Journal of Computational Physics **153,** 535–574 (1999)
Article ID jcph.1999.6294, available online at http://www.idealibrary.com on IDEAL®

Computation of Solid–Liquid Phase Fronts in the Sharp Interface Limit on Fixed Grids

H. S. Udaykumar,* R. Mittal,† and Wei Shyy*

**Department of Aerospace Engineering, Mechanics and Engineering Science and †Department of Mechanical Engineering, University of Florida, Gainesville, Florida 32611*

Received September 3, 1998; revised April 26, 1999

A finite-difference formulation is applied to track solid–liquid boundaries on a fixed underlying grid. The interface is not of finite thickness but is treated as a discontinuity and is explicitly tracked. The imposition of boundary conditions exactly on a *sharp interface* that passes through the Cartesian grid is performed using simple stencil readjustments in the vicinity of the interface. Attention is paid to formulating difference schemes that are globally second-order accurate in $\mathbf{x}$ and t. Error analysis and grid refinement studies are performed for test problems involving the diffusion and convection–diffusion equations, and for stable solidification problems. Issues concerned with stability and change of phase of grid points in the evolution of solid–liquid phase fronts are also addressed. It is demonstrated that the field calculation is second-order accurate while the position of the phase front is calculated to first-order accuracy. Furthermore, the accuracy estimates hold for the cases where there is a property jump across the interface. Unstable solidification phenomena are simulated and an attempt is made to compare results with previously published work. The results indicate the need to begin an effort to benchmark computations of instability phenomena.

1. INTRODUCTION

1.1. Motivation

Consider several objects embedded in a domain Ω as shown in Fig. 1. Each of the objects is separated from the surrounding fluid, designated phase 0, by a boundary Γ_{i0}. We shall use the term "immersed boundary" for this type of internal boundary in the flow field. Such a boundary may arise in several types of flow problems. For example, this may be a boundary that represents a solid geometry through which fluid flows. The solid boundary, as in many fluid–structure interaction problems, may execute motions under the influence of the flow around it. In other situations this boundary may represent a phase discontinuity at which

0021-9991/99 $30.00

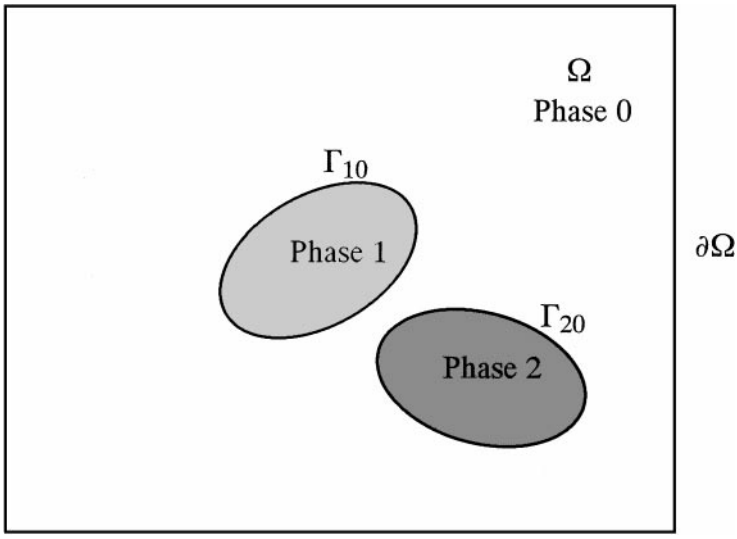

FIG. 1. Illustration of immersed boundaries in a flow field. The immersed boundaries can be stationary or moving. They can enclose solid, liquid, or gaseous phases. Furthermore, the boundary can move or deform by interaction with the flow around it.

a phase transition occurs, such as in the solidification of material from the melt, or in the vaporization of fluid. The boundary may also separate two fluids as in the case of bubbles or drops immersed in an ambient fluid. For the purposes of the discussion presented here we consider the solution of a transport equation for the variable ϕ in the domain Ω in which the boundaries Γ_{ij} are embedded,

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = \nabla \cdot k \nabla \phi + f, \tag{1}$$

where $\mathbf{u}$ is the convective speed which may be a function of ϕ and f is a body–force term. On the phase boundaries different types of boundary conditions may apply and may be written in the general form

$$F\left(\phi, \frac{\partial \phi}{\partial n}\right) = \alpha_{\Gamma_{ij}}. \tag{2}$$

In some cases the boundary may be a discontinuity in the variable ϕ, in which case jump conditions may be prescribed on Γ_{ij} such as

$$[\phi]_{\Gamma_{ij}} = \beta \qquad \text{or} \qquad \left[k \frac{\partial \phi}{\partial n}\right]_{\Gamma_{ij}} = \gamma. \tag{3}$$

Whether such boundaries in the flow domain are stationary or moving, computing solutions for the flow equations requires applying boundary conditions of the Dirichlet, Neumann, or mixed type on the surface. If the boundary moves, one would like to track the boundary in time. If the properties are discontinuous across the boundary, the discontinuity would be required to be maintained. While the deformation of the boundary may be induced by the transport processes around it, the boundary in turn can transmit its influence to the surrounding in the form of stresses or energy supplied at the infinitesimally thin surface. An

example of the former is the force applied by a stretched membrane due to internal forces developed within it and an example in the latter category is the release of latent heat or heat of combustion at the boundary. Such interactions of the boundary and the flow field will need to be calculated accurately.

In this paper we compute the motion of solid–liquid phase fronts by explicitly tracking the interface over a fixed Cartesian grid. Ideally one would like to simulate the effect of the boundary Γ_{ij} by treating it explicitly, without smearing the information at the interface, i.e., with minimum numerical diffusion. There are many ways of doing this; for a fixed boundary, when the shape is truly complex, one can resort to block-structured domain decomposition (Shyy, 1994), overset meshes (Steger, 1991; Johnson and Belk, 1995), or unstructured boundary-conforming curvilinear grids (Venkatakrishnan, 1996) to discretize the domain. For moving boundaries which may undergo large deformations, or are subject to topology changes in the course of their evolution, fixed grid techniques are advantageous. Generating boundary-fitted grids to conform to such complex boundaries becomes difficult. To circumvent this difficulty a variety of methods have been developed and applied to track fronts on fixed meshes. The popular methods for tracking moving boundaries in an Eulerian framework are the volume-of-fluid method and its refinements (Hirt and Nichols, 1981; Brackbill *et al*., 1992; Kothe and Mjolsness, 1992; Scardovelli and Zaleski, 1999), the level-set method (Osher and Sethian, 1988), and the phase-field method (Caginalp, 1984; Langer, 1986; Kobayashi, 1993; Wheeler *et al*., 1992). These methods perform very well in problems involving free surfaces. For the particular problems of interest here, namely, tracking of solid–liquid phase fronts, the level-set method (Chen *et al*., 1997; Sethian and Strain, 1992; Zhang *et al*., 1998), the phase-field method (Kobayashi, 1993; Wheeler *et al*., 1992), and enthalpy type methods (Voller and Prakash, 1987) have been employed. In these purely Eulerian methods, the interface is not tracked explicitly but is deduced based on a field variable such as the distance function, order parameter, or local enthalpy. The interface is of finite thickness and may occupy a few grid points in a direction normal to it. Although these methods converge to the sharp interface models as the grid size decreases, numerical difficulties restrict operation of these methods to interface thicknesses proportional to the grid size.

In mixed Eulerian–Lagrangian methods, the interface is tracked explicitly, while the computations are performed on fixed grids. Examples of this approach are the immersed boundary technique (Peskin, 1977; Unverdi and Tryggvason, 1992; Juric and Tryggvason, 1996; Udaykumar *et al*., 1997), cut-cell type approaches (Shyy *et al*., 1996; Udaykumar and Shyy, 1995b; Udaykumar *et al*., 1996; Pember *et al*., 1995; Quirk, 1992), the immersed interface method (LeVeque and Li, 1994), and the fictitious domain methods (Glowinski *et al*., 1994). In essence, these methods differ from the purely Eulerian method in that the boundaries are tracked explicitly as a set of curves.

Among mixed methods there are widely different ways of handling the interaction of the interface with the flow field. The most widely used mixed technique is the immersed boundary technique which was originated by Peskin (1977) and used extensively by Tryggvason and co-workers (Unverdi and Tryggvason, 1992; Juric and Tryggvason, 1996) and by the present group (Udaykumar *et al*., 1997; Kan *et al*., 1998). While explicitly tracking the interface, the method transmits the information regarding the discontinuity across the interface to the grid in much the same way as purely Eulerian methods. As demonstrated by Beyer and LeVeque (1992) this results in a method that is globally $O(h)$ accurate, where h is the grid spacing. On the other hand, the cut-cell treatment (Udaykumar

and Shyy, 1995b; Udaykumar *et al*., 1996) proceeds to reconstruct the domain on either side of the interface with considerable detail in regard to the piecing together of the fractional cells that arise due to the passage of the boundary through the grid. Thus, smearing of the interface is totally avoided in this method, with a conservative control volume treatment that demands great care in assembling flux information in cells adjoining the interface. However, such methods can become tedious, particularly when 3D computations are considered.

As far as solidification simulations are concerned the outstanding work with Eulerian–Lagrangian methods has been that of Juric and Tryggvason (1996). They showed that interface tracking can be used effectively in 2D to solve the problem of complex evolving solidification morphologies. Also, in their work, the physical parameters are directly related to the physically identifiable quantities, such as surface tension and anisotropy, unlike in the phase-field methods (Wheeler *et al*., 1992), where the computational parameters are only indirectly related to the actual physical quantities. Although phase-field methods have been used to obtain impressive results of the dynamics of phase fronts, the presence of a free parameter in such methods, namely the interface thickness parameter, renders the method less suited to direct comparison/benchmarking. Therefore, the work of Juric and Tryggvason (1996) (hereinafter abbreviated as JT96) is important as a starting point for benchmarking calculations of unstable solidification front calculations. However, even though the interface is explicitly tracked in JT96, the discontinuities at the interface are still spread over a few grid cells. In solid–liquid phase front evolution these discontinuities can arise from property jumps, sources of latent heat and solute, or capillary terms that act at the infinitesimally thin interface. Truly sharp interface methods have been few. Previous work in this regard has been performed by Udaykumar and Shyy (1995b) using the cut-cell method, by Almgren (1993) using a variational formulation, and by Saito *et al*. (1988) using the boundary integral approach. Recently LeVeque and co-workers (LeVeque and Li, 1994) have introduced the immersed interface method for tracking interfaces explicitly, while maintaining sharp discontinuities, and this method has shown promise for problems involving elliptic PDEs. The method has also been combined with the level-set approach to alleviate problems involved in tracking interfaces explicitly and has been applied to track interfaces in Hele–Shaw flow (Hou *et al*., 1997). In the context of finite-element methods, Schmidt (1996) has applied a variational formulation to compute dendritic growth in 2D as well as 3D. In this paper, we seek to present a simple finite-difference method for tracking sharp interfaces on a fixed Cartesian mesh. In contrast to a previous finite-volume approach presented in Udaykumar *et al*. (1996), the present finite-difference formulation is easier to implement. Traditional finite differencing is involved in the bulk of the domain. In cells containing the interface, modification of the stencil based on explicit knowledge of the interface is performed and the resulting discrete equations are solved over the whole domain. The simplicity of the treatment facilitates error estimation for the finite differencing. The computations performed support the expected error estimates. We make a concerted effort to benchmark our results by directly comparing with the work of JT96. In contrast to JT96, where the immersed boundary method was used and the interface spread over a few grid cells, here property jumps and interface boundary conditions as well as source terms at the interface are treated as discontinuities and included as such in the discretization procedure. In contrast to the immersed interface method (LeVeque and Li, 1994), the discretization procedure is simple and performed in the Cartesian directions and not in a rotated frame of reference. Therefore, it is elementary to extend from a

simple finite-difference scheme implemented on a Cartesian grid without immersed boundaries. Extension to higher dimensions and to Navier–Stokes equations should follow along the lines laid out in this paper. Work in this direction is in progress and will be reported elsewhere.

1.2. Remarks on Explicit Interface Tracking

When interfaces are tracked explicitly, periodic reorganization of the interface information becomes necessary. This can result from dilation or compression of the interface or can be due to topological changes of the boundary. In 2D, mergers and breakups of boundaries can be handled quite effectively, as demonstrated in Udaykumar and Shyy (1995a) and Juric and Tryggvasson (1996). In 3D the operations to be performed can be more complicated (Jayaraman *et al*., 1997; Snyder and Woodbury, 1993). Therefore, in 3D situations, explicit tracking of interfaces will be work-intensive in the context of mergers and breakups, in comparison to purely Eulerian methods. In the presence of surface tension, the growth of parasitic modes on the surface can result when an explicit interface update is carried out using reasonable time step sizes (Tu and Peskin, 1992). Thus, an implicit scheme for interface motion needs to be devised. If the flow solver is also implicit, as in Udaykumar *et al*. (1997), this presents no constraints since the interface and flow field evolution can be fully coupled through the iterations and taken simultaneously to desired levels of convergence. A related problem, that of stiffness due to surface tension, has been addressed by Hou *et al*. (1994) and an alternative formulation for the interface motion has been derived in terms of the variables θ (the angle made by the tangent to the interface with the horizontal) and L (the interfacial perimeter). It has been shown that this formulation alleviates the stiffness from interface tracking schemes in the presence of surface tension. The violation of the entropy condition, i.e., the inability of an explicit curve-evolution scheme to detect and circumvent crossover of the curve or the formation of cusps and fishtails, was pointed out by Osher and Sethian (1988). This situation can be encountered in problems such as curvature-driven growth, where the tendency to form cusps on the interface exists. When cusps are expected or encountered, special measures can be taken to surgically remove such points from the interfacial string (Glimm *et al*., 1988; Chorin, 1990).

While one has to be cognizant of the above issues in adopting an explicit tracking strategy, there are situations in which explicit interface information in fact becomes desirable. One instance is when a solid–liquid boundary is being tracked and where the no-slip condition is to be applied. This can be done for fixed grid flow solvers by applying the no-slip condition at the exact location of the interface as in the cut-cell or immersed interface approaches. Next, in the dynamics of membranes such as in problems of cell dynamics (Dong *et al*., 1988; Kan *et al*., 1998) and adhesion (Jones *et al*., 1995) in biofluids or the stretch of pliable aerodynamic surfaces (Smith and Shyy, 1996; Fauci and Peskin, 1988), the forces generated within the membranes depend on the stretching and bending of the membranes. This requires information on the tangential dilatation of the interface. Further, for the case in which a boundary is anchored to a surface, such as the adhesion of the membrane of a cell to a substrate, the forces transmitted to the membrane need to be calculated (Dembo *et al*., 1988), and explicit tracking is ideally suited to providing this information. The ability of mixed Eulerian–Lagrangian methods to incorporate both solid–liquid no-slip boundaries as well as fluid–fluid interfaces has been demonstrated in our previous work (Shyy *et al*., 1996; Udaykumar *et al*., 1997).

