRODUCTION

The simulation of crystal growth into an undercooled liquid has attracted much interest for a number of years because of the technological importance of understanding and controlling solidification processes in the production of advanced materials in the aerospace and semiconductors industries [1–10]. The numerical modeling of problems with phase change started over 50 years ago, when the first attempts at modeling stable melting/solidification—also known as the Stefan problem—were made [11, 12]. An excellent historical review is given in Juric and Tryggvason [8]; further extensive discussions and literature reviews in this topic can be found in Crivelli and Idelson [13], Floryan and Ramussen [14], Voller *et al.* [15], Udaykumar *et al.* [10], Udaykumar and Shyy [16], and Provatas *et al.* [9].

The presence of an interface between two phases, which in this work is assumed to be solid and liquid, complicates the solution of an otherwise manageable problem. The interface

0021-9991/01 $35.00

position must be calculated as part of the solution process, and interface conditions must be satisfied at this interface. This results in a highly nonlinear problem that is very sensitive to numerical error and prone to numerical instability. The early numerical models of melting/solidification were developed for the stable Stefan problem and avoided the actual calculation of the interface position, while still being able to calculate the thermal field [17–20]. However, when dealing with unstable solidification, as is the case with a crystal growing into an undercooled liquid, it is necessary to accurately calculate the interface position. This led to the development of diffuse-interface models, of which the phase-field method is the most widely used [6, 7, 9, 21–24]. In the phase-field method, a continuous transition between the two phases is introduced across a thin layer of finite thickness and an additional variable identifying the phase, called the phase field, is calculated. The disadvantage of this method is that it is difficult to relate the parameters in the equation for the field variable to physical parameters such as the surface tension. Furthermore, results depend on the prescribed interface thickness, which is required to be extremely small for accurate calculations [6, 7]. An excellent review of the phase-field methods can be found in [25].

Boundary integral methods, in which the problem is cast in the form of an integral representation of the interface, were developed next [4, 5, 26]. Even though these methods have been used to successfully simulate very complex solid–liquid interfaces, they are extremely complex mathematically and are computationally expensive. Most of these methods are also limited to problems with homogeneous material properties.

More recently, methods tracking the sharp interface have been developed that are very general in that they can handle the discontinuous properties at the interface and they provide the explicit location of the interface at all times. Sullivan *et al.* [2] and Palle and Dantzig [27] developed finite element schemes based on adaptive meshes in such a way that the interface is always described by nodal mesh points that move with the interface. On the other hand, Juric and Tryggvason [8], Udaykumar *et al.* [10], and Udaykumar and Shyy [16] developed finite difference models based on a fixed mesh, where the interface is tracked using a set of markers that define its position and shape. The method in [8] is not quite a sharp interface method since it uses a distribution function to represent the temperature and the heat source term at the interface but uses markers to follow the position of the interface itself. The method in [10] is strictly a sharp interface method. These methods successfully modeled unstable solidification of pure substances under a variety of different conditions, such as discontinuous conductivities, anisotropic surface tension, and kinetic mobility, and different symmetry modes. The moving mesh method in [2] was shown to be second-order accurate [28]; however, the approximation to the interface location in [8] and [10] is only first-order accurate.

In this work, a finite element method for front tracking is developed that is based on a fixed mesh that follows the interface in a way similar to the methods of Juric and Tryggvason [8] and Udaykumar *et al.* [10]. The method is extremely simple and restricts all calculations—except for the evaluation of the gradients in the direction normal to the interface—to the element level. Moreover, it will be shown that this method exhibits second-order convergence rates for the interface location.

The governing equations are introduced in the next section, and the numerical algorithm is presented in Section 3. The approximation errors are discussed in Section 4, and examples and further discussion are given in Section 5. Conclusions follow in Section 6.

2. PROBLEM FORMULATION

The solidification of a pure substance is addressed by assuming that conduction is the only mechanism of heat transport. The energy equation is given by

$$\eta \frac{\partial T_S}{\partial t} = \Lambda \nabla^2 T_S \quad \text{in the solid phase} \tag{1a}$$

and

$$\frac{\partial T_L}{\partial t} = \nabla^2 T_L \quad \text{in the liquid phase.} \tag{1b}$$

The conditions at the solid–liquid interface are

$$T_S = T_L = T_I \tag{2a}$$

$$\Lambda \frac{\partial T_S}{\partial n} - \frac{\partial T_L}{\partial n} = (1 - \gamma T_I) V. \tag{2b}$$

In this work, the subscripts S and L denote the solid and liquid phases, respectively. The equations are made nondimensional using a reference length H, the reference time $\tau = H^2/\alpha_L$, and the nondimensional temperature $T = c_{PL}(T^* - T_m^*)/L$, where $(^*)$ indicates a dimensional quantity. The parameters in (1) and (2) are $\eta = \rho_S c_{PS}/\rho_L c_{PL}$, $\Lambda = \kappa_S/\kappa_L$, $St = c_{PL}(T_\infty^* - T_m^*)/L$, and $\gamma = (c_{PS} - c_{PL})/c_{PL}$. In the above, ρ denotes density, c_P is specific heat, κ is thermal conductivity, T_m is the melting temperature for a planar interface, T_∞ is the far-field temperature, L is the latent heat, $V = (d\vec{S}_I/dt) \cdot \hat{n}$ is the nondimensional local interface velocity, $\vec{S}_I$ is the interface, and T_I is the local nondimensional interface temperature. T_I is given by the Gibbs–Thompson equation [8, 29]

$$T_I - \gamma T_I^2 + \sigma\kappa + \mu V = 0, \tag{3}$$

where κ is the local interface curvature, σ is the surface tension, and μ is the kinetic mobility (all nondimensional). The last two quantities may vary locally to account for anisotropic effects. In (2b), the normal derivative is defined as

$$\frac{\partial T}{\partial n} = \hat{n} \cdot \nabla T, \tag{4}$$

where $\hat{n}$ denotes the unit vector normal to the interface and pointing toward the liquid phase.

The boundary conditions associated with (1) will be of either the Dirichlet type, i.e.,

$$T(\vec{x}, t) = b(\vec{x}, t) \quad \text{for } \vec{x} \in \Gamma_D, \tag{5a}$$

or the Neumann type,

$$-\varepsilon \frac{\partial T}{\partial n}(\vec{x}, t) = q(\vec{x}, t) \quad \text{for } \vec{x} \in \Gamma_N, \tag{5b}$$

where the subscripts D and N denote, respectively, the portions of the boundary where a

Dirichlet or Neumann boundary condition is prescribed, $\varepsilon = \Lambda$ in the solid region, and $\varepsilon = 1$ in the liquid region. Mixed or Robins boundary conditions are also readily incorporated in the model, but they are not discussed here for the sake of brevity. The nondimensional form above offers the advantage that, in a given region where the far-field temperature is uniform, the Stefan number becomes the Dirichlet boundary condition.

Finally, a set of initial conditions is needed for the temperature and interface position:

$$T(\vec{x}, 0) = T_0(\vec{x}), \quad \vec{S}_I(\vec{x}, 0) = \vec{S}(\vec{x}). \tag{6a,b}$$

This completes the formulation of the problem.

3. FINITE ELEMENT METHOD

The two-dimensional model will be described. In three dimensions, the procedures are the same, with the obvious added complexities derived from the topology.

The discretization is based on a Galerkin formulation using the four-node isoparametric bilinear element. Without loss of generality, the discussion is restricted to the case of a rectangular domain, Ω, subdivided by a uniform rectangular mesh. It is also assumed that an element can be intersected by an interface in only two basic ways (Fig. 1). The element is subdivided into a triangle and a pentagon in the first case (Fig. 1a) and into two quadrilaterals in the second case (Fig. 1b). It will become clear later in Section 3.4 that the resolution that can be achieved with this method as two interfaces advance toward each other is of the same magnitude as the mesh size. Therefore, if an element is intersected by more than one interface segment, it is beyond the mesh resolution capabilities. If resolution at a smaller scale is required, the mesh must be refined.

The weak form of (1) and (2) is as follows: Given the initial conditions (6) and a set of markers that define the interface position S_I in $C^0(\Omega)$, find a temperature field T among all

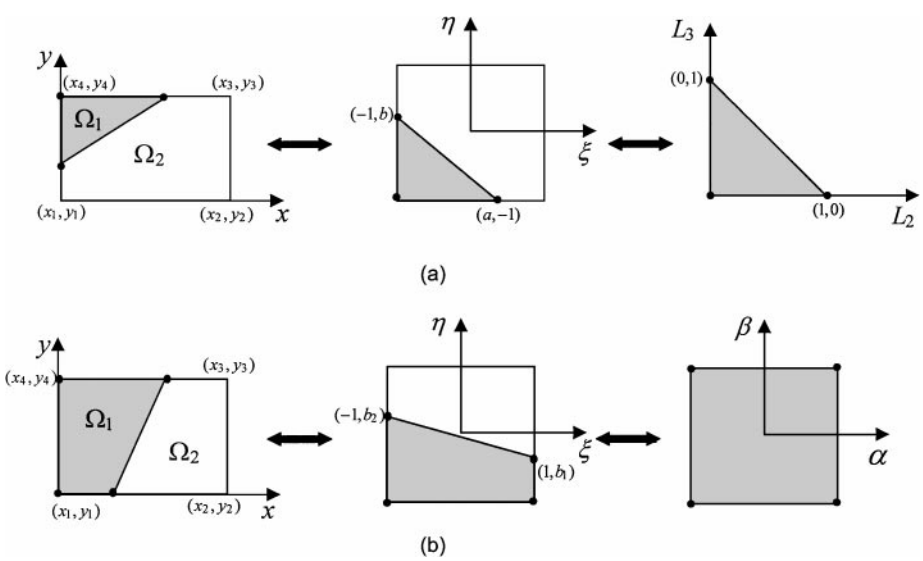


FIG. 1. Transformation of a portion of an intersected element into a standard form. (a) Triangle-pentagon intersect. (b) Biquardrilaterial intersect.

test functions in the space $H_D^1(\Omega)$ such that the expression

$$\int_\Omega \left(w\delta \frac{\partial T}{\partial t} + \varepsilon \nabla w \cdot \nabla T \right) d\Omega = \int_{S_I} w\,[1 - \gamma T_I]\, V\, dS_I + \int_{\Gamma_N} wq\, d\Gamma_N \tag{7}$$

is satisfied for all weighting functions w in the space $H_0^1(\Omega)$, and (2b) is satisfied at all marker points defining S_I.

In (7), $\delta = \eta$ in the solid region, $\delta = 1$ in the liquid, S_I denotes the interface, and $C^0(\Omega)$ is the set of all continuous functions defined in Ω. The set $H_D^1(\Omega)$ and the space $H_0^1(\Omega)$ are, respectively, the subset of the space $H^1(\Omega)$ consisting of functions that satisfy the Dirichlet boundary conditions (5a) and the subspace of $H^1(\Omega)$ consisting of functions that satisfy homogeneous boundary conditions on Γ_D. $H^1(\Omega)$ is the standard Sobolev space of functions that are square integrable, together with their first partial derivatives over Ω.

3.1. Galerkin Formulation

The semi-discrete Galerkin form of (7) is obtained by replacing the space $H^1(\Omega)$ with the finite-dimensional space of piecewise bilinear functions, $S^N(\Omega)$, defined over the finite element mesh of N nodes. Over each element,

$$T(x, y, t) = \sum_{j=1}^{4} N_j(x, y) T_j(t), \tag{8}$$

and, setting $w_i = N_i$, the semi-discrete Galerkin form is as follows: Find T in $S^N(\Omega)$ such that

$$\sum_{j=1}^{N} \left\{ \left[\int_\Omega \delta N_i N_j\, d\Omega \right] \dot{T}_j + \left[\int_\Omega \varepsilon \nabla N_i \nabla N_j\, d\Omega \right] T_j \right\} = \int_{S_I} N_i\,[1 - \gamma T_I]\, dS_I + \int_{\Gamma_N} N_i q\, d\Gamma_N \tag{9}$$

is satisfied for $i = 1, 2, \ldots, N$, where globally $T = \Sigma_{j=1}^{N} N_j(x, y) T_j(t)$. The discretization above yields a system of ordinary differential equations in time of the form

$$M\dot{T} + KT = F\,, \tag{10}$$

where the mass matrix M and the stiffness matrix K are given by

$$m_{ij} = \int_\Omega \delta N_i N_j\, d\Omega \quad \text{and} \quad k_{ij} = \int_\Omega \varepsilon \nabla N_i \nabla N_j\, d\Omega. \tag{11}$$

The right-hand side vector F contains the contributions of the line integrals and the Dirichlet boundary conditions.