2. THE NUMERICAL METHOD

2.1. Main Features

There are pros and cons to each of the various methods for solving moving boundary problems and the choice of a method is dictated by the physics and desired numerical accuracy. The main advantage of explicitly tracking solid–liquid boundaries is that no smearing of the boundary is involved. This renders explicit tracking methods suitable for solving flows around solid objects embedded on fixed Cartesian meshes (Goirier and Powell, 1996; Quirk, 1992; Young *et al*., 1992; Melton *et al*., 1993; Pember *et al*., 1995; Goldstein *et al*., 1993, 1995; Glowinski *et al*., 1994; Shyy *et al*., 1996; Udaykumar *et al*., 1996, 1997).

In this work we choose to track the solid–liquid front explicitly. Our effort is directed toward developing an accurate yet simple way of solving equations of the type in Eq. (1) on a uniform Cartesian mesh in the presence of embedded boundaries. We demonstrate that the present method achieves the following objectives:

1. The interface is tracked as a discontinuity and boundary conditions of the Dirichlet/Neumann type are applied on the tracked fronts.
2. The inclusion of embedded boundaries into the discretization scheme involves simple measures in the vicinity of the interface. Such points are few compared to the overall grid size.
3. Based on truncation error analysis, a discretization scheme can be developed so that global second-order accuracy in the field variable can be maintained. We show that this holds true in going from one to two dimensions and in the presence of moving boundaries.
4. For the solidification problem the interface velocity is computed directly from the Stefan condition and the normal gradients of the temperature are evaluated to second-order accuracy. The curvature-dependent boundary conditions are imposed at the exact interface location. This is in contrast to the approach of JT96, who adopted a Newton iteration technique to obtain the value of interface velocity so that the interface boundary condition was satisfied. Since we treat the interface as a sharp discontinuity, such measures are not required in our case. This facilitates unambiguous characterization of discretization errors.
5. The stiffness of the interface evolution in curvature-driven growth is alleviated by using an implicit formulation to couple the interface evolution with temperature field evolution. Therefore the time step restriction in our case is a convective criterion.
6. The issue of change of phase of a grid point when the boundary crosses over it is dealt with by a simple analogy with purely Lagrangian methods. This involves redefining the stencils in the points adjoining the interface to account for the grid points that have changed phase. The approach taken is shown to have no negative impact on the accuracy of the computations.

We now describe the main components of the technique.

2.2. A Finite-Difference Algorithm for Solid–Liquid Moving Boundaries

In the following we will follow the basic ideas of the immersed interface method and the cut-cell approach by applying one-sided differencing to obtain the discretization of the governing equations at control points that lie next to the boundary. Our objective is to solve a PDE of the form given in Eq. (1) in such a way that an $O(h^2)$ accurate spatial and $O(\delta t^2)$ accurate temporal discretization can be maintained. To illustrate the methods and

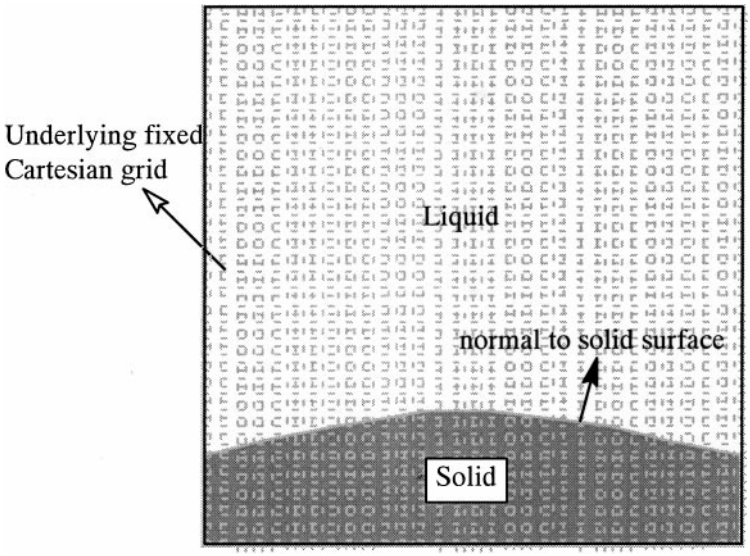

FIG. 2. Configuration for the solidification problem. The normal points from the solid to the liquid phase.

issues involved we choose as an example the solution of Eq. (1) for a typical moving boundary problem, such as solidification/melting of pure material from the melt. The problem, as illustrated in Fig. 2, may be formulated as follows:

Let the transported variable ϕ be the temperature in each phase. Then on the boundary of the solid the temperature is specified and is given by the Gibbs–Thomson condition (Kessler *et al.*, 1988),

$$\phi_\Gamma = \phi_m \left(1 - \frac{\sigma\kappa}{L_f}\right), \tag{4}$$

where σ is the surface tension, L_f is the latent heat of fusion, κ is the curvature, and ϕ_m is the melting temperature. At macroscopic scales, the effects of surface tension are negligible and the above boundary condition reduces to the statement that at the interface the temperature is equal to the melting temperature. The interface moves in accordance with the interfacial heat balance condition (Stefan condition):

$$V_n = \frac{1}{\rho_s L_f}\left[k_s\left(\frac{\partial\phi}{\partial n}\right)_s - k_l\left(\frac{\partial\phi}{\partial n}\right)_l\right]. \tag{5}$$

We wish to solve Eq. (1) in the regions occupied by the solid and liquid, as shown in Fig. 1 above, to desired levels of accuracy. We also wish to impose the boundary condition, Eq. (4), on the temperature fields in the two regions. Then the boundary between the phases is advanced using Eq. (5).

We will now describe the various components of the solution algorithm which meets the accuracy goals but also maintains a simple discretization scheme that can easily be extended to 3D. These can be listed as follows: interface tracking; communication between interface and flow solver; discretization of field equations and application of boundary conditions; and interface update.

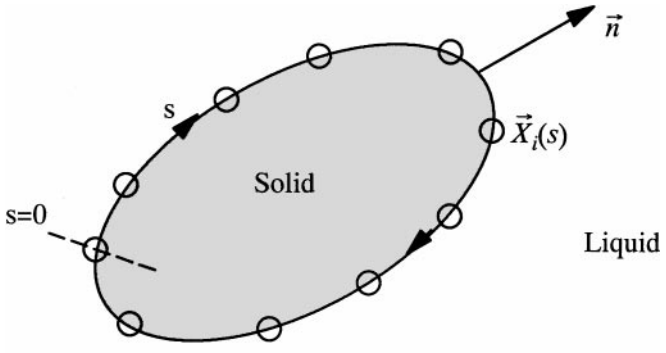

FIG. 3. Convention for definition of interface normal. The arclength parametrization allows identification of the phase on each side of the interface. The solid lies to the right as the interface is traversed in the direction of s, the arclength. Open circles denote interfacial markers.

We will now detail our approach in each of the above components and provide the relevant accuracy estimates.

I. *Tracking the Interface*

The interface is described by interfacial markers defined by the coordinates $\mathbf{X}(s)$. The spacing between the markers is maintained at some fraction of the grid spacing, $0.5h < ds < 1.5h$. The convention adopted is that as one traverses the interface along the arclength, the solid lies to the right. This is illustrated in Fig. 3. The functions $x(s) = a_x s^2 + b_x s + c_x$ and $y(s) = a_y s^2 + b_y s + c_y$ are generated. The coefficients $a_{x/y}$, $b_{x/y}$, and $c_{x/y}$ at any interfacial point i are obtained by fitting polynomials through the coordinates (x_{i-1}, y_{i-1}), (x_i, y_i), and (x_{i+1}, y_{i+1}). The coefficients $a_{x/y}$, $b_{x/y}$, and $c_{x/y}$ are stored for each marker point. The normal to the interface then points from the solid to the liquid and is given by

$$n_x = \frac{-y_s}{\left(x_s^2 + y_s^2\right)^{1/2}}, \qquad n_y = \frac{x_s}{\left(x_s^2 + y_s^2\right)^{1/2}}. \tag{6}$$

The curvature is then obtained, for the 2D planar case, from

$$\kappa = \nabla \cdot \hat{\boldsymbol{n}} = \frac{y_s x_{ss} - y_{ss} x_s}{\left(y_s^2 + x_s^2\right)^{3/2}}. \tag{7}$$

The derivatives x_s, x_{ss}, y_s, and y_{ss} are evaluated using central differencing along the arclength coordinate s. Cubic splines were also tried without observable differences in the results for previous test problems. Therefore, central differencing was adopted since it is easily applicable to different end conditions for the boundaries.

II. *Relationship between Interface and Grid*

Once the interface has been defined, the information on its relationship with the grid has to be established. There may be several interfaces (henceforth called objects) immersed in the domain. Each of the objects may enclose material with different transport properties. Therefore, with respect to phase 0, which is the surrounding fluid, there may be a discontinuity in transport properties, such as viscosity and conductivity across the

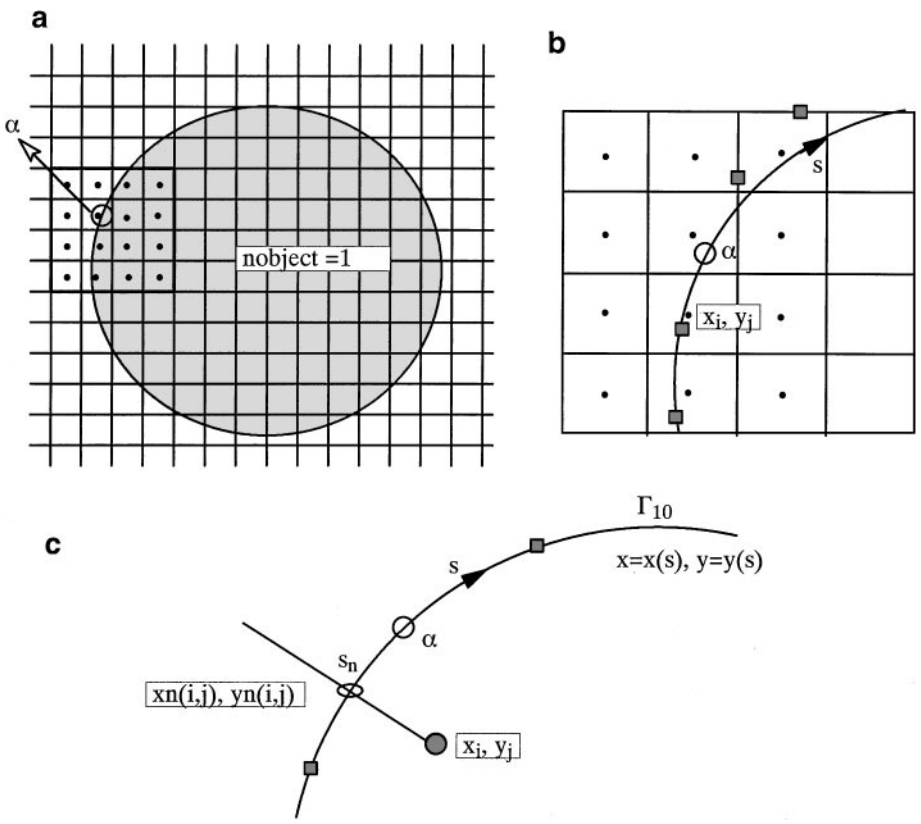

FIG. 4. Procedures leading to phase and border cell identification. (a) The box represents points in the vicinity of an interfacial marker α. (b) Interfacial marker placement in relation to the box identified. (c) Illustration of normal drawn from the grid point (i, j) to the interface. The open ellipse represents the intersection point of the normal with the interface which is defined by the curves $x(s), y(s)$.

boundary separating phase 1, which is enclosed within object 1 and phase 0. One needs to identify the phase inside object 1 as phase 1. This is accomplished in the following way.

On a Cartesian grid, it is a trivial matter to identify the cell in which a given interfacial point lies. Figure 4a shows a 4×4 block of cells surrounding an interfacial point α which is denoted by an open circle. One obtains, for each of the grid points (x_i, y_j) shown in Fig. 4b, lying in the block, the normal from the point to the interface. The location $xn(i, j), yn(i, j)$ where this normal intersects the interface is also obtained. This is easily done by employing the parametrization of the curve and the resulting polynomials:

$$x(s) = a_x s^2 + b_x s + c_x, \qquad y(s) = a_y s^2 + b_y s + c_y. \tag{8}$$

Now let the normal be described by the line $y = ax + b$. Since the normal passes through x_i, y_j, and xn, yn it is a simple matter to deduce that

$$a = \frac{yn - y_i}{xn - x_i} = -\frac{1}{(dy/dx)_n} = -\frac{2a_x s_n + b_x}{2a_y s_n + b_y}. \tag{9}$$

Solving for s_n, the arclength value where the normal contacts the interface, as shown in

Fig. 4c, we get

$$s_n = \frac{-b_x(xn - x_i) - b_y(yn - y_j)}{2a_y(yn - y_j) + 2a_x(xn - x_i)}. \tag{10}$$

Substituting for xn and yn as functions of s_n by using the expressions in Eq. (8) above, one gets an equation for s_n in the form

$$\begin{aligned}&\left(2a_y^2 + 2a_x^2\right)s_n^3 + (3a_xb_x + 3a_yb_y)s_n^2 + \left(2a_yc_y + 2a_xc_x + b_x^2 + b_y^2 - 2a_yy_j - 2a_xx_i\right)s_n \\ &\quad + (c_xb_x + b_yc_y - x_ib_x - y_jy_i) = 0.\end{aligned} \tag{11}$$

The solution for s_n is obtained by the Newton–Raphson method, by providing the initial guess $s_n^0 = s_\alpha$. s_α is the arclength value at the interfacial point to which the box shown in Fig. 4a belongs. Once s_n is known, xn and yn are obtained from Eq. (9).The normal at xn, yn is then calculated using Eq. (6), the derivatives x_s, x_{ss}, y_s, and y_{ss} being readily calculated using Eqs. (8) and the value of s_n. Now for the points (x_i, y_j) lying in the block in Fig. 4a, to determine on which side of the interface the point lies, one obtains the vector

$$\hat{\lambda} = \frac{(xn - x_i)\mathbf{i} + (yn - y_j)\mathbf{j}}{\sqrt{(xn - x_i)^2 + (yn - y_j)^2}}. \tag{12}$$

By taking the scalar product $\delta = \hat{n} \cdot \hat{\lambda}$, if $\delta > 0$, the point x_i, y_j lies outside object 1, i.e., in phase 0, since the normal to the interface, by the convention adopted previously, points from phase 1 to phase 0. If $\delta < 0$, the point lies inside the interface. When the point x_i, y_j in the box lies outside object 1, the point is assigned a value of -1 and when it lies inside it is assigned a value of $+1$. This leads to the picture shown in Fig. 5a. Then, the "true" border cells are obtained such that at least one neighbour (along the x or y direction) of a point (x_i, y_j) has a point with a flag of opposite sign. The flags of the cells that are not true border cells are reset to 0. This leads to the picture in Fig. 5b. For a closed object, in which we wish to assign the phase of the control points, the overall picture at this stage will be as shown in Fig. 5c, where the flags corresponding to each point have been illustrated by means of gray levels.

In order to identify all the points inside the object with the phase $=1$ value, we proceed as follows. All the true border cells are stored in a 1D array running from 1 to the total number of border cells (designated nb). The indices of these border cells, i.e., i and j values corresponding to x_i and y_j, are also stored as $ib(1 \to nb)$, and $jb(1 \to nb)$. Now, one successively arrives at each border cell that lies within the object, i.e., a cell with a value of $+1$. The immediate neighbours $(ib \pm 1)$ and $(jb \pm 1)$ are checked for the negative value of the flag. If, say, the left cell has a flag value of -1 then one proceeds to the right and sets the value in each cell as one marches along to the value of $+1$. This procedure is stopped when the next negative flag is hit. If a particular border cell has more than one neighbour with a negative flag, then the above procedure is conducted in both directions. The result, after setting the negative cells back to zero, is as shown in Fig. 5d, where the interior of the object has been assigned the phase 1 and the exterior the phase 0.

The above procedures occur only in the vicinity of the interface every time the interface is moved. The operations involved in obtaining the information above are therefore accomplished economically. All the necessary information to be stored is placed in 1D arrays of

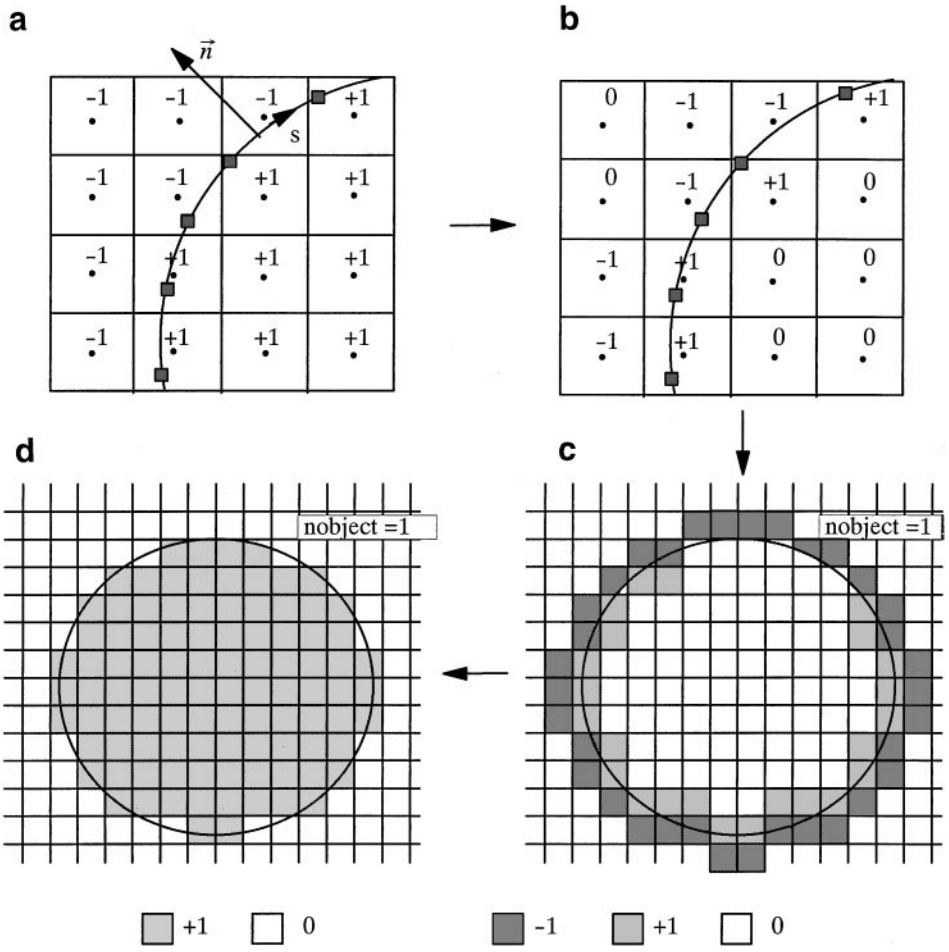

FIG. 5. (a) Identification of the side of the interface on which the grid points lie. ± 1 indicates that the point lies inside object 1; -1 indicates that the point lies outside object 1. (b) "True" border point. All points which do not have a neighbour in the opposite phase are set to 0. These are not border points. (c) Overall picture of the object after true border cells have been identified. The gray scale scheme is indicated below the figure. (d) Final picture after the phase identification process is complete.

length commensurate with the number of interfacial markers. One further aspect of this phase assignment procedure is that it carries over in a straightforward extension to 3D. The only significant difference would be that the intersection of the normal from the adjoining grid points will now be with the surface defining the object, not the curve.

III. *Solution of the Field Equations in the Presence of Internal Boundaries*

Discretization in the bulk of the domain. We have now obtained information on the boundary curves and on their location and effect on the control points x_i and y_j in the computational domain, in terms of the material properties to be assigned. Also, we now have information on where the boundary is located with respect to the grid. This is obtained from the border cell arrays as detailed in the previous section. The governing equation

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = k \nabla^2 \phi \tag{13}$$

is discretized as

$$\frac{\phi^{n+1}-\phi^n}{\delta t} = \frac{k}{2}(\nabla^2\phi^{n+1} + \nabla^2\phi^n) - \left(\frac{3}{2}\mathbf{u}^n \cdot \nabla\phi^n - \frac{1}{2}\mathbf{u}^{n-1} \cdot \nabla\phi^{n-1}\right), \qquad (14)$$

where the $O(\delta t^2)$ Crank–Nicolson discretization is adopted for the diffusion term and the $O(\delta t^2)$ explicit Adams–Bashforth method is adopted for the convection term. If an $O(h^2)$ central difference scheme is used to evaluate the spatial derivatives, we should have a nominally second-order accurate scheme in the bulk of the domain. The discrete approximation to the derivatives gives a five-point stencil in 2D and the final discrete form, at a grid point (i, j) and time level $n+1$, can be written as

$$\alpha_{i,j}\phi_{i,j}^{n+1} + \alpha_{i+1,j}\phi_{i+1,j}^{n+1} + \alpha_{i-1,j}\phi_{i-1,j}^{n+1} + \alpha_{i,j+1}\phi_{i,j+1}^{n+1} + \alpha_{i,j-1}\phi_{i,j-1}^{n+1} = \Sigma_{i,j}. \qquad (15)$$

Here the $\alpha_{i,j}$, etc., are the coefficients in the discretization corresponding to the members of the five-point stencil, and Σ is a source term containing the explicit terms as well as boundary conditions. The truncation error, due to the discrete operators mentioned above, is expected to be $O(h^2)$ and $O(\delta t^2)$ in space and time in the bulk of the domain. Usually boundary points will be unable to yield $O(h^2)$ local accuracy. Such points are few in number and if $O(h)$ accuracy can be ensured at such points the deterioration in global accuracy below $O(h^2)$ should be minimal. This requirement is along the lines of the immersed interface method of LeVeque and Li (1994).

Discretization at the immersed boundary. An $O(h)$ accurate discretization needs to be obtained at an internal boundary in the grid. Illustrating with a 1D case, the situation with an internal boundary on the grid is as shown in Fig. 6. In the figure, the point y_j lies in the solid phase, while y_{j+1} lies in the liquid phase. The two are separated at y_α by the boundary. If we are interested in discretizing the diffusion term at y_{j+1}, i.e., in the liquid phase, such that the truncation error is $O(h)$, then we proceed as follows (similar considerations will

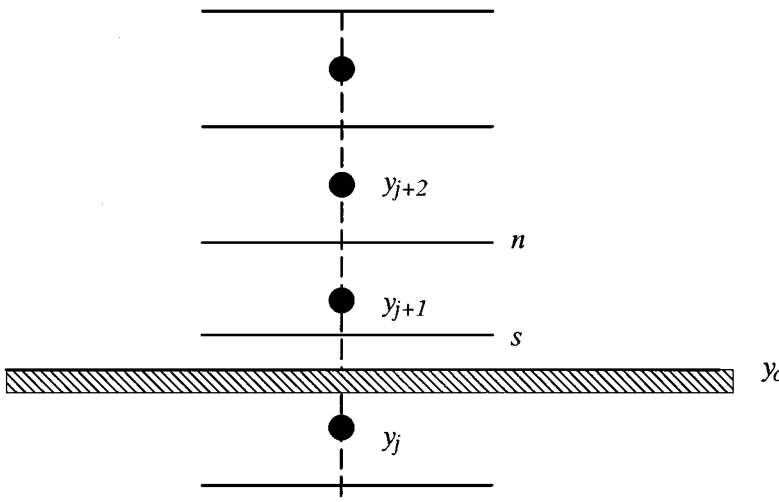

FIG. 6. Immersed boundary in the domain along the y direction. Grid point j lies in the solid phase, while $j+1$ lies in the liquid phase.

apply at y_j),

$$\left(\frac{\partial^2\phi}{\partial y^2}\right)_{j+1} = \frac{\left(\frac{\partial\phi}{\partial y}\right)_n - \left(\frac{\partial\phi}{\partial y}\right)_s}{y_n - y_s}, \tag{16}$$

where

$$\left(\frac{\partial\phi}{\partial y}\right)_n = \frac{\phi_{j+2} - \phi_{j+1}}{y_{j+2} - y_{j+1}} \tag{17}$$

$$\left(\frac{\partial\phi}{\partial y}\right)_s = \frac{\phi_{j+1} - \phi_\alpha}{y_{j+1} - y_\alpha}, \tag{18}$$

where ϕ_α is the value of the variable at the interface location, and

$$y_n - y_s = \frac{y_{j+2} + y_{j+1}}{2} - \frac{y_{j+1} + y_\alpha}{2}. \tag{19}$$

Therefore

$$\left(\frac{\partial^2\phi}{\partial y^2}\right)_{j+1} = \frac{2}{y_{j+2} - y_\alpha}\left(\frac{\phi_{j+2} - \phi_{j+1}}{y_{j+2} - y_{j+1}} - \frac{\phi_{j+1} - \phi_\alpha}{y_{j+1} - y_\alpha}\right). \tag{20}$$

It can be shown that the truncation error for the above expression is $O(h)$.

For the convection term, again taking the 1D situation, the discretization is performed as

$$V_{j+1}\left(\frac{\partial\phi}{\partial y}\right)_{j+1} = V_{j+1}\left(\frac{\phi_n - \phi_s}{y_n - y_s}\right), \tag{21}$$

where

$$\phi_n = \frac{\phi_{j+2} + \phi_{j+1}}{2} \tag{22}$$

$$\phi_s = \frac{\phi_{j+1} + \phi_\alpha}{2} \tag{23}$$

and $y_n - y_s$ is given by Eq. (19). It can again be shown that such a discretization yields $O(h)$ accuracy for the convection term at point $j+1$. Therefore in the 1D situation discretization of the governing equation using the above differencing gives $O(h)$ accuracy in cells adjoining the interface.

In the 2D case, the discretization near the interface entails the following procedures, which can be explained with reference to Fig. 7. Suppose an interface runs through the grid as shown. The point (i, j) is in the solid phase, while $(i, j+1)$, $(i, j+2)$, $(i-1, j)$, and $(i-2, j)$ are in the liquid phase. The points like (i, j) and $(i, j+1)$ which are adjacent to the interface, i.e., have an immediate neighbour in the opposite phase, have previously been identified as border cells. Therefore in assembling the discretization stencils for points in the domain, one first obtains the coefficients α and Σ in the discrete representation in Eq. (15) without regard to the presence of the interface. Then one visits each of the border cells identified previously and modifies the stencils of each point belonging to the opposite phase. The modification of the stencil is achieved by reassigning the appropriate values of α and Σ such that the differencing in Eqs. (20) and (21) is applied in the border cells. For example, in Fig. 7, when considering point (i, j) which is a border point, the stencils for

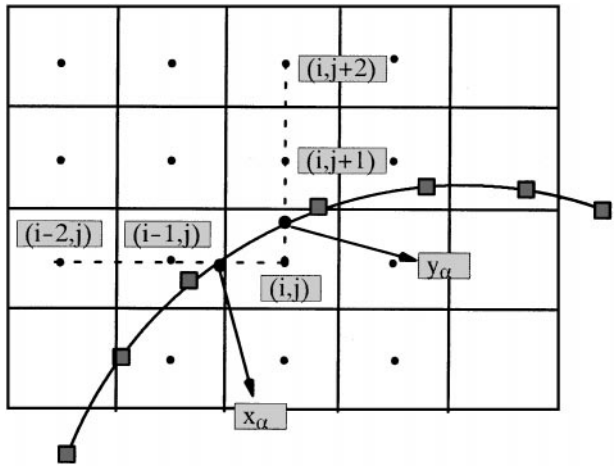

FIG. 7. Computing the gradients at the interface. The points where the interface in the cell cuts the $x =$ constant and $y =$ constant lines corresponding to the control point (x_i, y_j) are shown.

points $(i, j+1)$ and $(i-1, j)$ are modified as given by Eqs. (20) and (21) for the diffusion and convection terms, respectively. In 2D this type of operation needs to be accomplished in the x or y directions, depending on whether the neighbour in the i or j direction is in the opposite phase. If there is a cell in the opposite phase in both the i and j directions, then modifications of adjacent cells in both directions need to be performed.