The system of equations (10) is solved using the θ-method [30, p. 260]. The final system of linear algebraic equations takes the form

$$(M + \theta \Delta t K) T^{n+1} = (M + (\theta - 1)\Delta t K) T^n + [(1 - \theta) F^n + \theta F^{n+1}] \Delta t, \tag{12}$$

where the superscript n denotes the time level and θ is a user-defined parameter. In this work, the value $\theta = 0.5$, which corresponds to the second-order-in-time Crank–Nicolson–Galerkin method, has been used exclusively. Notice that the matrices M and K are rebuilt

at every iteration because of the motion of the interface. The system of linear algebraic equations (12) has been solved using a conjugate gradient iterative method.

3.2. Element Matrices

For elements that are completely contained within either the solid or liquid phase, the matrices above are readily calculated from (11), and the line integral over S_I is zero. Details of how the matrices are constructed for elements that are intersected by an interface are presented below.

3.2.1. *Mass and Stiffness Matrices*

In order to calculate the mass matrix of elements intersected by an interface, two isoparametric transformations are defined that map one portion of the element into a standard form (Fig. 1). These transformations are simple and easy to evaluate; for example, for the case in Fig. 1a, they are

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2}\begin{bmatrix} (1+\eta)\Delta x \\ (1-\xi)\Delta y \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} (1+a)L_2 - 1 \\ (1+b)L_3 - 1 \end{bmatrix}.$$

For the situation in Fig. 1b, they are

$$\begin{bmatrix} x \\ y \end{bmatrix} = \frac{1}{2}\begin{bmatrix} (1+\eta)\Delta x \\ (1-\xi)\Delta y \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} \alpha \\ \frac{1}{4}(1+\beta)[b_1 + b_2 + \alpha(b_1 - b_2)] - \frac{1}{2}(1-\beta) \end{bmatrix}.$$

Notice that the first transformation is a simple rotation and the second one always has the same form. The integrals over Ω_1 become

$$\int_{\Omega_1} F(x, y)\, dx\, dy = \int_0^1 \int_0^1 F(L_1, L_2)\frac{1}{4}(1+a)(1+b)\Delta x \Delta y\, dL_1\, dL_2 \tag{13}$$

when Ω_1 is a triangle, and

$$\int_{\Omega_1} F(x, y)\, dx\, dy = \int_0^1 \int_0^1 F(\alpha, \beta)\frac{1}{16}(\zeta + 2)\Delta x \Delta y\, dL_1\, dL_2 \tag{14}$$

when Ω_1 is a quadrilateral, where $\zeta = b_1 + b_2 + \alpha(b_1 - b_2)$. The integrals of the form in (13) are evaluated numerically using six integration points in the triangle; the ones of the form in (14) are calculated with a 3×3 Gauss quadrature. The elements of the mass matrix are given by

$$m_{ij} = \int_{\Omega} N_i N_j\, d\Omega + (\eta - 1)\int_{\Omega_1} N_i N_j\, d\Omega_1. \tag{15}$$

For the stiffness matrix, the expressions for the integral over Ω_1 become somewhat more involved because of the presence of the derivatives, but they can still be obtained in closed form. For the triangular cut, the derivatives are given by

$$\frac{\partial N_i}{\partial x} = \frac{2}{(1+b)\Delta x}\frac{\partial N_i}{\partial L_3} \quad \text{and} \quad \frac{\partial N_i}{\partial y} = -\frac{2}{(1+a)\Delta y}\frac{\partial N_i}{\partial L_2}, \tag{16}$$

and, for the quadrilateral cut, by

$$\frac{\partial N_i}{\partial x} = \left(\frac{8}{(\zeta+2)\Delta x}\right)\frac{\partial N_i}{\partial \beta} \quad \text{and} \quad \frac{\partial N_i}{\partial y} = -\frac{2}{\Delta y}\left(\frac{\partial N_i}{\partial \alpha} - \frac{(b_1-b_2)(1+\beta)}{(\zeta+2)}\frac{\partial N_i}{\partial \beta}\right). \tag{17}$$

The stiffness matrix is obtained from

$$k_{ij} = \int_\Omega \left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right) d\Omega + (\Lambda - 1)\int_{\Omega_1}\left(\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y}\right) d\Omega_1. \tag{18}$$

The same integration quadratures as for the mass matrix are used to evaluate the stiffness matrices.

Finally, we should mention that approximating the interface segment by a straight line within the element is consistent with the bilinear formulation and preserves the overall second-order accuracy in the approximation. An additional $O(h^2)$ error is introduced in the case of the triangular cut in Fig. 1a.

3.2.2. *Latent Heat Source Term*

A weakness of previously proposed models has been their need to distribute the latent heat produced at the interface over nodes away from the interface location, essentially smearing the interface. Full advantage of finite element methodology is taken, using the shape functions to integrate the source term exactly along the interface segment intersecting an element, and assigning the contributions to the element nodes. Consider the element shown in Fig. 2, which is intersected as shown. It is assumed that the interface segment within the element is always a straight line. The contribution of the latent heat generated on the interface segment to each node in the element is

$$Q_i = \int_{S_I} N_i(1-\gamma T_I)V\, dS_I. \tag{19}$$

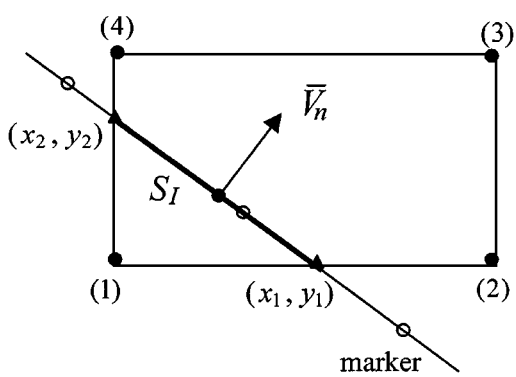


FIG. 2. Latent heat calculation and distribution to element nodes. (x_1, y_1) and (x_2, y_2) are the two intersections between the interface and the element boundaries. S_I denotes the interface segment, and $\bar{V}_n$ is the average velocity of the segment.

To evaluate the integral in (19), S_I is expressed parametrically as

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \begin{bmatrix} (x_2 - x_1)s + x_1 \\ (y_2 - y_1)s + y_1 \end{bmatrix} \tag{20}$$

and (19) is rewritten as

$$Q_i = \int_0^1 N_i(x(s), y(s))(1 - \gamma \bar{T}_I)\bar{V} L_I \, ds, \tag{21}$$

where $\bar{T}$ and $\bar{V}$ are, respectively, the average temperature and velocity of the interface segment and

$$L_I = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

is its length. Because the shape functions are a quadratic polynomial in s, they are integrated exactly using Simpson's rule or a two-point Gauss quadrature. The discretization of the source term in one dimension using a linear approximation involves an error of fourth order, hence, the largest error is introduced by the polygonal approximation of the interface and is $O(h^2)$ [31, pp. 50 and 196].

3.3. Interface Tracking

The interface is described using a set of marker points located on the interface, as done in [8, 10]. If the shortest side of a uniform rectangular grid is denoted by h, the distance d between any two adjacent markers is controlled so that $0.3h < d < h$. If the distance between any two adjacent markers becomes larger than h, another marker is added in between. If d becomes shorter than $0.3h$, then a marker is removed, provided that the distance between the newly adjacent markers does not exceed h; otherwise, a marker is relocated halfway between its neighbors.

To calculate the curvature and the unit normal to the interface, a local quadratic interpolant centered at each interior marker is constructed. For markers lying on the boundary of the domain, the condition that the interface intersect the boundary at an 90° angle is imposed, and symmetry is used to construct a quadratic interpolant centered at this point. More sophisticated interpolants, such as cubic and cubic splines, have also been utilized with no measurable improvement in the solutions. This agrees with the conclusions of Chorin [32] and Udaykumar *et al.* [10], who also reported that nothing is gained by utilizing higher-order approximations to the interface.

Using the arc length coordinate as shown in Fig. 3,

$$\begin{bmatrix} x(s) \\ y(s) \end{bmatrix} = \frac{(s - s_2)(s - s_3)}{(s_1 - s_2)(s_1 - s_3)} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \frac{(s - s_1)(s - s_3)}{(s_2 - s_1)(s_2 - s_3)} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \frac{(s - s_1)(s - s_2)}{(s_3 - s_1)(s_3 - s_2)} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}, \tag{22}$$

where $s_2 = \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2}$ and $s_3 = s_2 + \sqrt{(x_{i+1} - x_i)^2 + (y_{i+1} - y_i)^2}$.

The expressions for the curvature and the unit normal at the midpoint, $s = s_2$, are

$$\kappa = \frac{y_s(s_2)x_{ss} - x_s(s_2)y_{ss}}{S^3} \tag{23}$$

$$\hat{n} = \frac{y_s(s_2)\hat{i} - x_s(s_2)\hat{j}}{S}, \tag{24}$$

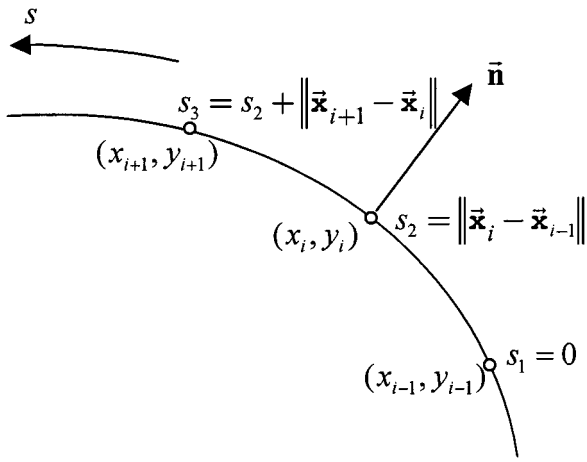


FIG. 3. Local arc length coordinates to calculate curvature and direction of the normal to the interface.

with

$$S = \sqrt{(x_s(s_2))^2 + (y_s(s_2))^2}.$$

The derivatives are readily calculated from (22). Using the fact that $s_1 = 0$,

$$\begin{bmatrix} x_s(s_2) \\ y_s(s_2) \end{bmatrix} = \frac{s_2 - s_3}{s_2 s_3} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \frac{2s_2 - s_3}{s_2(s_2 - s_3)} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \frac{s_2}{s_3(s_3 - s_2)} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix} \tag{25a}$$

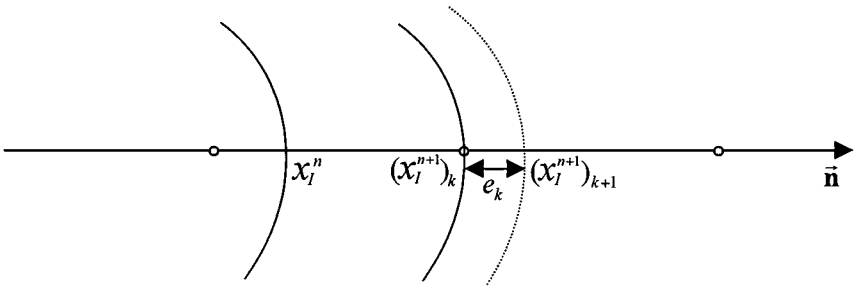
$$\begin{bmatrix} x_{ss} \\ y_{ss} \end{bmatrix} = \frac{2}{s_2 s_3} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} + \frac{2}{s_2(s_2 - s_3)} \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} + \frac{2}{s_3(s_3 - s_2)} \begin{bmatrix} x_3 \\ y_3 \end{bmatrix}. \tag{25b}$$

3.4. Interface Position and Velocity

The method used to determine the position and velocity of the interface is the most critical aspect of this kind of method because it determines the order of convergence of the numerical approximation to the interface position (convergence will be discussed in detail in the next section). These quantities are obtained simultaneously from the interface condition (2b). Two basic approaches have been used, both of which involve an iteration that uses the velocity at the last time step to estimate the next position of the interface; that is, the initial guesses are $(x_I^{n+1})_0 = x_I^n + \Delta t V^n$ and $(V^{n+1})_0 = V^n$. Subsequent approximations follow one of the two methods described below.

Direct approximation. In this method the jump in the heat flux at the interface is evaluated directly from the currently assumed interface position and the resulting temperature field. A new interface velocity is then calculated directly from (2b), and a new estimate of the interface position is derived by multiplying the average of the velocity at the last time step and the new velocity by the time step. This is the simplest way available to update the surface position.

Incremental method. In this method, (2b) is used to update the interface position rather than the velocity. Figure 4 shows a schematic and the notation in one dimension for simplicity. At the current iteration, the latest estimate of the interface position is $(x_I^{n+1})_k$. The



$$(x_I^{n+1})_{k+1} = (x_I^{n+1})_k + e_k$$

$$(v_I^{n+1})_{k+1} = \frac{(x_I^{n+1})_{k+1} - x_I^n}{\Delta t}$$

FIG. 4. Schematic of the incremental velocity updating method.

temperature field is calculated using this as the position of the interface. Introducing a correction e_k, the derivatives in (2b) are evaluated using $(x_I^{n+1})_k + e_k$ as the interface position and the velocity $(V^{n+1})_k$. Satisfaction of (2b), which is done using Newton's method, requires the solution of the nonlinear equation $P(e_k) = 0$, where P is a polynomial in e_k. Notice that the velocity can be made to be implicit by using $[(x_I^{n+1})_k + e_k]/\Delta t$ instead of $(V^{n+1})_k$ in (2b). However, this can lead to a loss of stability and has not proved to be very effective.

Several methods have been implemented in conjunction with a variety of approximations to the normal derivatives in (2b). Here, the discussion is restricted to two of these methods—linear approximation and quadratic finite element approximation—together with the incremental approach to update the interface position. Other methods implemented so far are not as effective or thus behave very similarly to the methods described below and are better discussed in detail elsewhere. Using the notation shown in Fig. 5a in one dimension to simplify the expressions and assuming that the solid is to the left of the interface,

$$\frac{\partial T_S}{\partial n} \cong \frac{T_I - T_2}{a + e} \tag{26a}$$

$$\frac{\partial T_L}{\partial n} \cong \frac{T_3 - T_I}{a - e} \tag{26b}$$

for the linear approximation and

$$\frac{\partial T_S}{\partial n}(x) \cong \left(\frac{2x - 3a - e}{a(2a + e)}\right) T_1 + \left(\frac{2a + e - 2x}{a(a + e)}\right) T_2 + \left(\frac{2x - a}{(a + e)(2a + e)}\right) T_I \tag{27a}$$

$$\frac{\partial T_L}{\partial n}(x) \cong \left(\frac{2x - 3a + 2e}{(e - a)(e - 2a)}\right) T_I + \left(\frac{2x + e - 2a}{a(e - a)}\right) T_3 + \left(\frac{2x - a + e}{a(2a - e)}\right) T_4 \tag{27b}$$

for the quadratic finite element approximation. Equation (27a) is written with respect to a local coordinate system with its origin at $x = p_1$ in Fig. 5a, and (27b) is written with respect to a local system located at $x = p_I$ in the same figure. In order to find the temperatures

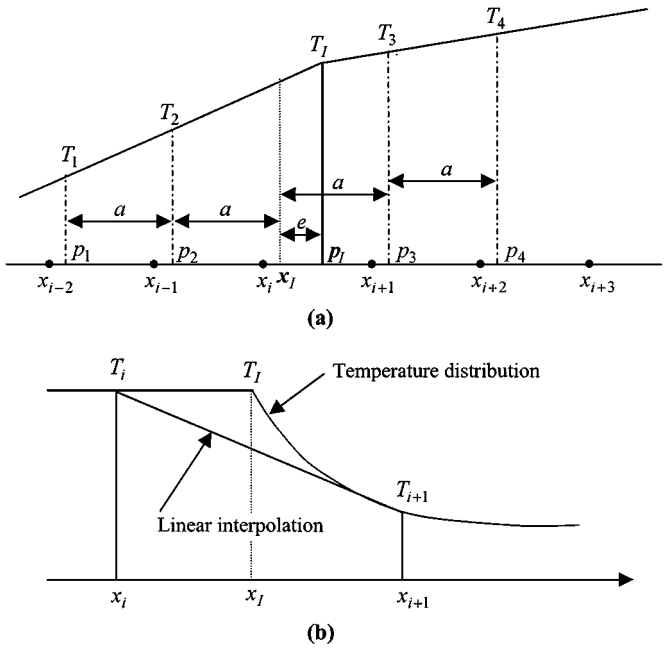


FIG. 5. Schematic and notation for the calculation of the temperature gradients at the interface. (a) Location of the temperature interpolation points. (b) Error introduced by linear interpolation over an element containing the interface.

T_i, $i = 1, 2, 3, 4$, the coordinates (x_i, y_i) of the points in the normal direction to the interface are located and the elements containing these points are identified. The temperatures at the points are obtained from the bilinear interpolant to the temperature field over the elements containing them.

As long as $e \ll a$, (26) yields an $O(a^2)$ approximation to the derivatives at the midpoints $1/2(p_2 + p_I)$ and $1/2(p_I + p_3)$. The quadratic finite element approximations, (27), are $O(a^3)$ when evaluated at the points $p_L = (2a + e)(3 - \sqrt{3})/6$ and $p_R = (2a - e)(3 + \sqrt{3})/6$ to the left and right of p_I, respectively, where the points are given in their corresponding local coordinate systems. The points p_L and p_R correspond to those in a set of two Gauss points over each element that are closest to p_I on each side of the interface [30, p. 196].

In Fig. 5b, we show (one dimensionally for simplicity) a temperature profile typical of the problems considered in this work for an element that contains the interface, together with the linear interpolant to the temperature over the element. It can be observed that the linear interpolant over the element is inaccurate because of the kink in the temperature field at the interface. Therefore, in order to preserve the accuracy in the calculation, we avoid interpolating the temperature over this element by choosing the distance a in Eqs. (26) and (27) large enough to ensure that the sampling points T_2 and T_3 in Fig. 5a do not lie in the element that contains the interface. In practice, $a = h$ has been used with no detrimental effects, even though, strictly speaking, it should be $a = h\sqrt{2}$ in a square mesh. However, smaller values, such as $a = h/2$, produce a loss of accuracy in the calculation. This is important because it shows that the resolution of the method is restricted to the mesh size, as mentioned at the beginning of Section 3, and justifies the assumption that elements can be intersected by one interface segment only.

For the direct method, the expressions for the normal derivatives are obtained from (26) and (27) by setting $e = 0$. Regardless of the method used, a relaxation parameter must be introduced in the calculation of the interface velocity, that is,

$$(V^{n+1})_{k+1} = (V^{n+1})_k + \beta(V^* - (V^{n+1})_k), \tag{28}$$

where V^* is the velocity that has just been calculated using one of the methods described above, and β is the relaxation parameter. The value of β is problem dependent and is chosen to help keep the number of iterations in a time step less than 8. In the direct method, β is kept constant and values between 0.2 and 0.9 have been used, which are larger than the value 0.1 used by Udaykumar *et al.* [10]. In calculations performed so far using the incremental methods, β has been chosen between 0.005 and 0.03. This gives a relaxation factor comparable to that used by Juric and Tryggvason [8]. The convergence criterion used in the calculations presented in this work is that the change in the interface velocity between iterations must be less than 0.001% at all marker points before the iteration stops and one moves to the next time step.

The calculation proceeds as follows:

1. At time $t = t_n$, the temperature field T^n is known at every node in the fixed mesh, and the interface position S^n and the velocity V^n in the direction normal to the interface are known at every marker. Set $i = 0$.
2. For each marker, calculate an initial guess for the position of the interface at time $t^{n+1} = t^n + \Delta t$ from $(x^{n+1})_0 = x_I^n + V^n \Delta t$.
3. Find a new temperature field $(T^{n+1})_i$ using the new interface position.
4. At each marker, calculate e_i from (2b) and (26) or (27).
5. Obtain a new interface position and velocity at each marker from $(x^{n+1})_{i+1} = (x^{n+1})_i + e_i$, $V^* = (V^{n+1})_i + e_i/\Delta t$, and (28).
6. Calculate the maximum relative change in the velocity δV at the markers:

$$\delta V = \max \frac{\|(V^{n+1})_{i+1} - (V^{n+1})_i\|}{\|(V^{n+1})_{i+1}\|}. \tag{29}$$

7. If $\delta V < 10^{-5}$, set $x_I^{n+1} = (x^{n+1})_{i+1}$, $V^{n+1} = (V^{n+1})_{i+1}$ and start the next time step. Otherwise, set $i = i + 1$ and go back to step 3.

4. ACCURACY AND CONVERGENCE

It is well known that finite element approximations to the heat equation using bilinear elements converge at a second-order rate in space, provided that the solution possesses square-integrable first partial derivatives [30, p. 112]. These conditions are generally satisfied in the problems of interest here. However, the approximation to the interface position is not automatically second-order accurate; in fact, the algorithms of Juric and Tryggvason [8] and Udaykumar *et al.* [10] yield only a first-order convergence rate for the interface position. Although a formal error analysis of the accuracy and the convergence rate of the approximation to the interface position obtained with the present algorithm was not possible here, a problem with a known analytical solution is used to show the convergence rate of the present method.

The convergence properties of the present finite element method are discussed in terms of the problem of axially symmetric solidification in the xy-plane resulting from a line sink perpendicular to the plane located at the origin, assuming that, initially, the plane is at a uniform temperature T_∞^*, for which a similarity solution exists [29, p. 118; 33, p. 294]. The solution is given by

$$T_S^*(r,t) = \frac{Q}{4\pi\kappa_S}\left[Ei(\lambda^2) - Ei\left(\frac{r^2}{4\alpha_S t}\right)\right] + T_m^* \tag{30a}$$

$$T_L^*(r,t) = T_\infty^* - \frac{T_\infty^* - T_m^*}{Ei(\Lambda\lambda^2)} Ei\left(\frac{r^2}{4\alpha_L t}\right), \tag{30b}$$

where $\alpha_S = \kappa_S/\rho_S c_{PS}$ and $\alpha_L = \kappa_L/\rho_L c_{PL}$ are the solid and liquid thermal diffusivities, Q is the sink strength, and $Ei(x)$ is the exponential integral $Ei(x) = \int_x^\infty (e^{-s}/s)\,ds,\ x > 0$. The interface location $R(t)$ is given by

$$R(t) = 2\lambda\sqrt{\alpha_S t}, \tag{31}$$

where λ is the unique root of the equation

$$\lambda^2 e^{\lambda^2} + St\Lambda \frac{e^{(1-\Lambda)\lambda^2}}{Ei(\Lambda\lambda^2)} = \frac{Q}{4\pi\rho\alpha_S L}, \tag{32}$$

and the density is assumed to be constant.

Figure 6 illustrates the two-dimensional problem, which is solved for one quarter of the plane over a region of dimensions 1 cm × 1 cm, as shown. The physical properties used in the calculations are $L = 20000$ J/Kg, $\rho_S = \rho_L = \rho = 10^4$ Kg/m^3, $c_{PS} = 100$ J/Kg · K, $c_{PL} = 160$ J/Kg · K, $\kappa_S = 20$ J/s · m · K, $\kappa_L = 10$ J/s · m · K, and $Q = 1000$ J/s · m · K. For this combination of parameters, the value of λ in (31) is 0.10514688 and $St = 0.08$ based on $T_\infty^* - T_m^* = 10$ K. These properties were chosen because they are similar to those of Pb-Sb alloys. The Stefan number was picked small to increase the difficulty of the numerical problem. To avoid the difficulties associated with the singularity at the source and to make a comparison of results for different meshes meaningful, a square of size 0.4 mm × 0.4 mm—corresponding to the largest mesh size used in the calculations—that contains the origin is removed from the domain and the calculations are started using as initial conditions the analytical solution at $t = 10$ s when the interface is at $r = 2.974 \cdot 10^{-3}$ m, as shown in Fig. 6. Along the boundaries defined by $\{x = 0.4\,\text{mm},\ 0 \le y \le 0.4\,\text{mm}\}$, $\{y = 0.4\,\text{mm},\ 0 \le x \le 0.4\,\text{mm}\}$, $x = 10$ mm, and $y = 10$ mm, the analytical solution is imposed at all times as a Dirichlet boundary condition on the temperature field. Along the rest of the segments, $x = 0$ and $y = 0$, the symmetry condition of zero heat flux in, respectively, the y- and x-directions is imposed. Meshes of 25×25, 50×50, 100×100, and 200×200 elements corresponding to a mesh parameter h of 0.4, 0.2, 0.1, and 0.05 mm, respectively, were used, and the time step is set accordingly at $\Delta t = 0.04$, 0.02, 0.01, and 0.005 seconds, respectively. The evolution of the position of the interface obtained on a 50×50 mesh of bilinear elements is shown in Fig. 6 at intervals of 10 seconds. Initially, 25 markers describe the interface; the final interface position at $t = 100$ s contains 97 markers.