Consider the point (i, j) in Fig. 7. Since the point $(i, j+1)$ lies in the opposite phase, the values of y_α and ϕ_α need to be incorporated into the discrete form for $(\partial^2\phi/\partial y^2)_{j+1}$ and $V_{j+1}(\partial\phi/\partial y)_{j+1}$ as given in Eqs. (20) and (21). To obtain the value of y_α for the cell (i, j) one computes locally the intersection of the boundary segment in that cell with the $x =$ constant line that passes through the grid point, shown as the dashed vertical line in the Fig. 7. This intersection is easily carried out using the information acquired previously regarding the interfacial point that is closest to the point (i, j). Let this interfacial point be indexed k. Then (X_k, Y_k) is the coordinate of this closest interfacial point. Now we have already obtained the functions $X(s)$ and $Y(s)$ on the interface for each point. Thus, the arclength value s_α along the interface where the curve intersects the line $x = x_i$ is obtained by solving

$$(a_x)_k s_\alpha^2 + (b_x)_k s_\alpha + (c_x)_k = x_i \tag{24}$$

for the value of s_α. Once s_α is obtained, y_α can be calculated from

$$y_\alpha = (a_y)_k s_\alpha^2 + (b_y)_k s_\alpha + (c_y)_k. \tag{25}$$

One also needs to obtain ϕ_α, i.e., the boundary condition to be applied at the location where the interface cuts the $x =$ constant line. The values on the interface are available at the locations of the markers. These are again obtained in the form $\Phi(s) = a_\phi s^2 + b_\phi s + c_\phi$. Thus, the value of ϕ_α can be obtained from

$$\phi_\alpha = (a_\phi)_k s_\alpha^2 + (b_\phi)_k s_\alpha + (c_\phi)_k. \tag{26}$$

These values can now be incorporated in the discrete form at $(i, j+1)$. Similar considerations apply for the cell $(i-1, j)$ which also lies in the opposite phase to cell (i, j). It was

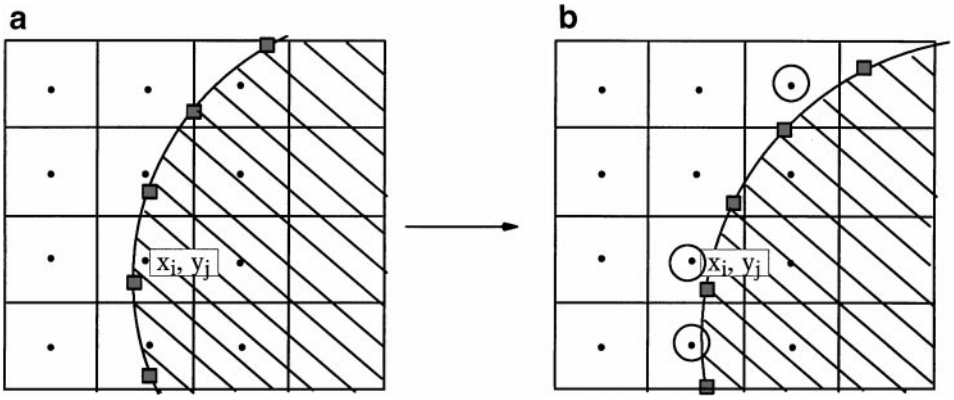

FIG. 8. Change of phase of cells due to moving boundary. (a) Location of the interface at time level n. The hatched region is solid. Grid point i, j lies in the solid phase. (b) Location of the interface at time level $n+1$. The interface has moved across the grid point (i, j) which now lies in the liquid phase. The cells that have changed phase are indicated by open circles.

pointed out by one of the reviewers that the discretization described above is similar, in the 1D case, to that described in Crank (1984, pp. 163–168) for one-dimensional fronts.

IV(A). *Moving Boundary and the Issue of Change of Material*

When a boundary immersed in the flow domain moves across the fixed Cartesian grid points, some characteristic computational issues can arise. One problem encountered at an Eulerian grid point is the discontinuous change of material following boundary motion. This is illustrated in Fig. 8. As shown there, at the left is the initial position (at time level n) of the boundary at which the cell (i, j) lies in the solid phase. Once the boundary moves (at time level $n+1$) to the position shown in Fig. 8b, there are a few cells which emerge from the solid into the liquid. For the solution of the field equation at the time level $(n+1)$, using the discrete form in Eq. (15), one requires information regarding ϕ^n at each grid point. However, for the cells that have just emerged from the solid, the grid points at which the computations are being performed have no prior history in the liquid phase; i.e., there has been a change of material at the point (i, j). Therefore there is no information on $\phi_{i,j}$ at time level n that is physically meaningful with respect to the phase or material into which it has emerged. Such newly emerged points have to be treated in a special manner. Note that purely Eulerian methods do not face this problem since in such methods the interface is not tracked as a discontinuity, but is smeared onto the grid. In the immersed boundary technique, although the interface itself is tracked explicitly, its interaction with the underlying grid is simulated by smearing the material discontinuity across the interface using a Heaviside function. However, if one wishes to avoid the smearing of the interface one has to contend with the issue of change of phase of a grid point.

A solution to this problem is obtained by analogy with boundary-fitted moving grid methods. Consider a planar interface that is being tracked using a boundary-fitted grid as shown in Fig. 9. Let the position of the interface at time level n be as in Fig. 9a. The interface is then moved to the position shown in Fig. 9b at time level $n+1$. The location of the grid point close to the interface at time level n is shown in Fig. 9b and denoted as β. If one were to regrid in the cell that has been stretched by adding one more grid point in that cell, one would have to estimate the value of the function ϕ at the freshly created grid point shown in

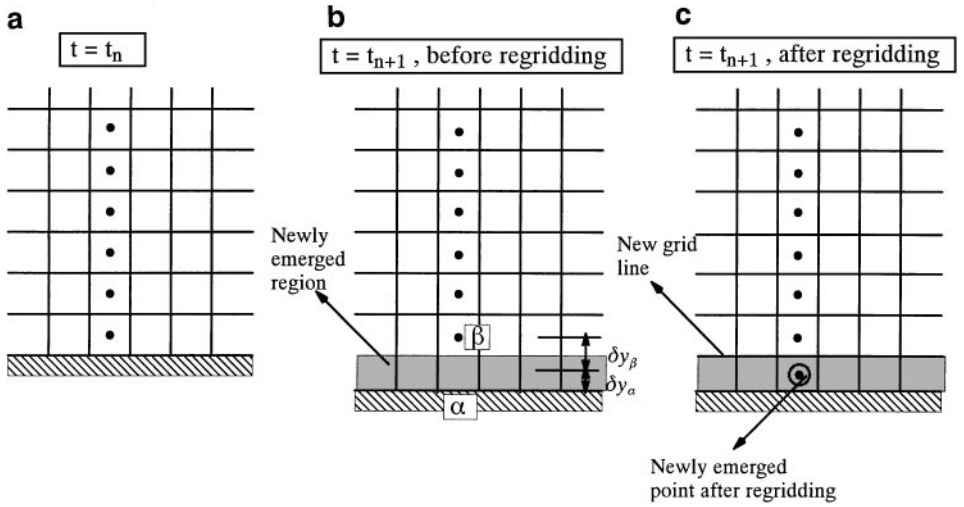

FIG. 9. Boundary-fitted grid replenishment procedure when the moving boundary stretches the grid beyond a certain desirable limit. (a) Grid at time level n. (b) Grid at time level $n+1$ after the boundary has moved. The shaded region shows the new area created in the liquid region by boundary motion. (c) Regridding is performed to replenish grid points in the stretched grid region. The open circle shows the newly created grid point.

Fig. 9c. But this point was previously (i.e., at time level n) in the solid phase and hence has no history in the liquid. In such a case, in order to obtain ϕ^{n+1} one would simply interpolate to find this value using

$$(\phi_{\text{newcell}})^{n+1} = \frac{\delta y_\alpha \phi_\beta^{n+1} + \delta y_\beta \phi_\alpha^{n+1}}{\delta y_\alpha + \delta y_\beta}. \tag{27}$$

Here ϕ_{newcell} is the value at the newly emerged point shown in Fig. 9c, ϕ_β and ϕ_α are the values at the points β and α shown in Fig. 9b, and δy_α and δy_β are as shown in Fig. 9b. Thus, at the end of time level $n+1$ we have the information at the newly created point based on the computed value of $(\phi_\beta)^{n+1}$ and the known value at the interface $(\phi_\alpha)^{n+1}$. A similar strategy is adopted in the case of the present fixed grid calculations for obtaining the value at a freshly cleared cell, as shown in Fig. 10. At time level $n+1$ the cell (i, j) has just emerged

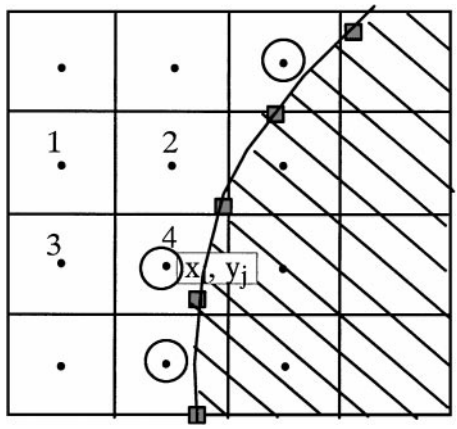

FIG. 10. Finding the value at the newly cleared cell. A linear interpolant is constructed using points numbered 1 to 4 such that the boundary value at the interface is correctly enforced.

into the liquid phase. Therefore the value of $(\phi_{i,j})^{n+1}$ is obtained by constructing a bilinear interpolant between the surrounding points and points on the interface. This interpolation is performed by using the points 1 to 4 illustrated in the figure and requiring that the value of ϕ at the interface as computed in cell 4 (which corresponds to the newly emerged cell) achieve the prescribed interface value ϕ_α. This expression takes the form

$$\alpha_1\phi_1 + \alpha_2\phi_2 + \alpha_3\phi_3 + \alpha_4\phi_4 = \phi_\alpha, \tag{28}$$

where the α's are the geometric coefficients in the bilinear interpolant. Therefore the value at the newly emerged cell is obtained from

$$(\phi_4)^{n+1} = \frac{\phi_\alpha - \alpha_1\phi_1^{n+1} - \alpha_2\phi_2^{n+1} - \alpha_3\phi_3^{n+1}}{\alpha_4}. \tag{29}$$

Note that if two adjacent cells emerge into the liquid at the same time the above formula still holds and the final values at the adjacent points are established in the course of iteration. Thus, for newly emerged cells, Eq. (15) is not employed to compute $(\phi_{i,j})^{n+1}$ for one time step. At the next time step, i.e., time level $n+2$, the cell (i, j) is no longer a newly emerged cell and the computation at that point can then proceed using Eq. (15) since the previous time step value (corresponding to time level $n+1$) is now available. However, at time level $n+2$, for the convection term, the Adams–Bashforth formula still cannot be used since the value of ϕ exists only for one previous time level. In such a case either an $O(\delta t)$ accurate scheme will have to be tolerated for an additional time step or some other two-time-level $O(\delta t^2)$ accurate method such as the Runge–Kutta method will have to be used.

IV(B). *Moving Boundary: Computing the Velocity of the Interface*

Depending on the physical problem, the velocity with which the solid–liquid interface moves can be determined in several ways. For example, if a fluid–structure interaction is being simulated, then the hydrodynamic forces acting on the object, integrated over the surface of the object, will determine the resulting motion. In solidification problems, the interface velocity is driven by the temperature field as given by the Stefan condition, Eq. (5). Taking the particular example of solidification, it is necessary to obtain the gradients of the temperature, i.e., $\frac{\partial\phi}{\partial n}$, in each phase to $O(h^2)$ accuracy in order to obtain the velocity of the interface. This can be done in different ways. First, since

$$\left(\frac{\partial\phi}{\partial n}\right)_\alpha = n_x\left(\frac{\partial\phi}{\partial x}\right)_\alpha + n_y\left(\frac{\partial\phi}{\partial y}\right)_\alpha \tag{30}$$

one can obtain gradients $(\frac{\partial\phi}{\partial x})_\alpha$ and $(\frac{\partial\phi}{\partial y})_\alpha$ to second-order accuracy and thereby evaluate the normal gradient from Eq. (30) to second order. However, in practice this is found to be tedious to implement and the results were found to be poor. We instead follow along the lines of our previous work where the gradient was evaluated by using a normal probe. This is done by extending a normal probe into each phase from the interfacial marker location as illustrated in Fig. 11. In previous work (Udaykumar and Shyy, 1995b), we used a single node on the normal probe located a distance h from the interface and described the temperature field at the node point using a biquadratic function, $\phi(x, y) = ax^2 + by^2 + cxy + dx + ey + f$, where the six coefficients were solved by locating six grid points around the node in the

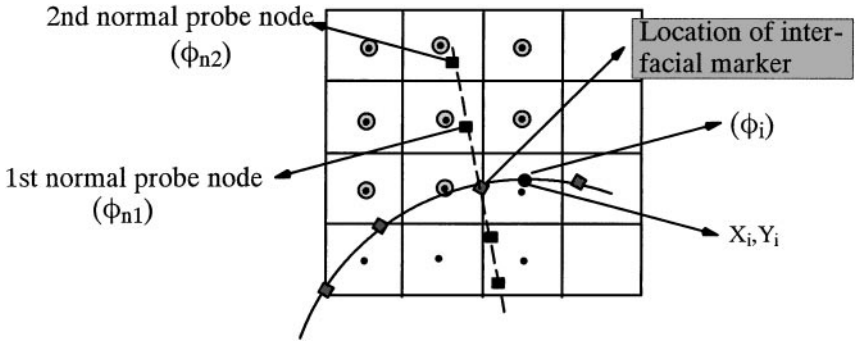

FIG. 11. Obtaining the normal gradient at the interface marker location to find the velocity of the interface in the solidification problem. Values at the normal probe nodes are obtained by bilinear interpolation.

phase in which the gradient was desired. Here we instead use two nodes on the normal probe and obtain the temperature values at each node by bilinear interpolation from the neighbouring grid points. The two nodes on the normal probe are located at distances of h and $2h$ from the interface. The points that are involved in the calculation are shown using shaded circles in Fig. 11. As can be seen in the figure, for the node proximal to the interface, one of the surrounding grid points may lie in the opposite phase. Then the value of the temperature ϕ_i and coordinates (X_i, Y_i) at the interfacial point shown by the closed circle is chosen for the bilinear interpolation. Then based on the two points on the normal probe and the known boundary value on the interface, an $O(h^2)$ estimate of the normal gradient can be found in each phase across the interface as follows:

$$\left(\frac{\partial \phi}{\partial n}\right)_{\alpha} = \frac{4\phi_{n1} - \phi_{n2} - 3\phi_{\alpha}}{2h}. \tag{31}$$

These normal gradients are then used to determine interface velocity using Eq. (5). We show later that the three-point bilinear estimate yields results that are more accurate than the two-point biquadratic estimate used in previous work for the normal probe values.