In order to assess the error in the calculations, the simulations were run from $t_0 = 10$ s to $t_F = 85$ s. The radial distance to the origin R_i of the markers was calculated at every time

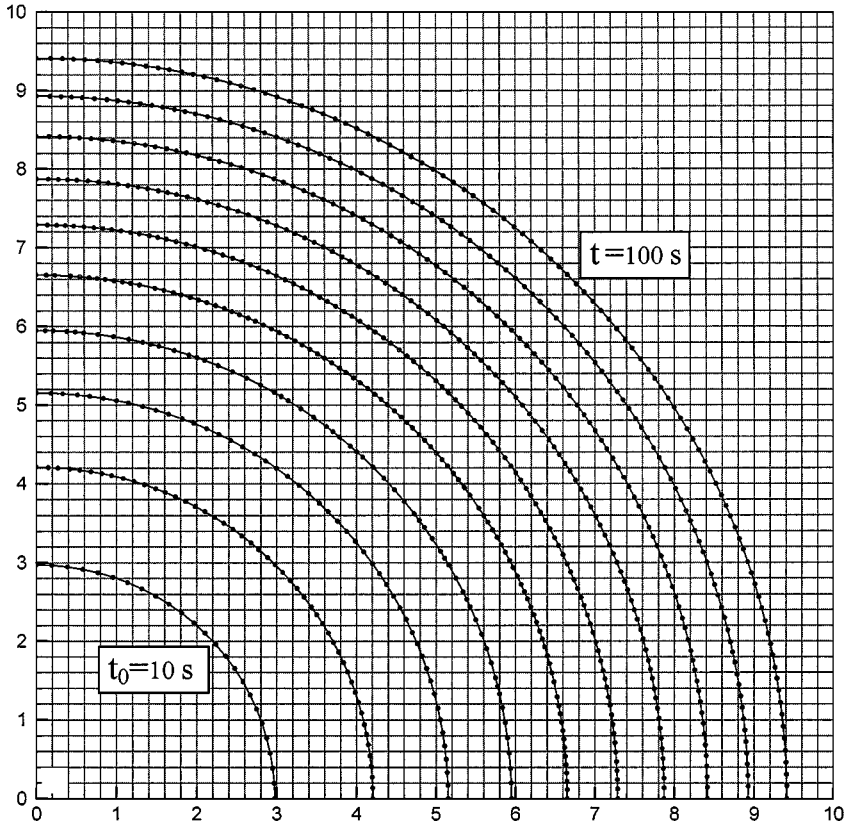


FIG. 6. Axially symmetric solidification (interface position calculated in a 50 by 50 element mesh).

step, and the relative error in their position was obtained using both the Euclidean norm, given by

$$E_2 = \frac{1}{R}\sqrt{\frac{1}{n}\sum_{i=1}^{n}(R_i - R)^2}, \tag{33}$$

and the maximum norm,

$$E_M = \max|R_i - R| \quad (i = 1, n), \tag{34}$$

where R is the exact interface position and n is the total number of markers. This gives a measure of the average and maximum errors as a function of time. The errors at each time step are then added and divided by the total number of time steps to obtain one number representative of the average error in the interface position throughout the whole calculation. This was done for each of the methods using each of the four meshes to obtain convergence rates. The results are presented in Figs. 7 and 8, where part (a) shows the average error at the markers as a function of time and part (b) shows the total average error and the convergence rate in a log–log scale. Both the quadratic and the linear schemes to calculate the interface position yield second-order accuracy, however, the quadratic

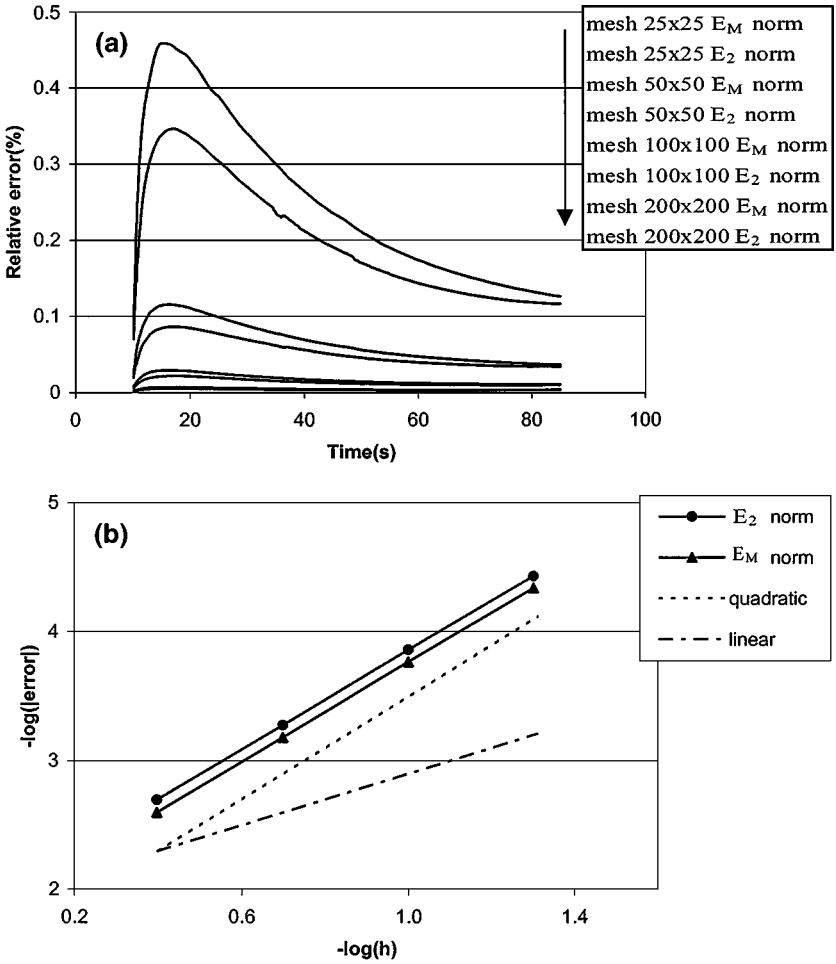


FIG. 7. Linear approximation to the temperature gradient. (a) Average error in the interface position. (b) Convergence rate.

approximation is more accurate, generally generating errors about 20% smaller than the linear approximation. The error in the approximation of the interface velocity is shown in Figs. 9 and 10 for the linear and quadratic cases, respectively. It shows an $O(h)$ convergence rate, as is expected because the interface velocity is determined by the temperature gradients at the interface, and the gradients are approximated to $O(h)$ by the finite element solution. It is interesting to note that the quadratic scheme does not produce a more accurate approximation to the interface velocity, i.e., the errors for both cases are practically the same. The convergence rate is expected to be first order, since it involves the evaluation of the temperature gradients at the interface; this is reflected in Figs. 9b and 10b. In the first part of the figures, it is observed that there are significant oscillations in the velocity error that, in one-dimensional calculations, have been identified as oscillations in the interface velocity itself. Ways to eliminate or at least minimize these oscillations are currently being explored.

The behavior of the error at individual markers has been examined at three marker points located at 0, 22.5, and 45 degrees with respect to the positive x-axis. In general, these follow

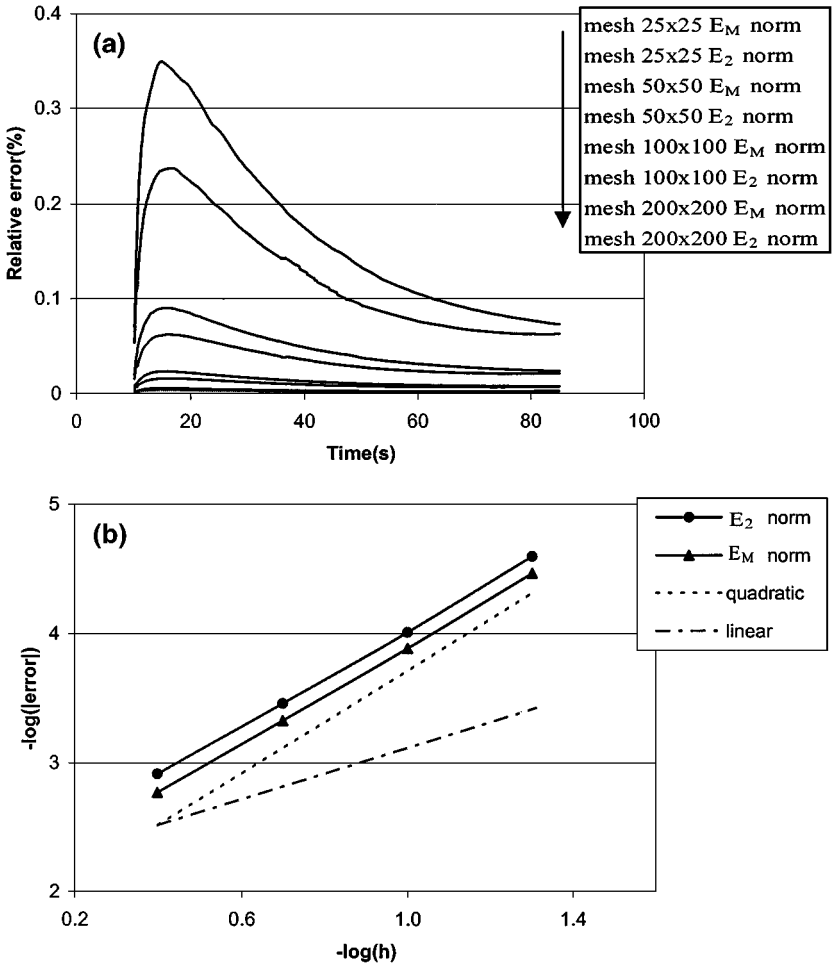


FIG. 8. Quadratic interpolation to the temperature gradient. (a) Average error in the interface position. (b) Convergence rate.

the same behavior as the average errors. The maximum errors occurred in the marker at 22.5 degrees and are shown in Fig. 11 for the linear case. In part (a), oscillations in the interface position error are observed that decrease rapidly in magnitude as the mesh is refined, but the oscillations persist even in the finest mesh. In part (b), only the portion where the interface goes between 4.5 and 5.5 mm has been depicted for better clarity. Oscillations in the error in the interface velocity are observed that also decrease in amplitude with mesh refinement, but not as rapidly, and remain with a magnitude of about 0.25% for the 200×200 mesh. It is clear to us that these oscillations are related to perturbations generated when the interface markers go across inter-element boundaries, to perturbations that arise when new markers are added on the interface, and to the latent heat source generated at the interface across which the conductivity changes. Further approaches to alleviate this problem are currently under study. The data related to this error analysis are summarized in Table I. We should also point out that this is but one of many one- and two-dimensional tests that have been performed to assess the accuracy of the algorithm; all the results consistently show that the algorithm is second-order accurate.

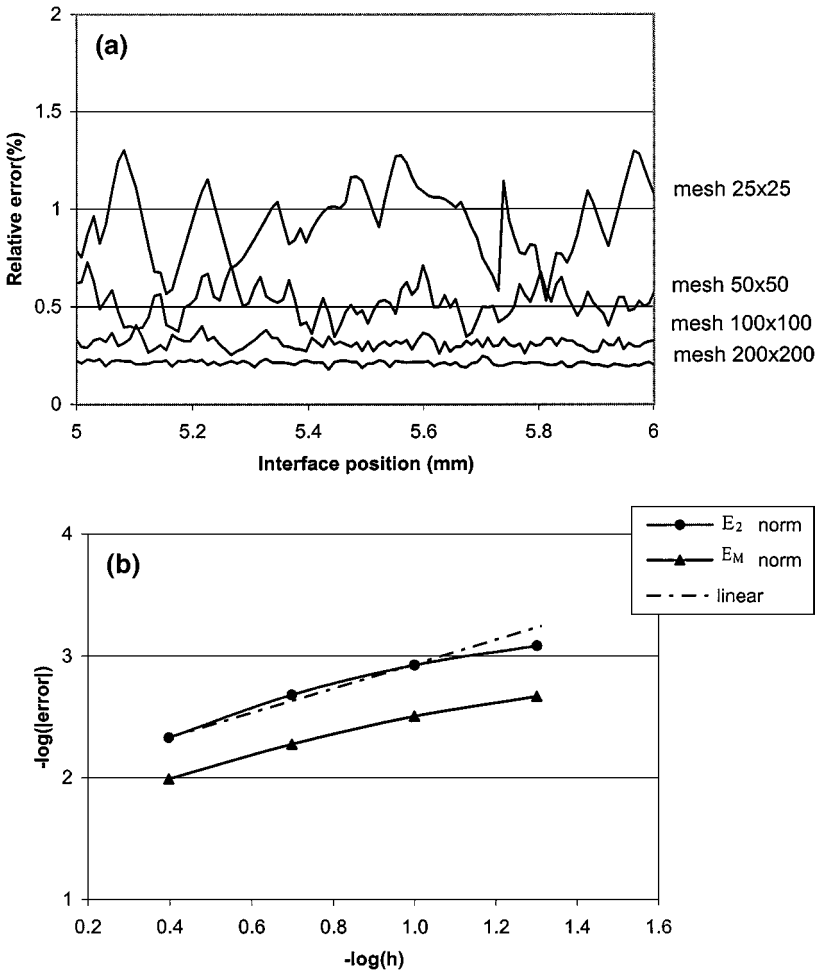


FIG. 9. Linear approximation to the temperature gradient. (a) Error in the interface velocity. (b) Convergence rate for the interface velocity.