IV(C). *Implicit Update for Curvature-Driven Growth*

In solidification phenomena, when a solid advances into an undercooled melt the interface becomes unstable and complex front shapes are produced (Kessler *et al*., 1988). It has been shown in an interesting paper by Hou *et al*. (1994) that for problems involving curvature-driven growth, the presence of capillarity terms on the interface evolution equation can lead to a severe numerical stability constraint if the interface is updated explicitly. Hou *et al*. (1994) show that the criterion takes the form

$$\delta t \leq \xi \delta x^3, \tag{32}$$

where ξ is a problem-dependent quantity. This constraint is obviously extremely limiting for computations that seek to resolve finer structures on the interface with highly refined grids. An alternative formulation based on the parametrization of the interface evolution in terms of the variables θ (the angle made by the tangent to the interface with the x axis) and L (the interface length) alleviates the diffusional stability condition arising from the

capillarity effects. However, the implementation of the θ–L formulation in the general case (for instance, in the event of topological changes or nonperiodic interfaces or in 3D) can become difficult. We find that maintaining the (x, y) formulation for defining interface marker positions but coupling the interface motion with the solution of the temperature field gives a stable interface update. The stability criterion is vastly improved so that time steps given by a convective criterion can be applied. This criterion amounts to requiring that the interface traverse no more than one grid cell per time step. Therefore the interfacial motion will be computed with the time step controlled by

$$\delta t = \alpha_t \frac{h}{V_{\max}}, \tag{33}$$

where h is the grid spacing, α_t is a user-specified value (which we set to 0.1), and $V_{\max}$ is the maximum velocity of points on the interface. The fully implicit solution procedure is as follows:

0. Start time step $n+1$. Iteration number $k=0$. Initial guess for $\phi^{n+1,0}$ is given as ϕ^n. Initial guess for velocity $V^{n+1,0} = V^n$.

1. Start iteration loop. Iteration number $k = k+1$.

2. Solve temperature field (using the line solver) for a few iterations (say N_1). Do not converge temperature field fully.

3. Calculate the interface velocity (V^*) from Eq. (5). Underrelax interface velocity. In other words,

$$V^{n+1,k+1} = \alpha_v V^* + (1.0 - \alpha_v) V^{n+1,k}. \tag{34}$$

4. Update interface position:

$$\mathbf{X}^{n+1,k+1} = \mathbf{X}^n + \delta t \mathbf{V}^{n+1,k+1}. \tag{35}$$

5. Check convergence of temperature and interface velocity. If convergence is achieved go to next time step. If not go to 1 and iterate.

In the present calculations the inner iteration number N_1 is given to be 5. The number of outer iterations n required per time step is typically around 10 when the interface has not reached a steadily propagating condition but decreases to about 5 thereafter. For the dendritic growth calculation large regions of the interface reach constant velocities such as the groove and tip regions. Some variation in the number of iterations is noticed when new features are formed such as during sidebranch formation. A typical value of α_v, the velocity underrelaxation, is specified to be 0.1.

3. TEST CASES FOR SCALAR TRANSPORT

The accuracy of the current differencing scheme is demonstrated by solving the general transport equation, Eq. (1). First, we compare the solution obtained for an inviscid flow around a circular cylinder. The cylinder of radius 1 is immersed in the flow domain of size 6×6 units. The arrangement is shown in Fig. 12a. The equation to be solved is the Laplace equation for ϕ, where ϕ now represents the velocity potential function. Therefore, for this case, $u = 0$ and $f = 0$. The diffusion equation is then time-stepped to steady state. The exact

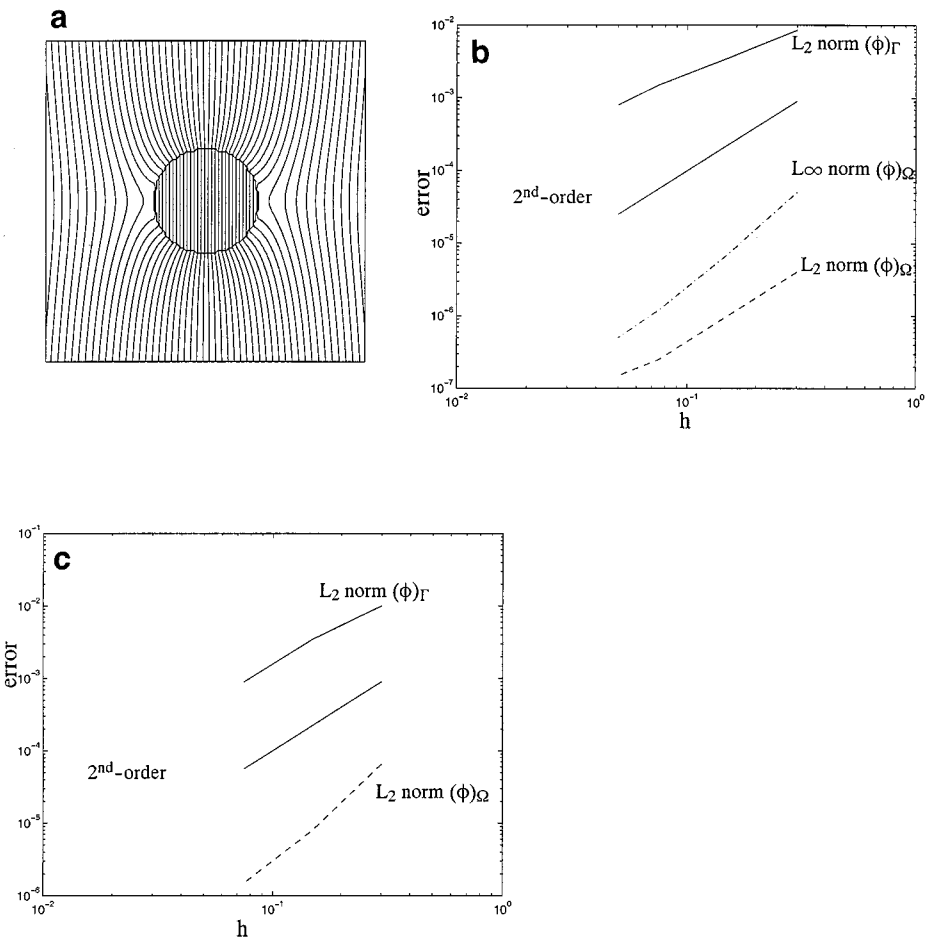

FIG. 12. Solution of the inviscid flow around a circular cylinder. (a) The potential function contours. (b) Errors in the solutions by imposing Dirichlet conditions on the cylinder. The error norms are plotted against grid size. (- - -) The L_2 norm in the solution errors in the domain Ω; (-·-) the L_∞ norm; (—) the L_2 norm of the errors in satisfying the Neumann boundary condition at the interface (Γ). (c) Errors in the solutions by imposing Neumann conditions on the cylinder. (- - -) The L_2 norm in the solution errors in the domain Ω; (—) the L_2 norm of the errors in satisfying the Dirichlet boundary condition at the interface (Γ).

solution for this problem, for the stream function, is given by

$$\phi(r,\theta) = \left(r + \frac{1}{r}\right)\cos(\theta), \tag{36}$$

where

$$r = \sqrt{(x-3)^2 + (y-3)^2}, \qquad \theta = \tan^{-1}\frac{(y-3)}{(x-3)}. \tag{37}$$

This exact solution is imposed at the boundaries of the domain and Dirichlet conditions corresponding to the exact value above are imposed on the surface of the cylinder. The computed solution in Fig. 12a shows the contours of ϕ, the potential function. The grid refinement study was performed by computing the solution for grids with 21×21, 41×41, 81×81, and 121×121 points. The L_2 norm of the error was computed for the solution over

the whole domain from the computed and exact solutions. The L_∞ norm was also obtained. These norms are defined as

$$L_2 = \sqrt{\frac{\left(\sum_{i,j}(\phi_{i,j} - \phi_{\text{exact}})^2\right)}{N^2}} \tag{38a}$$

$$L_\infty = \max|\phi_{i,j} - \phi_{\text{exact}}|, \tag{38b}$$

where N is the number of grid points along each coordinate direction. These two error norms are shown on a logarithmic scale in Fig. 12b. As can be seen in comparison with the reference second-order line, the accuracy of the calculations is indeed second order in h, the grid spacing. In Fig. 12b we also show the error in satisfying the Neumann condition for the potential at the immersed boundary. Again, the boundary condition is satisfied to second-order accuracy with respect to grid size. The same problem can also be solved by specifying the Neumann boundary condition on the cylinder due to the no-penetration condition. Therefore the grid refinement study was done by applying the Neumann condition $\frac{\partial \phi}{\partial n} = 0$ on the immersed boundary. In Fig. 12c, we plot the log(error) vs log(h) for the grids 21×21, 41×41, and 81×81. Second-order accuracy is also obtained in this case. Also shown in Fig. 12c is the error in satisfying the Dirichlet condition on the interface when the Neumann condition is imposed. In other words, with particular reference to the inviscid flow problem, this represents the accuracy in computing the potential function value at the cylindrical boundary when the no-penetration condition is imposed via the Neumann condition. It can be seen that imposing the Neumann condition does attain the Dirichlet boundary value to second-order accuracy. Thus, second-order global accuracy can be achieved when the discretization on the boundary is $O(h)$ for the case of the diffusion equation for both Dirichlet and Neumann conditions.

To study the effect of discretizing the convection term on the order of accuracy of the computed solution, we now solve the convection–diffusion equation, Eq. (14). The velocity field is assumed to be uniform, so that $\mathbf{u} = 1\mathbf{i} + 1\mathbf{j}$, and the Peclet number $\text{Pe} = |\mathbf{u}|d/k$ is 20, where d is the diameter of the cylinder and k is the diffusion coefficient. The boundaries of the domain are maintained at a value of $\phi = 1$, while a Dirichlet condition is imposed on the surface of the cylinder where $\phi = 0$. Figure 13a shows the isotherm contours and the clustering of the isotherms in the boundary layer close to the cylinder surface and at the top right corner of the domain due to the convection and boundary conditions imposed. In the absence of the exact solution, the solution on the finest grid level, namely the 221×221 grid, is taken to represent the exact solution and the errors for the other grids, namely 21×21, 61×61, 101×101, 141×141, and 181×181, are computed with reference to the solution on the 221×221 grid. The L_2 norm obtained is plotted in Fig. 13b. The maximum error in the domain is also obtained and plotted. As can be seen from the figure the discretization procedure leads to an $O(h^2)$ accurate solution.

We next solve a moving boundary problem using the methods developed for the solidification problem discussed in Section 2. One problem in phase change that has an exact solution is the melting of a flat interface. This so-called Neumann problem has been used in previous work with the cut-cell method (Shyy *et al*., 1996; Udaykumar *et al*., 1996). The diffusion equation is solved in each phase, solid and liquid, across the melting front,

$$\frac{\partial T}{\partial t} = \nabla \cdot \alpha_i \nabla T, \qquad i = \text{L, S}, \tag{39}$$

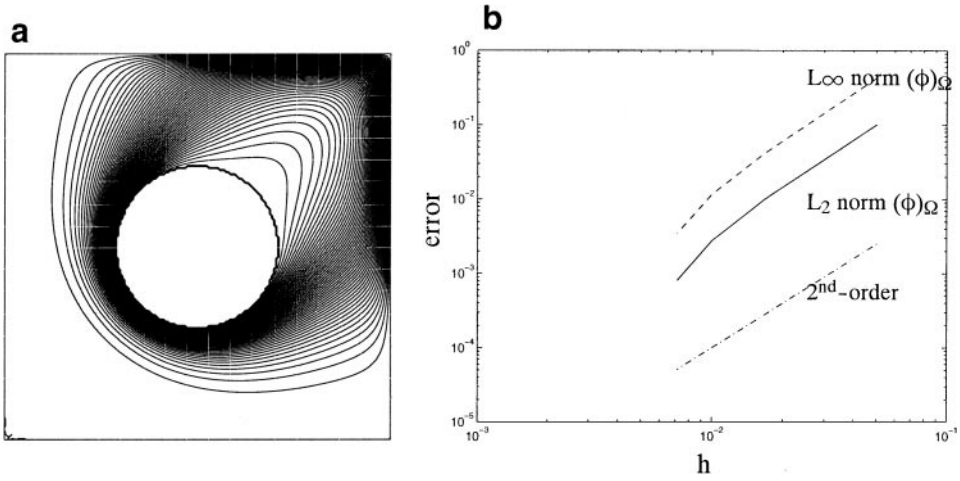

FIG. 13. Solution of the convection–diffusion equation with an immersed circular boundary. The Reynolds number is 20. (a) The isotherm contours showing the clustering near the front surface of the cylinder where large gradients occur. (b) The errors in the solution plotted against grid size. The finest grid solution is taken as the reference in obtaining the solution. (- - -) The L_∞ norm of the error; (—) the L_2 norm of the error; (-··-) reference second-order line.

where α_i is the diffusion coefficient given by $k_i/\varrho_i C_{\mathrm{p}i}$, where k is the conductivity, ϱ is the density, and C_{p} is the specific heat at constant pressure and the subscript i indicates the phase L (liquid) or S (solid). The boundary condition on the front is imposed as the melting temperature T_{m}. The interface then is moved using the Stefan condition, Eq. (5). The temperature, length, and time are nondimensionalised as

$$\phi = \frac{T - T_{\mathrm{m}}}{T_{\mathrm{L}} - T_{\mathrm{m}}}, \qquad Y^* = \frac{Y}{Y_{\mathrm{L}}}, \qquad t^* = \frac{t}{\alpha_{\mathrm{ref}}/Y_{\mathrm{L}}^2}, \tag{40}$$

where Y_{L} is the extent of the domain, T_{L} is the temperature in the liquid at the boundary of the domain, α_{ref} is a reference diffusion coefficient, and the asterisk indicates nondimensional values. The diffusion equation takes the form

$$\frac{\partial \phi}{\partial t} = \nabla \cdot \alpha_i \nabla \phi, \tag{41}$$

where α_i is the diffusion coefficient of phase i. The interface is then at a temperature $\phi = 0$ and the front velocity is given by

$$V_n^* = \mathrm{St}\left(\frac{k_{\mathrm{s}}}{k_{\mathrm{L}}}\left(\frac{\partial \phi}{\partial n}\right)_{\mathrm{s}} - \left(\frac{\partial \phi}{\partial n}\right)_{\mathrm{l}}\right). \tag{42}$$

St is a nondimensional parameter called the Stefan number given by $\mathrm{St} = C_{\mathrm{pL}}(T_{\mathrm{L}} - T_{\mathrm{m}})/\rho_{\mathrm{L}} L_{\mathrm{f}}$ and denotes the ratio of sensible to latent heat effects. A schematic of the computational setup for the problem is shown in Fig. 14a, where the nondimensional domain size and boundary conditions are indicated. Melting proceeds from the top wall and the front progresses in the downward direction. The exact solution to this problem takes the following

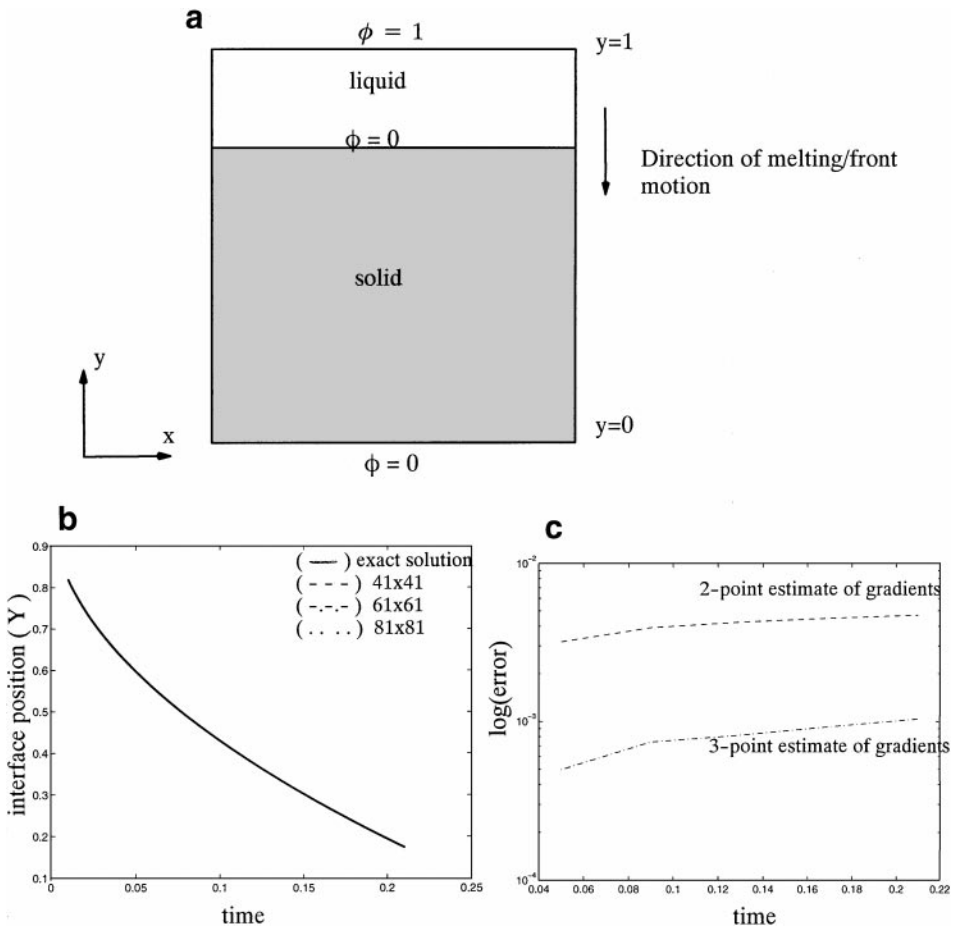

FIG. 14. Solution of the melting of a flat interface. Melting is initiated at the top of the domain. The front moves downward. (a) Schematic of the problem geometry and boundary conditions. (b) Comparison of interface locations obtained using different grid sizes. (c) Errors in interface location from the calculations using the two velocity estimates against the time elapsed. (- - -) Error due to the two-point estimate of interface velocity; (-··-) error due to the three-point estimate of interface velocity.

form (Crank, 1984): The temperature in the liquid is given by

$$\phi(y,t) = 1 - \operatorname{erf}\left(\frac{y-1}{2\sqrt{t}}\right) \Big/ \operatorname{erf}\lambda. \tag{43}$$

The temperature in the solid is $\phi = 0$. The interface location is given by

$$S(t) = 1 - 2\lambda\sqrt{t}, \tag{44}$$

where λ is given by the root of the equation,

$$\lambda e^{\lambda^2} \operatorname{erf}\lambda = \frac{\mathrm{St}}{\sqrt{\pi}}, \tag{45}$$

and St is the Stefan number as defined above. The initial conditions are specified in computing the solution based on the exact solution. The initial temperature field is therefore

specified as

$$\phi(x, y, t_0) = 1 - \operatorname{erf}\left(\frac{y-1}{2\sqrt{t_0}}\right) \Big/ \operatorname{erf}\lambda \tag{46}$$

and the interface is initially placed at

$$S(t_0) = 2\lambda\sqrt{(t_0)}, \tag{47}$$

where t_0 is the initial time at which the computation is started.

The computations were performed for a Stefan number of 2.85 for which the value of λ is 0.9. In Fig. 14b, the interface locations from the computations are compared for different grid sizes (41×41, 61×61, and 81×81) with the exact interface location (solid line). The exact and computed solutions are indistinguishable in the scale of the plot. In Fig. 14c the errors in interface location from two interface velocity estimations are compared. In the first case (dashed line) the velocity estimate is obtained by a two-point estimate for the normal gradient of the temperature which appears in the expression for the interface velocity, Eq. (42). This two-point estimate was used in previous work (Udaykumar *et al*., 1996). The procedure involves inserting a normal probe of length h, the grid spacing, into each phase from the interface. The temperature is obtained at the end of the normal probe by describing a biquadratic function for the temperature field around the end of the normal probe in each phase. Then the normal gradient is obtained from a two-point estimate based on the value obtained at the end of the normal probe and the known value at the interface. A second procedure has been used in Fig. 14c based on a three-point estimate of the normal gradients described in Section 2. In Fig. 14c the two-point estimate for interface velocity results in the interface location curve that deviates from the exact solution and the error is shown by the dashed line. The three-point estimate (represented by a dash-dot line) gives interface locations that are indistinguishable from the exact solution on the scale of the plot. The temporal variation of the errors incurred in the interface location using these two estimates are shown in Fig. 14c. As the interface evolves the accumulation of error in time appears to be minimal. However, it is noted that the three-point estimate of interface velocity gives an order-of-magnitude lower error than the two-point estimate.

While the three-point estimate clearly gives better accuracy in tracking the interface, Fig. 15 indicates that it does not yield the expected second-order accuracy that was indicated from the truncation error estimates obtained in Section 2. In Fig. 15a we plot the L_2 norm of the temperature field errors against the grid spacing h at two time instants (after 100 time steps ($t = 0.1$) and 200 time steps ($t = 0.2$)). A reference second-order line is also shown for comparison (dashed line). In Fig. 15b we show the errors in computed interface location plotted against the grid spacing for the above two time instants. It is noted that while the temperature field in the domain itself shows second-order behaviour, the interface location error shows behaviour closer to first order. This indicates that the advection of the interface, i.e., the interface velocity, is only first-order accurate in space. At longer times, i.e., after 200 time steps, when the interface has traversed a large part of the domain, the situation regarding the errors remains the same, as shown in Fig. 15b. Two interesting observations can be made from the figure. First, although the three-point estimate for the normal gradient is nominally second-order accurate, the plot of log(error) vs log(h) for interface location indicates only first-order accuracy. Second, the field equation is itself solved to second-order accuracy in h, despite the boundary location being only first-order accurate. The first

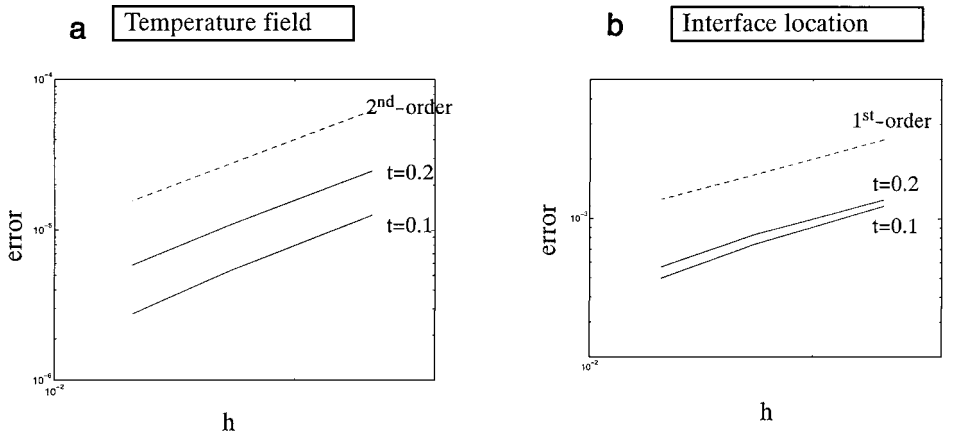

FIG. 15. Errors in the solutions obtained for the melting of the flat interface. The log(error) vs log(h) curves are shown at two different time instants, after 100 time steps ($t = 0.1$) and after 200 time steps ($t = 0.2$). (a) L_2 norm of errors in computed temerature field at the two time instants plotted against grid size. The reference second-order line is also shown. (b) L_2 norm of errors in the computed interface location. The reference first-order line is also shown.

effect can be explained based on the fact that the three-point estimate of the gradient of temperature is second-order accurate only if the temperature field is exactly specified. Since the computed temperature field itself is second-order accurate, and not exact, the gradient information that can be extracted from this computed temperature field can be at best first-order accurate. That the temperature field as computed from the discretization in Eqs. (15) and (20) and the interface velocity as computed from Eqs. (31) and (42) are $O(h^2)$ can be seen as follows. When the interface is advected by specifying the exact value of the velocity at each time step the error in the computed temperature field is seen to be $O(h^2)$ as shown in Fig. 16a. The L_2 norm of the temperature field error is plotted against grid size in that

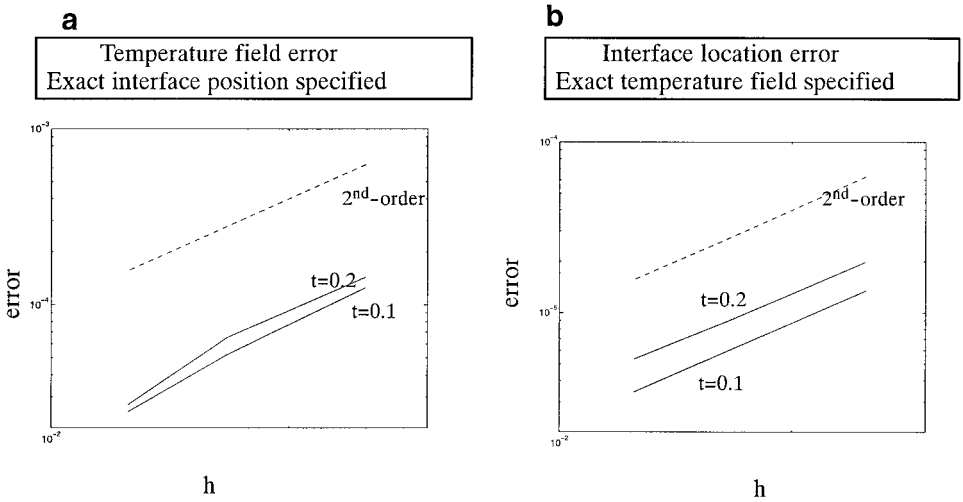

FIG. 16. Errors obtained by decoupling the temperature field and interface position in the 1D melting problem. (a) log–log plot of temperature field errors. The interface position is specified exactly and the temperature field is computed. (b) log–log plot of interface position errors. The temperature field is specified exactly, the interface velocity is computed from the given temperature field, and the interface is advected with this velocity.

figure. Now, when the exact temperature field is specified in the domain and the interface is advected by computing the velocity according to Eq. (42), the L_2 norm of the interface position error is obtained and shown in the log–log plot against grid size in Fig. 16b. Clearly, these two calculations are second-order accurate. The fully coupled calculation shown in Fig. 15 yields second-order accuracy in the temperature field but first-order accuracy in the interface position.

We show that the above accuracy estimates carry over into the 2D case. Here the stable freezing of a circular interface is computed. An initial circular seed is placed in the center of a square of dimension 1×1 units. The circle is grown by extracting heat by a heat sink located at (0.5, 0.5). An exact solution to this Stefan problem exists (Carslaw and Jaeger, 1959; Juric and Tryggvason, 1996), where the temperature field in the solid is given by

$$T(r,t) = \frac{Q}{4\pi}\left\{\mathrm{Ei}\left(\frac{-r^2}{4t}\right) - \mathrm{Ei}(-\lambda^2)\right\}, \tag{48}$$

where Q is the heat sink strength, r is the radial coordinate, λ is a constant, and Ei is the exponential integral given by

$$\mathrm{Ei}(x) = \int_{-\infty}^{x} \frac{e^v}{v}\, dv. \tag{49}$$

In the liquid phase,

$$T(r,t) = 1 - \frac{\mathrm{Ei}(-\beta r^2/4t)}{\mathrm{Ei}(-\beta\lambda^2)}, \tag{50}$$

where β is the ratio of thermal diffusivities α_s/α_l. The radius of the freezing circle is given by

$$R(t) = 2\lambda\sqrt{t}, \tag{51}$$

where λ is the root of the equation

$$Q = 4\pi e^{\lambda^2}\left(\frac{\lambda^2}{\mathrm{St}} - \frac{k_l e^{-\beta\lambda^2}}{k_s \mathrm{Ei}^{-\beta\lambda^2}}\right). \tag{52}$$

In line with the work of JT96 we specify the value of temperature at the grid points at the center of the domain in the solid phase. This is to circumvent the singular behaviour at the heat sink location.