5. EXAMPLES OF UNSTABLE SOLIDIFICATION

The problems used to evaluate the accuracy and convergence properties of the algorithm in the previous section were cases of stable solidification. Concentration here will be on simulations of solidification into an undercooled liquid. First, some problems solved by both Juric and Tryggvason [8] and Udaykumar *et al.* [10] are examined and then results from the present study are compared with those reported in their work. These calculations are performed in a square region of side 4 units long. The initial seed is specified by $x = x_c + R\cos(\theta)$ and $y = y_c + R\sin(\theta)$, where (x_c, y_c) is the center of the region and $R = 0.1 + 0.02\cos(4\theta)$—a case with four-fold symmetry. The initial temperature of the seed is T_m, and the initial temperature of the liquid is set to $T_\infty < T_m$. The undercooling, $T_\infty - T_m$, is chosen so that $St = -0.5$. The surface tension and kinetic mobility are isotropic, $\sigma = \mu = 0.002$. The heat capacity and thermal conductivity are the same in both phases, i.e., $\eta = \Lambda = 1$. The boundaries of the domain are adiabatic. As in [8, 10], solutions were obtained with meshes of 100×100, 200×200, and 400×400 bilinear elements in

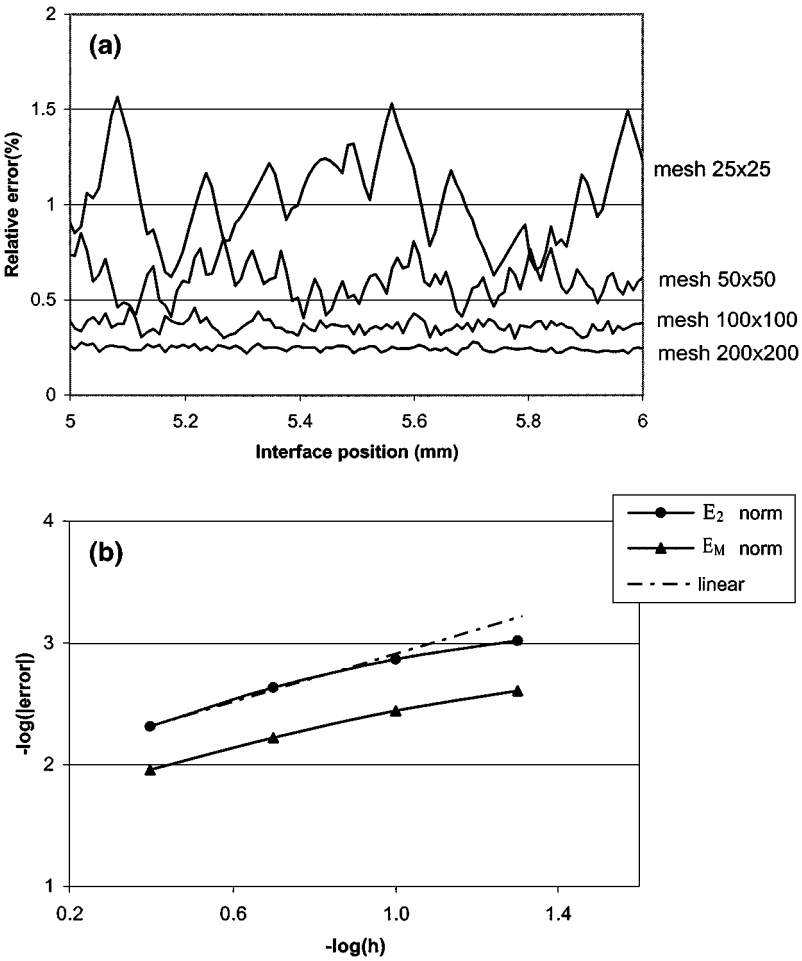


FIG. 10. Quadratic interpolation to the temperature gradient. (a) Error in the interface velocity in the maximum norm. (b) Convergence rate for the interface velocity.

order to illustrate the influence of mesh refinement in the solution. The results are shown in Fig. 12, where no difference is observed between the results obtained in the 200×200 and the 400×400 meshes, substantiating the high rate of convergence of this model. The initial seed and the shapes obtained at times 0.2, 0.4, 0.6, 0.8, and 1.0 are pictured. In the coarse mesh, the results differ substantially from the results reported in Fig. 7 of [8]. The differences are considerably reduced in the finer meshes, but the results obtained in the 400×400 mesh exhibit secondary branches that are less pronounced and have flatter boundaries than those in [8]. On the other hand, these shapes show excellent agreement with those shown in Fig. 21 of [10]—no differences can be observed for the finest mesh.

Simulations with unequal solid and liquid conductivity, where $0.01 \leq \Lambda \leq 5$, that can be compared to the results in Fig. 11 of [8] and Fig. 22 of [10] are now discussed. These were obtained with a 400×400 mesh. These results differ from those of [8], which were obtained in a 300×300 grid, but agree very well with the results shown in [10]. In particular, the cases with $\Lambda = 5.0$, 1.0, and 0.5 shown in Fig. 22 of [10] were calculated, and the results

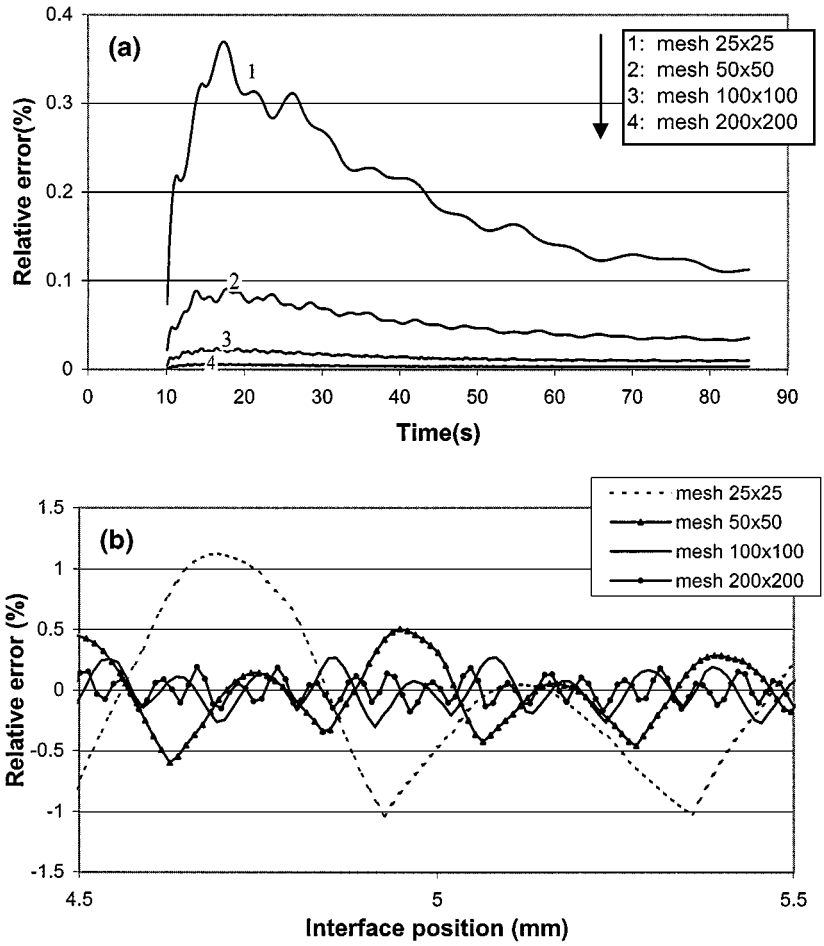


FIG. 11. Error in the interface position and velocity for marker at 22.5 degrees.

were the same. Figure 13 shows the interface shapes and temperature field for $\Lambda = 0.01$, which are to be compared with Fig. 22d in [10]. These solutions are somewhat different, but not significantly different. However, a large discrepancy is observed in the velocity of solidification. The results reported in [10] show plots from time 0.0 to 0.25, when the upper fingers reach a distance of about $z = 3.3$, while the shapes shown in Fig. 13 reach the same distance at $t = 0.45$.

A grid anisotropy test performed using the data in the first example is shown in Fig. 14. In order to avoid boundary effects, the simulations were stopped when the temperature at any node on the boundary changed by more than 0.1%. Results when the principal axes of the seed are at angles of 0, 19, and 45 degrees to the x-axis are shown in parts (a), (b), and (c), respectively. Figure 14d shows the three calculations rotated to 0 degrees and superimposed. As can be observed, the solutions are almost identical. However, if the simulations are continued for a longer time, the effect of the boundary causes the solutions to diverge from each other. Similar independence from the grid orientation has been observed in other situations as well.

Simulation results are presented next for a case with anisotropic surface tension that corresponds to $St = -0.8$, $\sigma(\theta) = 0.002[1 + 0.4(8/3\sin^4(2\theta) - 1)]$, $\eta = \Lambda = 1$, and $\mu = 0$,

TABLE I
Error Data for Two-Dimensional Solidification Calculations

Mesh			All markers				Marker 1		Marker 2		Marker 3	
Elements	h (mm)	Δt (s)	E_2 (R)	E_m (R)	E_2 (V)	E_m (V)	Error (R)	Error (V)	Error (R)	Error (V)	Error (R)	Error (V)
a. Linear approximation												
25 × 25	0.4	0.04	0.20097	0.25152	0.46497	1.02089	0.18832	0.37750	0.19600	0.43319	0.25140	0.31334
50 × 50	0.2	0.02	0.05315	0.06617	0.20821	0.52731	0.04883	0.14673	0.05238	0.19259	0.06603	0.07314
100 × 100	0.1	0.01	0.01381	0.01716	0.11867	0.31227	0.01265	0.06285	0.01368	0.11185	0.01705	0.02537
200 × 200	0.05	0.005	0.00373	0.00463	0.08295	0.21512	0.00353	0.03283	0.00370	0.07683	0.00454	0.00755
b. Quadratic approximation												
25 × 25	0.4	0.04	0.12182	0.17005	0.48639	1.10142	0.10509	0.39915	0.11413	0.46970	0.16990	0.25414
50 × 50	0.2	0.02	0.03459	0.04704	0.23275	0.60349	0.02943	0.16125	0.03331	0.21800	0.04704	0.06098
100 × 100	0.1	0.01	0.00974	0.01303	0.13565	0.36044	0.00837	0.06894	0.00951	0.12938	0.01292	0.01707
200 × 200	0.05	0.005	0.00253	0.00342	0.09522	0.24753	0.00230	0.03683	0.00248	0.08958	0.00333	0.00767
$\rho = 10000$ kg/m^3, $L = 20000$ J/kg, $\mathrm{St} \equiv \frac{c_{pL}(T_\infty - T_m)}{L} = 0.08$			$c_{ps} = 100$ J/(kg × K), $T_m = 0$,		$c_{PL} = 160$ J/(kg × K), $T_\infty = 10$		$k_S = 20$ J/(s × m × K), Q_S (heat sink) = 1000 J/(m × s),			$k_L = 10$ J/(s × m × K), $\lambda = 0.10514688$		

[a] All errors are relative. Tolerance = 10^{-6} for the time step iteration, 10^{-8} for the equations solver.