The results for the steady growth of the initially circular crystal are shown in Fig. 17 for a Stefan number of 1.0. The material properties in the solid and liquid are taken to be equal. The exact radius of the circle was compared with the computed value and the errors obtained for three grids 61×61, 81×81, and 101×101. The L_2 norm of the temperature field error is shown on a log–log plot in Fig. 17a. The second-order accuracy of the temperature field is maintained in the 2D case. Similar to the situation in 1D, the interface position is obtained with first-order accuracy as shown in Fig. 17b. The computed and exact interface locations for 2D stable freezing for a lower Stefan number of 0.1 are shown in Fig. 18a. The order of accuracy of the temperature field calculation as well as that of the interface position follows the same trend as that for the higher Stefan number case as seen from Figs. 18b and 18c. When the Stefan number is low the latent heat released at the interface is high compared to the sensible heat flux at the interface. From Eq. (42), there is therefore a significant discontinuity in the gradients of the temperature field at the interface. When an

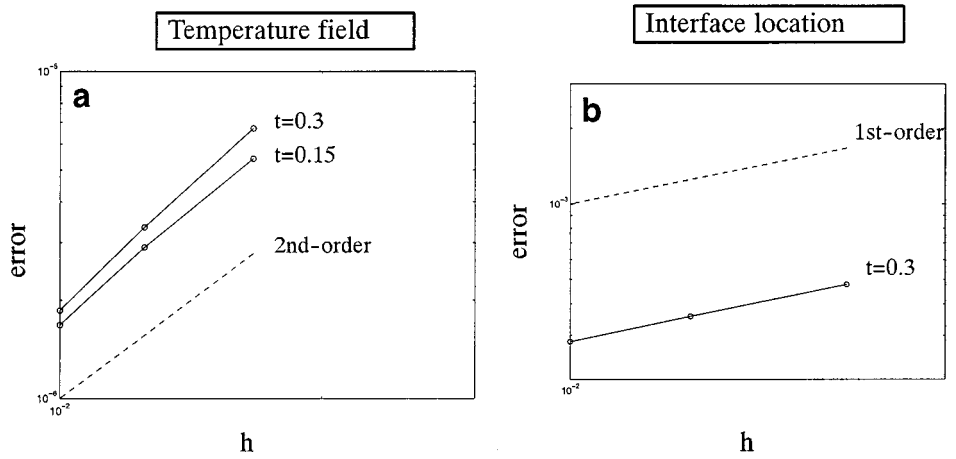

FIG. 17. 2D stable freezing (Stefan) problem. Stefan number = 1.0. (a) log–log plot of the temperature field errors vs grid size. (b) log–log plot of the interface position errors vs grid size.

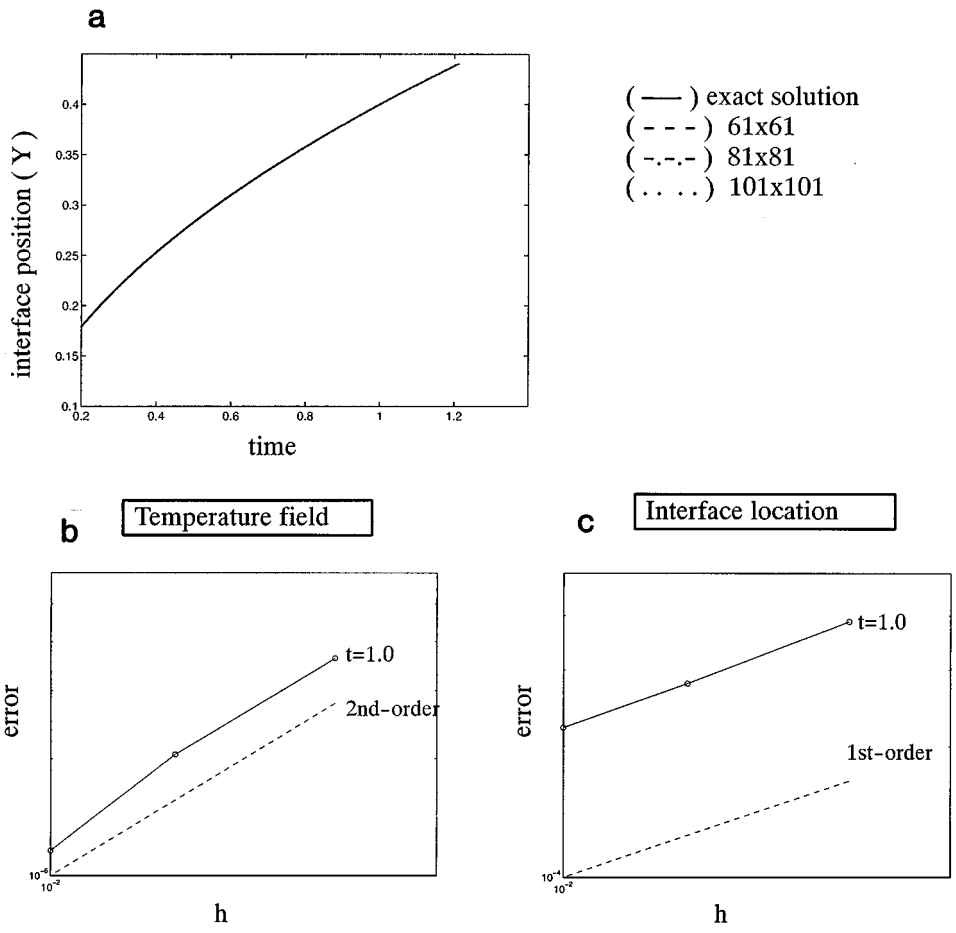

FIG. 18. 2D stable freezing (Stefan) problem. Lower Stefan number, St = 0.1. (a) Interface position; comparison of exact and computed radius of the freezing circle computed using the grids indicated. (b) log–log plot of the temperature field errors vs grid size. (c) log–log plot of the interface position errors vs grid size.

Eulerian, i.e., fixed grid, method is used to compute the low Stefan number situation, this discontinuity is smeared over a few grid points, leading to a deterioration in accuracy. This behaviour for the low Stefan number case is mentioned by JT96. In the present case since the interface is maintained as a discontinuity and one-sided differences are used in cells adjacent to the interface, no difference should exist between calculations at the different Stefan numbers. This can be seen from comparing Figs. 17 and 18. The error in predicted interface position in both cases is around 0.01%. Thus, although the interface errors appear to converge with first order with respect to the grid size, the calculated interface positions are highly accurate irrespective of the strength of the discontinuity in the interface temperature gradients.

We now demonstrate that the treatment of the property jumps by the present sharp interface model is accurate and that the presence of such jumps does not alter the accuracy findings reported above. The stable solidification problem above is computed for a Stefan number of 1.0 with a jump in the conductivities, i.e., $k_l/k_s = 0.2$. Note that this jump is treated as a sharp discontinuity and one-sided differencing is involved at grid points near the interface. Therefore, unlike in purely Eulerian methods the jump in material property is not distributed over a region of the grid. The computed and exact interface positions are shown in Fig. 19a.

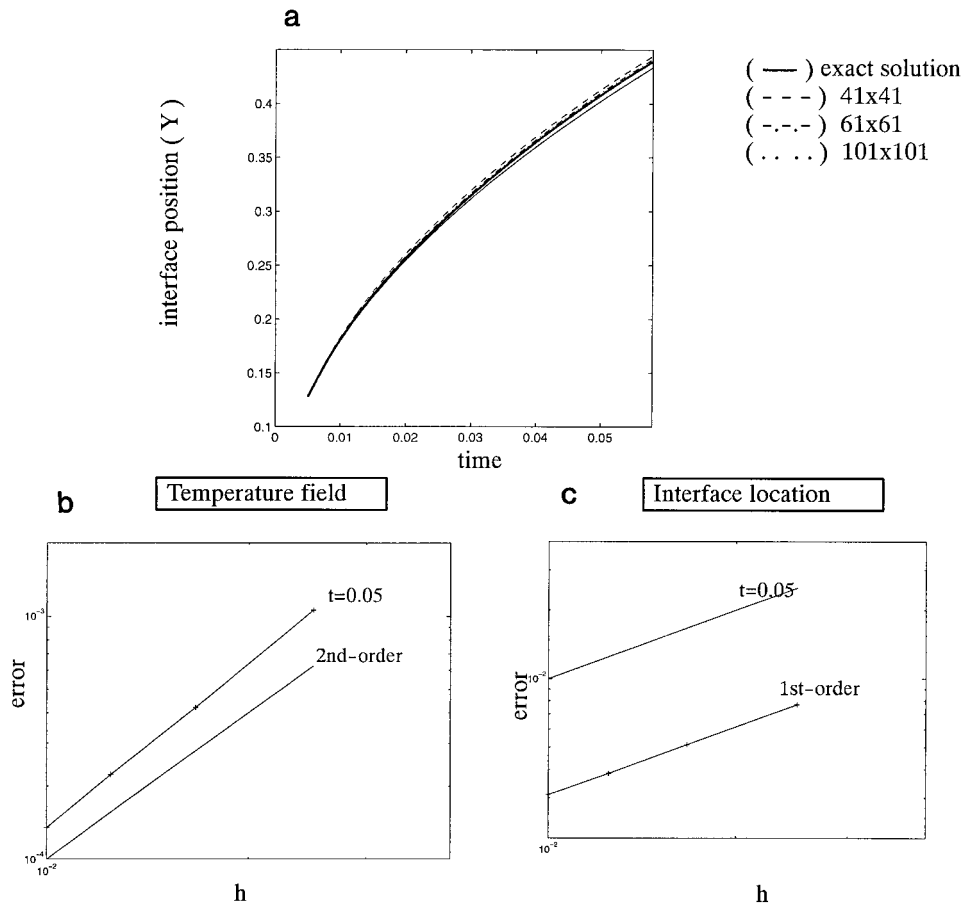

FIG. 19. 2D stable freezing (Stefan) problem with discontinuous material property. $k_l/k_s = 0.2$. Stefan number $St = 1.0$. (a) Interface position; comparison of exact and computed radius of the freezing circle computed using the grids indicated. (b) log–log plot of the temperature field errors vs grid size. (c) log–log plot of the interface position errors vs grid size.

The interface position converges to the exact value with grid refinement. It is shown in Fig. 19b that the temperature field computed is second-order accurate, while the interface position, as shown in Fig. 19c, is first-order accurate. Note that this result corresponds with the previous cases of equal material properties across the interface. Therefore, as expected, the jump in property across the interface does not affect the order of accuracy of the numerical method used here.

We next look at computations of unstable phase boundaries that may arise in many systems (Pelce, 1988; Kessler *et al*., 1988). Such instabilities contribute to phenomena such as viscous fingering in porous media, modeled by Hele–Shaw flow (DeGregoria and Schwartz, 1986; Hou *et al*., 1994), and the formation of cellular and dendritic microstructures in solidification from the melt (Wheeler *et al*., 1992; Juric and Tryggvason, 1996). The physics underlying such pattern forming instabilities has been reviewed by Langer (1980) and Kessler *et al*. (1988). As mentioned in Section 1, the numerical computation of curvature-driven growth of fronts, such as in dendritic growth and Hele–Shaw flows, places time step restrictions that are extremely severe unless measures are taken to remove numerical stiffness. As discussed before, in our computations the update of the interface is coupled with the evolution of the temperature field and this implicitness allows the choice of time step to be restricted to a convective stability limit. In each of the calculations below we place a seed crystal in a domain with insulated boundaries. The initial condition is a uniformly undercooled melt with the conditions

$$\begin{aligned} T(\mathbf{x}, 0) &= \mathrm{St} \qquad \text{in the liquid} \\ T(\mathbf{x}, 0) &= 0 \qquad \text{in the solid.} \end{aligned} \tag{53}$$

Note that the initial condition applied here corresponds to a nucleus of solid placed instantaneously in an undercooled melt. The boundary condition at the interface for the temperature includes the effect of capillarity and attachment kinetics and is given, in nondimensional form (Juric and Tryggvason, 1996), by a generalized Gibbs–Thomson condition:

$$T_i + \left(\frac{c_\mathrm{l}}{c_\mathrm{s}} - 1\right) T_i^2 + \sigma(\theta)\kappa + \mu(\theta) V_i = 0. \tag{54}$$

For the surface tension parameter σ and attachment kinetic parameter μ, we take the forms given by Almgren (1993):

$$\sigma(\theta) = \sigma\left(1 + A_\mathrm{s}\left(\tfrac{8}{3}\sin^4\left(\tfrac{1}{2}m(\theta - \theta_0)\right) - 1\right)\right) \tag{55}$$

$$\mu(\theta) = \mu\left(1 + A_\mathrm{k}\left(\tfrac{8}{3}\sin^4\left(\tfrac{1}{2}m(\theta - \theta_0)\right) - 1\right)\right). \tag{56}$$

The integer m decides the anisotropy of the interfacial parameters σ and μ. In the following we use $m = 4$ (fourfold symmetry) and $m = 6$ (sixfold symmetry).