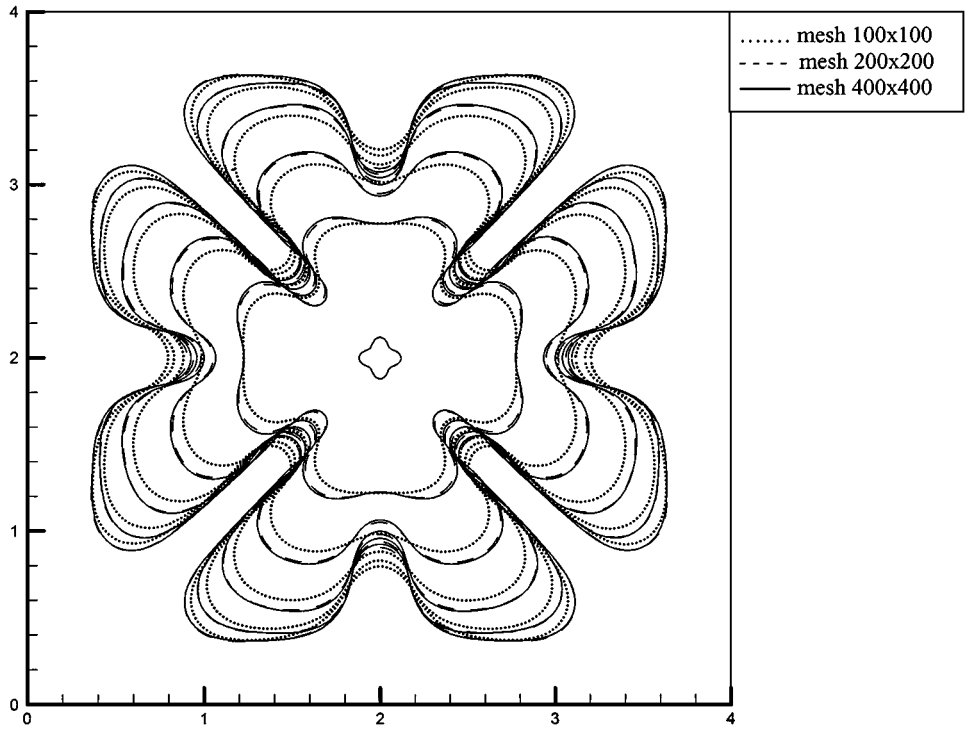


FIG. 12. Effect of mesh refinement in the interface position for unstable solidification with isotropic surface tension and kinetic mobility.

which is the case with four-fold anisotropy in the surface tension presented in Fig. 25a of [10]. The shapes obtained in a 400×400 mesh are shown in Fig. 15. They are very similar to those given in [10], but are symmetric about both coordinate axes, as opposed to the shapes in [10], which are not symmetric about the y-axis. Most of the rest of the simulations reported in [8, 10] have been repeated with the present algorithm with similar results. There are interface shapes that are different from those in [8], but the present results compare well with those in [10] except for occasional differences. The most common difference is the persistence of symmetry in the present calculations in cases where symmetry is not present in the figures of [10].

Simulations were made in an attempt to reproduce the results of Sullivan and Lynch [34], who considered solidification of succinonitrile in a perturbed planar two-dimensional front. A linear stability analysis predicts the most unstable wavelength to be $\lambda_0 = 27.4\,\mu$m. With this in mind, calculations were performed in a region $6\lambda_0$ wide by $4\lambda_0$ high using a mesh of 300×200 bilinear elements and the physical data given in Table 2 of [34]: $c_{PS} = 4.7 \times 10^{-4}$ cal/mm^3 K, $c_{PL} = 4.59 \times 10^{-4}$ cal/mm^3 K, $\kappa_S = 5.36 \times 10^{-5}$ cal/mm $\cdot$ s $\cdot$ K, $\kappa_L = 5.32 \times 10^{-5}$ cal/mm $\cdot$ s $\cdot$ K, $T_m^* = 331.25$ K, $L = 0.01086$ cal/mm^3, $T_\infty^* = 321.786\,K$, $\sigma = 6.53 \times 10^{-5}$ mm $\cdot$ K, and $\mu = 5.0 \times 10^{-3}$ K/mm $\cdot$ s. This results in a Stefan number $St = -0.4$. The problem also differs from the previous ones in that the vertical boundaries are kept at fixed constant temperatures: T_m on the left-hand side and T_∞ on the right. The top and bottom boundaries are adiabatic. Sufficient information could not be obtained to reproduce the exact initial temperature used in [34], therefore, constant solid and liquid initial temperatures were used. The initial plane solid boundary is located at $x = 0.1\lambda_0$ and

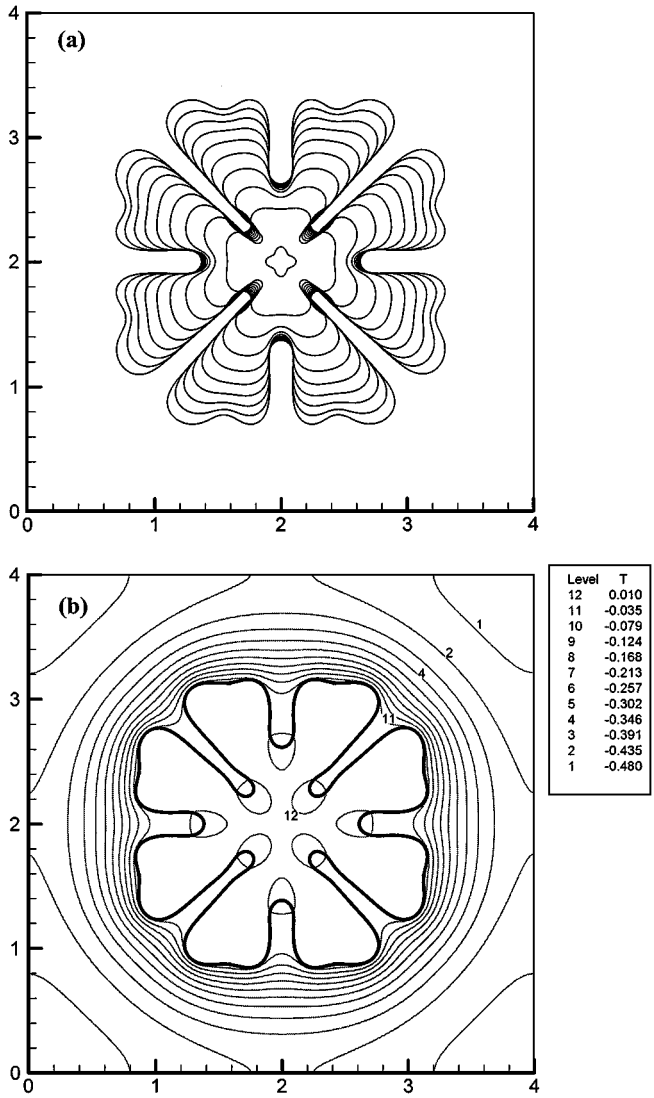


FIG. 13. Solution for a solid-to-liquid conductivity ratio $\Lambda = 0.01$. (a) Interface position at times $t = 0.05i$, $i = 1$–9. (b) Temperature contours at $t = 0.35$.

is perturbed with a sinusoidal wave of amplitude $\lambda_0/500$ and wavelength λ_0 as given in [34]. Figure 16 shows the results of the simulation. The temperature contours show hot spots (labeled 12) where grooves form and one of the branches tends to slow down. Remelting can be observed at the tips of the smaller arms that are left behind and in the interdendritic regions at the far left where the temperature rises above melting. A direct comparison with those of [34] is not possible because of the difference in the initial conditions, which are critical in these problems. Large changes in the shapes that develop are caused by very small changes in the initial conditions. However, qualitatively, the same shapes and general dynamics of the interface at the beginning of the development of the dendrites as shown in Figs. 13, 15, and 17 of [34] were observed, except that the instability develops much faster. In the present calculations, solidification is faster at the beginning and near the end,

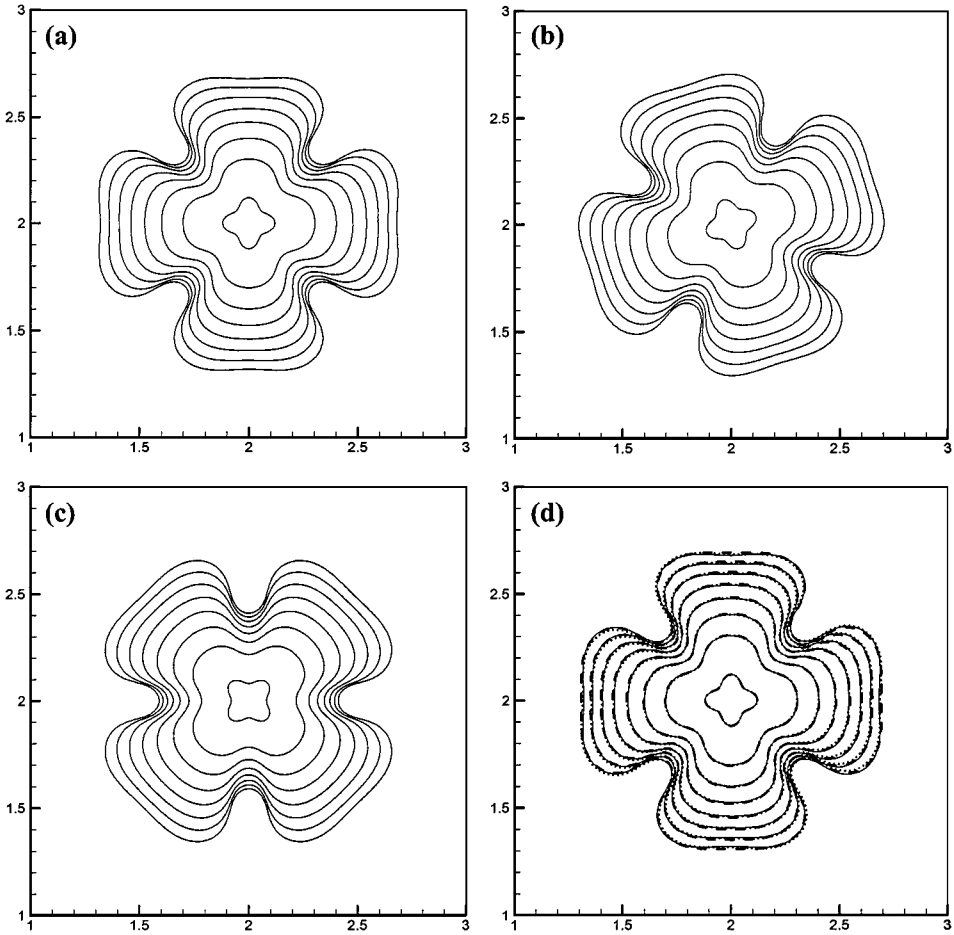


FIG. 14. Effect of the seed orientation with respect to the mesh. (a) The seed's axes have the same orientation as the x- and y-axes. (b) Seed's axes oriented at a 19-degree angle to the x- and y-axes. (c) Seed's axes at 45 degrees to the x- and y-axes. (d) Three cases (a–c) rotated to the same orientation and superposed.

when the temperature gradients are largest. For the case of solidification from a perturbed plane front, numerical experiments were performed in which different levels of noise were introduced during the calculation. Different solutions were obtained for different levels of noise. In some cases, where no secondary branches would develop without continuously perturbing the system, secondary and tertiary branches developed when the noise was added. It is therefore believed that some of the branching that has been reported in the literature may have been numerically driven. Since a criterion to determine how these perturbations should be introduced is lacking, these results, which probably could not be reproduced by other researchers, are not reported here.

In order to assess the accuracy of the present model in predicting the tip growth rate of a dendrite, numerical experiments were performed and compared with the two-dimensional boundary integral solution obtained by Meiron [35] using the microscopic solvability hypothesis. Calculations were performed for a single dendrite arm growing in a square domain in the first quandrant from a circular seed centered at the origin, as illustrated in Figure 17c, in a way similar to calculations performed by Karma and Rappel [25]. This choice provides

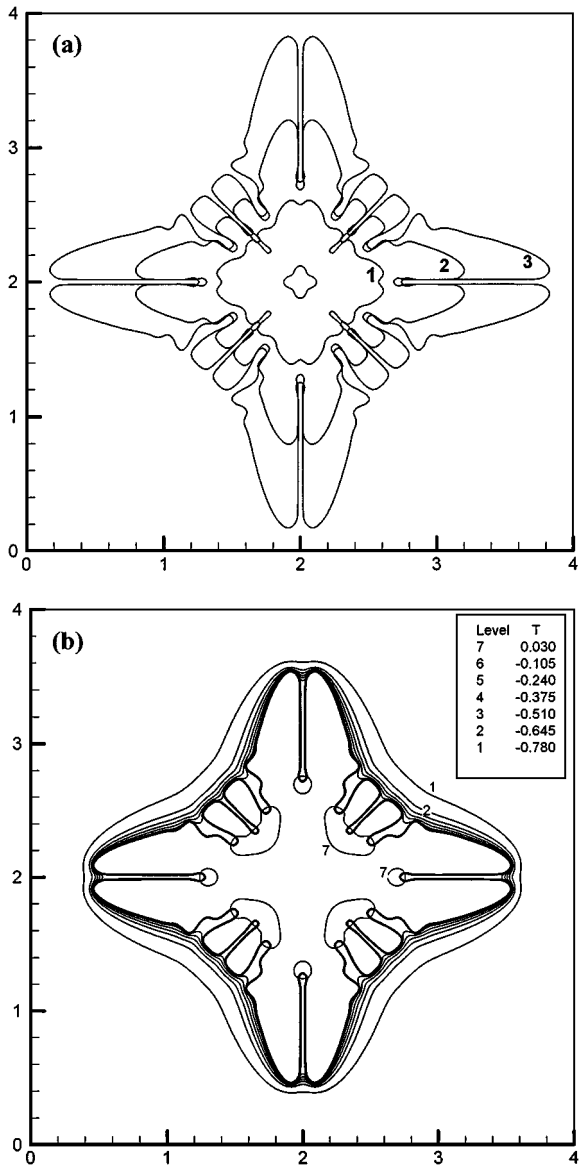


FIG. 15. Unstable solidification with anisotropic surface tension and adiabatic boundary conditions. (a) Interface shape at times $t = 0.015$, 0.03, and 0.045. (b) Temperature field and interface at $t = 0.038$; the thicker line is the interface.

the maximum growth length before the effect of temperature diffusion reaches the boundary and affects the velocity of the dendrite tip. Adiabatic boundary conditions were imposed on all four sides.

In order to compare results, the same nondimensional interfacial conditions, surface tension, anisotropy function, and parameter definitions as in Meiron [35] are used, i.e.,

$$T_I = -\sigma f(\theta)\kappa \tag{35}$$

$$f(\theta) = 1 + \zeta[1 - \cos 4(\theta - \theta_0)], \tag{36}$$

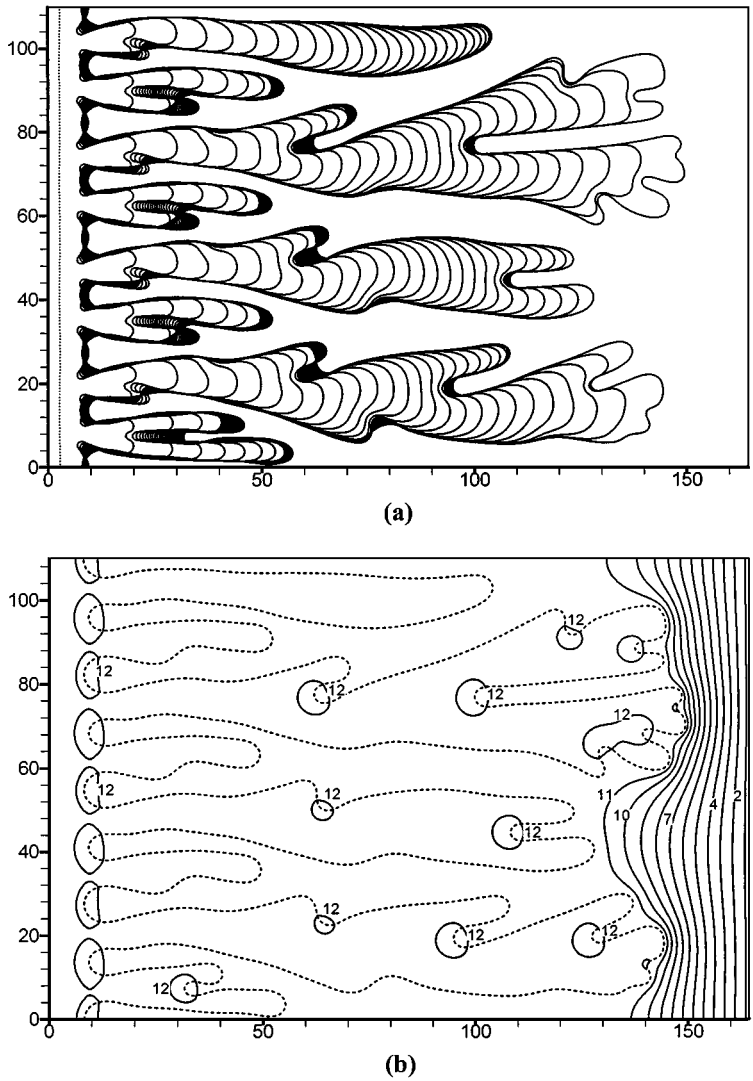


FIG. 16. Unstable solidification from a perturbed plane front showing the initial perturbation to the interface (dotted line) to the far left. The interface is shown from $t = 0$ to $t = 0.052$ s at time intervals $\Delta t = 0.002$ s and the temperature field at $t = 0.052$ s.

where $\sigma = \sigma^*/LH$ and σ^*/L is the capilarity length scale. The parameter ζ represents the strength of the anisotropy and θ is the angle between the normal to the interface, $\hat{n}$, and the x-axis and θ_0 defines the direction of maximum growth. We define the nondimensional velocity by $V = HV^*/\alpha_L$, which is related to the nondimensional velocity V^M in Meiron [35] by the expression $V^M = \sigma V/2$.

Results of the calculations for five different undercoolings, $St = -0.25, -0.35, -0.45, -0.55$, and -0.65, were obtained using the values $\zeta = 0.4$ and $\theta_0 = \pi/4$. Figures 17a and 17b show the dendrite tip velocity as a function of time for $St = -0.25$ and -0.65, respectively. Figure 17c illustrates the development of the dendrite at $St = -0.45$, and Fig. 17d shows the steady-state velocity calculated using the present method compared with those obtained from Fig. 8 of [35]. These results are listed in Table II, along with the

TABLE II

Comparison of Steady-State Tip Velocities Calculated Using the Present Method with Those Calculated in Ref. [35] Using the Microscopic Solvability Hypothesis

$(-)St$	σ	Domain Size	Mesh	V	V^M	CPU Time (hrs)
0.25	2×10^{-4}	18×18	600×600	0.40	4.0×10^{-5}	56
0.35	4×10^{-4}	12×12	600×600	1.05	2.1×10^{-4}	53
0.45	10^{-3}	6×6	500×500	1.76	8.8×10^{-4}	25
0.55	10^{-3}	4×4	400×400	5.10	2.55×10^{-3}	12
0.65	2×10^{-3}	4×4	400×400	7.43	7.43×10^{-3}	12

Note. Simulations were performed using one processor on a 400-MHz SGI 2000 origin machine.

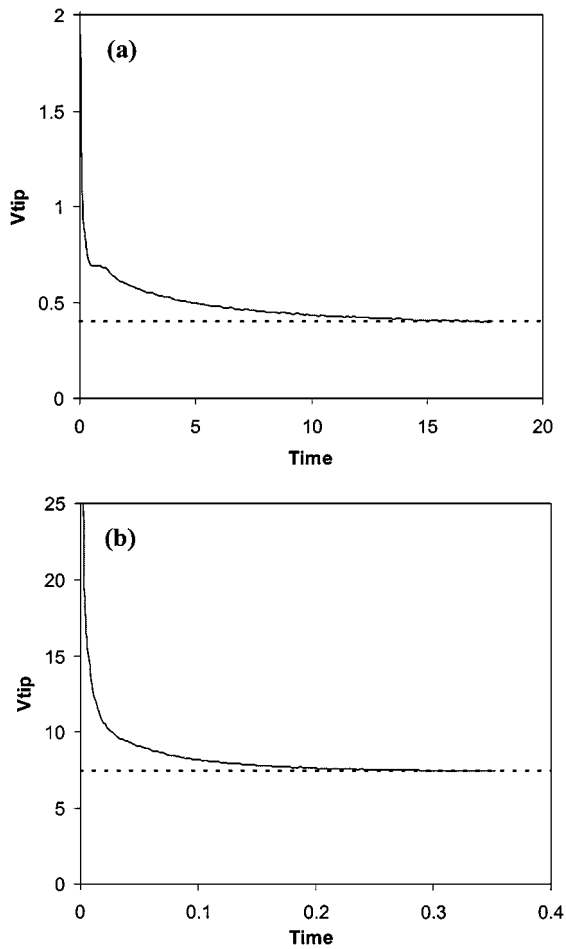


FIG. 17. Comparison of the predicted dendrite tip growth rate with the steady-state results of [35]. (a) Tip velocity as a function of time for $St = -0.25$. (b) Tip velocity as a function of time for $St = -0.65$. (c) Domain and interface contours for $St = -0.45$ for $0 \le t \le 2.2$ at intervals of $\Delta t = 0.1$. (d) Dimensionless tip velocity versus undercooling: -- calculated;—reference [35].

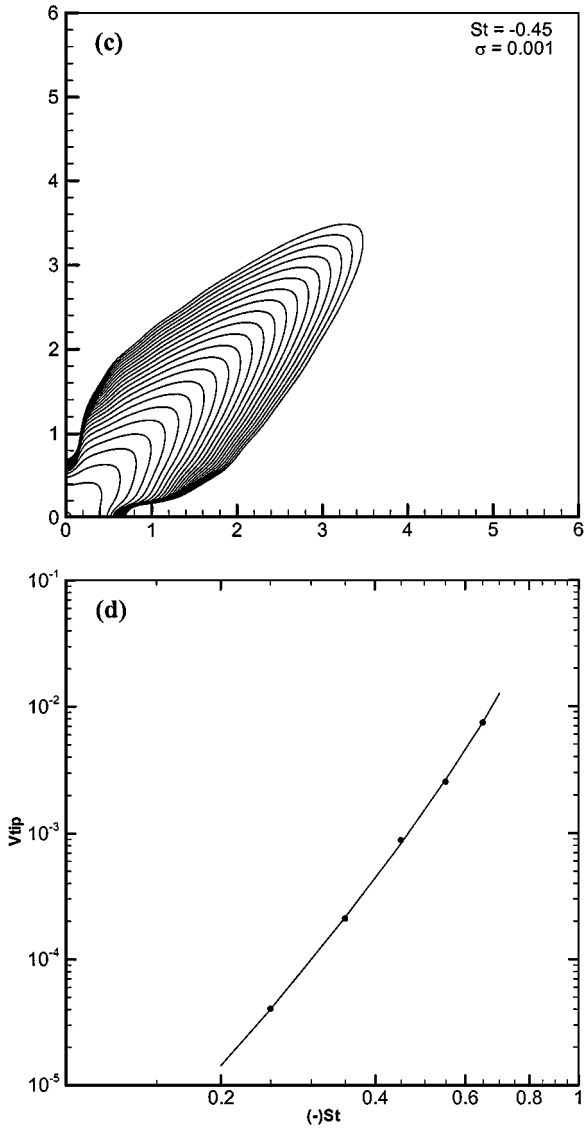


FIG. 17—*Continued*

parameters used in the calculation and the approximate CPU times expended. The number of time steps performed in each case varied between 10,000 and 25,000; calculation was carried until the velocity shows less than 1% variation over 1000 time steps. The estimated steady-state velocity was obtained by averaging the calculated instantaneous velocity over the last 100 consecutive time steps. The CPU column is included in order to convey an idea of the efficiency of the present model. Accurate predictions of the CPU time required in a simulation are difficult to make because it varies according to the problem. It must also be noted that no special efforts were made at this point to optimize the code. The calculations shown were performed on a 400-MHz SGI Origin 2000 machine. This performance cannot be compared with those of the models in [9] and [10] because no computational costs were reported there.

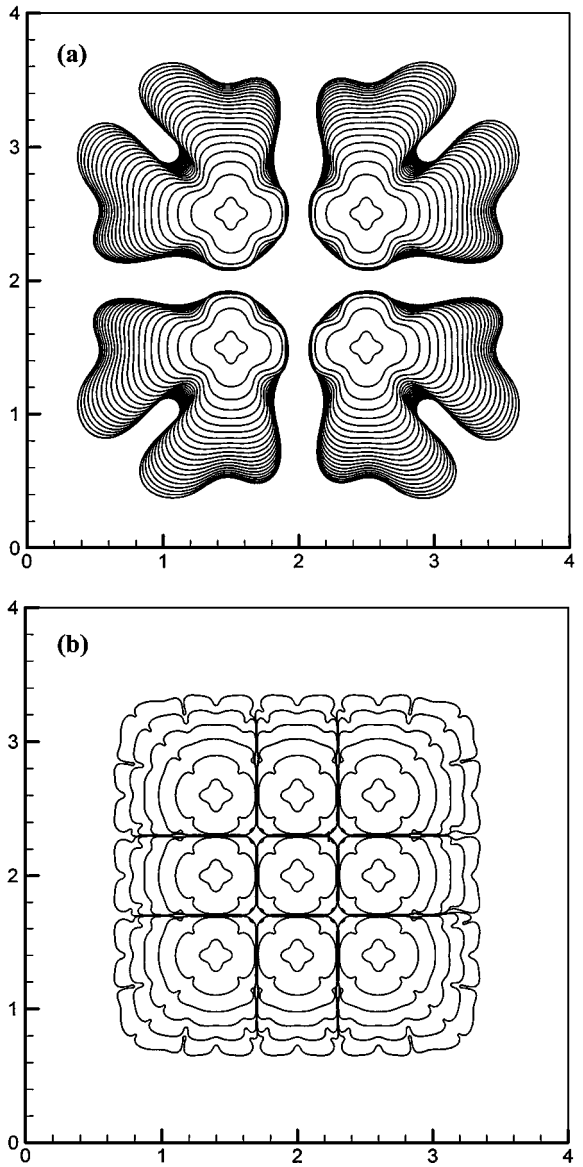


FIG. 18. Interactions in the solidification of several dendrites with adiabatic boundary conditions. (a) Four seeds in a 200 × 200 mesh at $St = -0.5$; time range from 0 to 0.4 and 20 interfaces for each seed shown at time interval $\Delta t = 0.02$. (b) Nine seeds in a 400 × 400 mesh at $St = -1.0$; time range from 0 to 0.015 plotted at intervals $\Delta t = 0.003$.

Examples that illustrate the interaction between dendrites growing simultaneously are now presented. Conditions are the same as in the first example shown in Fig. 12; in particular, all the boundaries are adiabatic. The results of two calculations are shown in Fig. 18. The one in part (a) includes four identical seeds, the mesh is 200 × 200 elements, and the Stefan number is −0.5. Because the magnitude of the Stefan number is less than one, more energy is released by latent heat than is required to raise the temperature of the melt; therefore, as the dendrites get closer together, the solidification process will come to a halt. A case with nine squarely distributed seeds, a 400 × 400-element mesh, and $St = -1$ is shown in part (b). The dendrites can be seen growing together and forming grain boundaries.

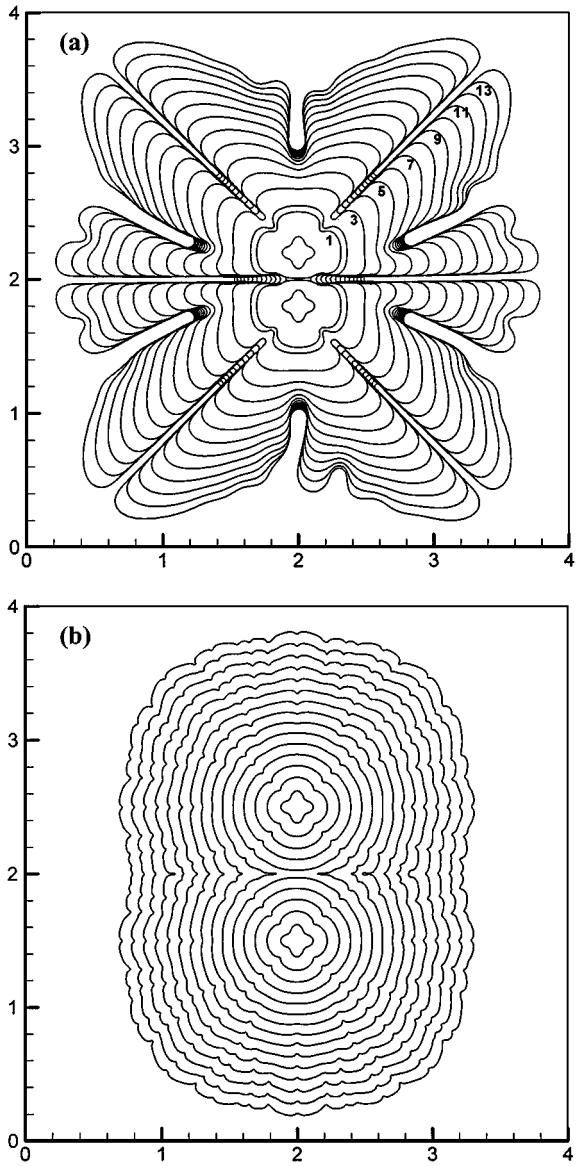


FIG. 19. Merging two initial interfaces into one. (a) Fixed temperature boundary conditions at $St = -0.8$. Interfaces between $t = 0$ and $t = 0.13$ at intervals $\Delta t = 0.01$ except for numbers one and two, which are very close to when merging takes place. (b) Adiabatic boundary conditions at $St = -1.2$; results between $t = 0$ and $t = 0.014$ at intervals $\Delta t = 0.001$.

The final cases also use the same data as the first example. Results depicted in Fig. 19a show a simulation calculated in a 200×200 mesh in which the boundaries were kept at a fixed temperature to allow heat loss from the computational region at $St = -0.8$. The centers of the seeds were initially 0.4 unit apart, and the interfaces were allowed to merge when they came in contact. The dendrites were merged when the distance between any two markers not in the same interface became smaller than $0.4\,h$, where h denotes the mesh size. Figure 19b shows another case calculated in a 400×400 mesh using adiabatic boundary conditions at a Stefan number $St = -1.2$. In this case, the interface is very stable

and becomes smoother as time goes by. Because the magnitude of the Stefan number is greater than one, the heat released is not great enough to stop the solidification and the dendrites can get together even though the boundary conditions are adiabatic. The centers of the seeds are initially one unit apart.

6. CONCLUSIONS

A finite element model for interface tracking based on a fixed mesh that is simple and easy to use has been developed. The method shows second-order convergence in the approximation to the interface and first-order when approximating the interface velocity. This is consistent with the fact that the interface position depends directly on the temperature field and the velocity on the temperature gradient at the interface, and has been achieved by ensuring that all approximations are consistent with second-order accuracy. The approximation to the interface velocity shows oscillations that could not be eliminated. However, it is believed that it is important to do so in order to be able to fully control the introduction of noise in the calculations, since this may significantly alter the solutions in unstable problems. A comparison of results obtained with this method with some of the results in [8, 10, 34] shows that the different methods do not always yield the same solution to unstable solidification problems and underscores the need to establish well-designed benchmark problems to evaluate the existing numerical models. It is believed that at least part of the cause for these discrepancies lies in the different levels of numerical noise present in the algorithms. This shows up in the present method in the approximation to the interface velocity, even though a great deal of effort went into attempting to eliminate these numerical perturbations during the development of the algorithm.

A comparison between the direct sharp-front method and some of the results that have been obtained with the phase-field method has been left for future study.

A special effort was made to include all the information needed for other researchers to model each of the cases that were simulated using the present algorithm. Basic numerical techniques that address this type of simulation have already been established. However, comparisons between the methods are not readily available and continuing progress will be greatly enhanced if careful comparisons and benchmarking is effected. This is probably the best way to improve our understanding of the different techniques needed in order to introduce refinements that will bring them to the next level of reliability.

ACKNOWLEDGMENTS

This work was supported in part by the National Aeronautics and Space Administration under grant NCC8-96 and by the National Science Foundation under grant DMR-0072955. We are also indebted to Prof. David R. Poirier for his constant input and assistance during the development of the model.

REFERENCES

1. D. G. McCartney and J. D. Hunt, A numerical finite difference model of steady-state cellular and dendritic growth, *Metall. Trans. A* **15A**, 983 (1984).
2. J. M. Sullivan, Jr., D. R. Lynch, and K. O'Neill, Finite element simulations of planar instabilities during solidification of an undercooled melt, *J. Comput. Phys.* **69**, 81 (1987).

3. K. H. Tacke, Application of finite difference enthalpy methods to dendritic growth and ripening, in *Free Boundary Problems: Theory and Applications*, edited by K. H. Hoffmann and J. Sprekels (Longman Sci. Tech., Essex, UK, 1990), Vol. 2, pp. 636–643.
4. J. A. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98**, 231 (1992).
5. R. Almgren, Variational algorithms and pattern formation in dendritic solidification, *J. Comput. Phys.* **106**, 337 (1993).
6. A. A. Wheeler, B. T. Murray, and R. J. Schaefer, Computation of dendrites using a phase field model, *Physica D* **66**, 243 (1993).
7. J. A. Warren and W. J. Boettinger, Prediction of dendritic growth and macrosegregation patterns in a binary alloy using the phase-field method, *Acta Metall. Mater.* **43**, 689 (1995).
8. D. Juric and G. Tryggvason, A front-tracking method for dendritic solidification, *J. Comput. Phys.* **123**, 127 (1996).
9. N. Provatas, N. Goldenfeld, and J. Dantzig, Adaptive mesh refinement computation of solidification microstructures using dynamic data structures, *J. Comput. Phys.* **148**, 265 (1999).
10. H. S. Udaykumar, R. Mittal, and W. Shyy, Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids, *J. Comput. Phys.* **153**, 535 (1999).
11. H. G. Landau, Heat conduction in a melting solid, *Q. Appl. Math.* **8**, 81 (1950).
12. J. Crank, *Free and Moving Boundary Problems* (Oxford Univ. Press, London, 1984).
13. L. A. Crivelli and S. R. Idelsohn, A temperature based finite element solution of phase change problems, *Int. J. Numer. Meth. Eng.* **23**, 99 (1986).
14. J. M. Floryan and H. Ramussen, Numerical methods for viscous flows with moving boundaries, *Appl. Mech. Rev.* **42**, 323 (1989).
15. V. R. Voller, C. M. Swaminathan, and B. G. Thomas, Fixed grid techniques for phase change problems: a review, *Int. J. Numer. Meth. Eng.* **30**, 875 (1990).
16. H. S. Udaykumar and W. Shyy, Simulation of interfacial instabilities during solidification—I. Conduction and capillarity effects, *Int. J. Heat Mass Transfer* **38**, 2057 (1995).
17. J. Crank and R. D. Phahle, Melting ice by the isotherm migration method, *Bull J. Inst. Math. Appl.* **9**, 12 (1973).
18. L. E. Goodrich, Efficient numerical techniques for one-dimensional thermal problems with phase change, *Int. J. Heat Mass Transfer* **21**, 615 (1978).
19. V. R. Voller and M. Cross, Accurate solution of moving boundary problems using the enthalpy method, *Int. J. Heat Mass Transfer* **24**, 545 (1981).
20. A. N. Alexandrou, An inverse finite element method for directly formulated free boundary problems, *Int. J. Numer. Meth. Eng.* **28**, 2283 (1989).
21. J. B. Collins and H. Levine, Diffuse interface model of diffusion-limited crystal growth, *Phys. Rev. B* **31**, 6119 (1985).
22. G. Caginalp, Surface tension and supercooling in solidification theory, in *Applications of Field Theory to Statistical Mechanics*, edited by L. Garrido (Springer-verlag, Berlin, 1985), pp. 216–226.
23. G. Caginalp, An analysis of a phase field method of a free boundary, *Arch. Rat. Mech. Anal.* **92**, 205 (1986).
24. J. S. Langer, Models of pattern formation in first-order phase transitions, in *Directions in Condensed Matter Physics*, edited by J. Grinstein and G. Mazenko (World Scientific, Philadelphia, 1986), pp. 165–186.
25. A. Karma and W. J. Rappel, Quantitative phase-field modeling of dendritic growth in two and three dimensions, *Phys. Rev. E* **57**, 4323 (1998).
26. K. Brattkus and D. I. Meiron, Numerical simulation of unsteady crystal growth, *SIAM J. Appl. Math.* **52**, 1303 (1992).
27. N. Palle and J. A. Dantzig, An adaptive mesh refinement scheme for solidification problems, *Metall. Mat. Trans. A* **27A**, 707 (1996).
28. D. R. Lynch and J. M. Sullivan, Heat conservation in deforming element phase change simulations, *J. Comput. Phys.* **57**, 303 (1985).
29. V. Alexiades and A. D. Solomon, *Mathematical Modeling of Melting and Freezing Processes* (Hemisphere, Washington, DC, 1993).

30. J. C. Heinrich and D. W. Pepper, *Intermediate Finite Element Method: Fluid Flow and Heat Transfer Applications* (Taylor and Francis, Philadelphia, 1999), chapter 7.
31. G. Strang and G. Fix, *An Analysis of the Finite Element Method* (Prentice-Hall, Englewood Cliffs, NJ, 1973).
32. A. J. Chorin, Curvature and solidification, *J. Comput. Phys.* **58**, 472 (1985).
33. H. Carslow and J. Jaeger, *Conduction of Heat in Solids, 2nd* ed. (Clarendon, Oxford, 1959).
34. J. M. Sullivan, Jr, and D. R. Lynch, Non-linear simulation of dendritic solidification of an undercooled melt, *Int. J. Numer. Meth. Eng.* **25**, 415 (1988).
35. D. I. Meiron, Selection of steady states in the two-dimensional symmetric model of dendritic growth, *Phys. Rev. A* **33**, 2704 (1986).