In order to verify that capillarity effects are adequately captured in the application of the Gibbs–Thomson effect, Eq. (54), we first determine that the nucleation radius is correctly produced by the algorithm. To effect this we undercool the melt, i.e., set the temperature in the liquid to the value $T(x, y, 0) = -\mathrm{St} = -0.5$. The nucleation radius is then given by $R^* = \sigma/\mathrm{St}$, where σ is the nondimensional surface tension (=1 in our case). We then place circular crystals of radius $0.99R^*$, R^*, and $1.1R^*$ at the center of the domain of size 4×4 and track the evolution of the interface. Note that the implicit interface update allows a time step of $O(10^{-2})$ to be used in time stepping the interface. When an explicit update was

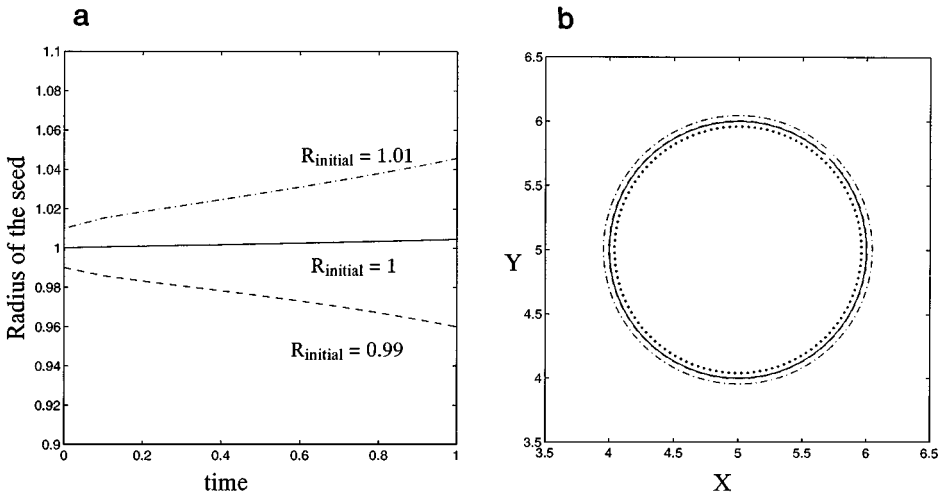

FIG. 20. Verification of the nucleation radius. Stefan number = 0.5. (a) Radius of circles with initial radii equal to, greater than, and less than the nucleation radius. (b) The shapes of the circular crystals after the time elapsed in (a). (—) R_{initial}; (- - -) $R_{\text{initial}} = 1$ after $t = 1$; (-·-) $R_{\text{initial}} = 1.01$; after $t = 1$; (· · ·) $R_{\text{initial}} = 0.99$ after $t = 1$.

employed the method became unstable and the interface was destroyed in a few time steps. We found that the time step at which an explicit update was stable agreed very well with the criterion identified by Hou *et al*. (1994) and given in Eq. (32). As shown in Fig. 20a, for the duration of time computed, the crystal with $R < R^*$ shrinks, while that with $R > R^*$ expands and $R = R^*$ changes only slightly due to numerical errors in computing the curvature. The shapes of the three interfaces after the time duration elapsed are shown in Fig. 20b. The circle with $R = R^*$ is shown to have changed only very slightly from its initial shape.

We next perform a grid refinement study to demonstrate convergence of the interfacial evolution with grid refinement. For this case we choose the system adopted by JT96. The computations are performed in a square domain of dimension 4×4 units. The initial interface is a fourfold symmetric structure placed at the center of the domain with a radius

$$R = 0.1 + 0.02\cos(4\theta). \tag{57}$$

This represents a circle with a small fourfold symmetric perturbation on its surface. The melt is undercooled and the Stefan number chosen is $\text{St} = 0.5$. The surface tension $\sigma = 0.002$ and the kinetic parameter $\mu = 0.002$. The interfacial parameters are isotropic. The results are shown in Fig. 21. We perform computations on 101×101, 201×201, and 401×401 grid sizes. For the 401×401 mesh, as shown in Fig. 21b, we also explore the effect of an initial orientation of the nucleus away from the grid directions. In the case shown the initial seed was oriented at 37° from the horizontal. In Fig. 21a, the interface shapes for the three grids are compared at time intervals 0.2 apart. It can be clearly seen that as the grid is refined the interface shapes converge rapidly and the solution on the 401×401 grid can be considered to be grid independent. The tip-splitting behaviour is shown by our solution in agreement with established physics. The interface shape obtained finally is seen to differ somewhat from that of JT96. However, even for the 101×101 grid, our solutions seem to indicate close resemblance of the result to the finer grid cases, while JT96 appear to show very different interface behaviour for the 101×101 calculation. In our simulations the coarse

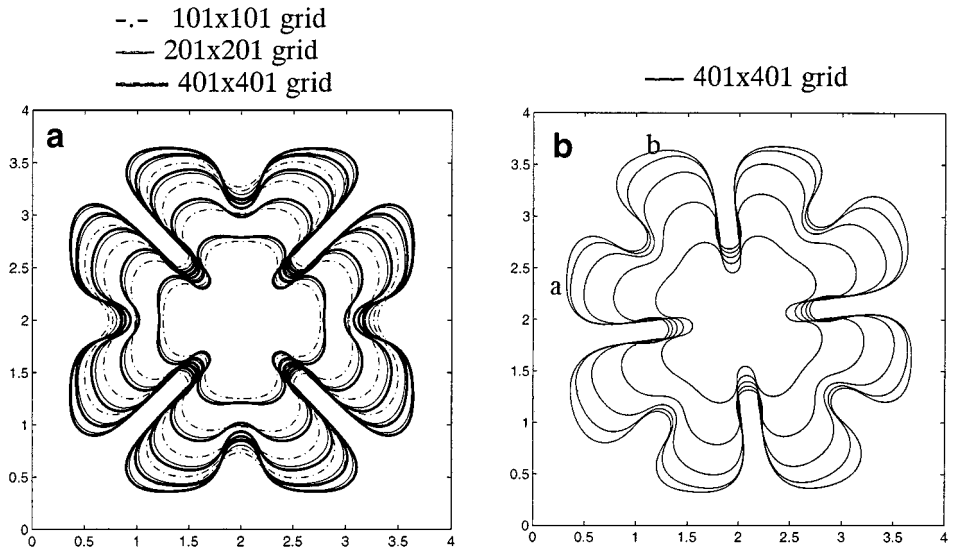

FIG. 21. Grid effects study for the case of unstable solidification with isotropic surface tension. Stefan number $\mathrm{St} = 0.5$, surface tension parameter $\sigma = 0.002$, and interface kinetics parameter $\mu = 0.002$ (no anisotropy A_s, $A_k = 0$). The interface is shown at nondimensional time instants 0.2 apart. (a) Grid refinement and convergence of the interface shape with decreasing grid spacing. (b) Orientation effect explored by orienting the initial seed at 37° from the horizontal.

grid calculations appear to show greater effects of dissipation and hence the interfacial features are coarser. This tendency is also indicated in the calculations of Hele–Shaw flows by Hou *et al*. (1997). In contrast, the calculations of JT96 for this case appear to show finer scales of the interface for the smaller grid density (i.e., 101×101 grid). While there appear to be multiple tip-splitting events leading to a highly corrugated structure for the 101×101 grid, there are two tip-splitting events for the 201×201 case and one event for the finer grids. In our case there is only one tip-splitting event for all grids. Both our method and that of JT96 explicitly track the interface. However, while the present method calculates the interface evolution in the sharp interface limit and does not distribute the latent heat over a finite region of the grid, the method of JT96 does. The initial conditions applied by us are somewhat different from those of JT96. However, the initial conditions result only in a short-time transient behaviour and application of the same initial conditions as theirs did not appear to significantly alter the long-time behaviour. In the context of benchmarking the results for this particular case we point out that Chen *et al*. (1997) have simulated this same case using the level-set method. In their case the coarse grid solution behaves similarly to ours in that the interface instability only shows coarse features and the features sharpen as the grid is refined. However, in their work, it appears that the interface is less unstable than ours. For instance, on a 400×400 grid, a tip-splitting instability is only just beginning at the final instant of their calculation. Thus, it appears that even the converged results for this case using the immersed boundary technique (JT96), the current sharp interface method, and the level-set method of Chen *et al*. (1997) do not agree on the rate or nature of the instability in the highly nonlinear stages. Karma and Rappel (1996) have used the phase-field method to test the theory for unstable solidification (Kessler *et al*. (1988)) and indicate the need to benchmark calculations in order to ascertain whether the theory is indeed applicable for a wide range of operation of dendrites. It is clear that since the methodologies for performing

these types of calculations are beginning to evolve to fairly well-developed algorithms, benchmarking of the computations is rapidly becoming necessary. With respect to the effects of grid orientation, we show in Fig. 21b that the principal evolutionary characteristics of this isotropic case are free of grid anisotropy effects. Only in the final stage of the growth when the interface approaches the outer boundary do we detect differences in the results from the two orientations. This is because even for the relatively high Stefan number computed (i.e., $\mathrm{St} = 0.5$) the square domain begins to influence the interface at the later times. The thermal boundary layer ahead of the interface interacts with the square domain and begins to act so as to flatten the interface along the sides of the domain. This effect is clearly seen in the last time instant picture of the interface in Fig. 21b. In both Figs. 21a and 21b therefore one would ascribe the flattening of the interface at the final instants to the square shape of the domain and the application of an adiabatic boundary condition there. The flattening effect is due to the global thermal transport interacting with the square domain and not due to grid anisotropy effects. This is supported by the observation that the lobes of the tip-split interface that are closer to the outer boundary (due to the orientation of the initial seed with respect to the square domain) such as lobe *a* have flattened out more than the lobes such as *b* that are farther away from the sides. In this case the surface tension is isotropic, yet grid anisotropy has no influence on the interface shape.

We look at the effects of discontinuous properties across the interface and assess the effect of such discontinuities on the stability of the interface. Four cases are shown in Fig. 22. These correspond to (a) $k_\mathrm{l}/k_\mathrm{s} = 0.2$, (b) $k_\mathrm{l}/k_\mathrm{s} = 0.5$, (c) $k_\mathrm{l}/k_\mathrm{s} = 2$, and (d) $k_\mathrm{l}/k_\mathrm{s} = 100$. Contrary to the discussion provided by JT96, the interface in the case of large solid conductivity appears to be stabilized. JT96 argue that the only role of the conductivity is to accelerate the instability development and the conductivity jump at the interface should have no impact on the stability of the interface itself. This is not supported by our calculation in Fig. 22a. We show that for large enough solid phase conductivity, the instability can indeed be stabilized (for the given values of surface tension and kinetic parameters). Although from Eq. (42) the ratio of solid to liquid phase conductivities plays the role of increasing or decreasing the interface velocity, for equal specific heats, c_s and c_l, the ratio $k_\mathrm{l}/k_\mathrm{s} = \alpha_\mathrm{l}/\alpha_\mathrm{s}$, the ratio of thermal diffusivities. Therefore, when the thermal conductivity in the solid is large, heat diffuses in the solid faster than in the liquid. This leads to a tendency of the deformed initial nucleus to revert to a circular shape and the initial perturbations of the interface are smoothed out by the stabilizing diffusion field in the solid. As the diffusivity in the solid decreases, as in Fig. 22b, where $k_\mathrm{l}/k_\mathrm{s} = \alpha_\mathrm{l}/\alpha_\mathrm{s} = 0.5$, this stabilizing mechanism weakens, and in this case the instability of the interface ensues but at a slower rate than in the case shown in Fig. 21. Note that even in Fig. 22b diffusion of heat in the solid phase appears to lead to lower curvature values at the interface, resulting in a coarser structure. When the ratio of conductivities $k_\mathrm{l}/k_\mathrm{s} > 1$, the instability of the interface is enhanced. As seen in Fig. 22c, a narrower groove is formed in the development of the structure and the instability occurs more rapidly than in Fig. 21. The final shape shown in Fig. 22c is at $t = 0.3$, which can be contrasted with the final shape in Fig. 21, which is at $t = 1.0$. However, in agreement with JT96, it can be noticed that increasing the liquid phase conductivity leads to an acceleration of the instability without significantly altering the nature of the instability itself. In Fig. 22d, we show the case where the ratio of conductivities is very large, i.e., $k_\mathrm{l}/k_\mathrm{s} = 100$. It is clear from this figure that the increased conductivity in the liquid accelerates the development of the instability. Here the interface is already approaching a second tip-splitting event, in contrast to the single tip-splitting event observed in the previous cases. If the interfaces

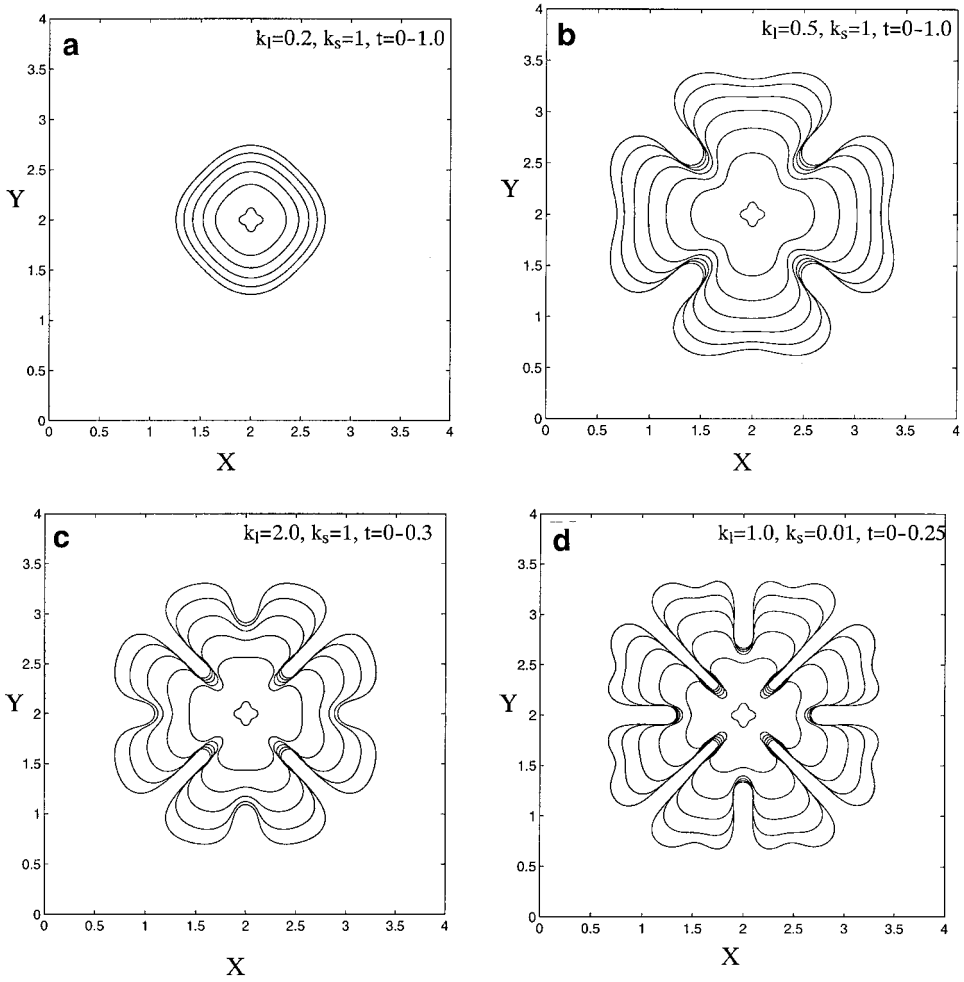

FIG. 22. Effect of property jumps across the interface for the case of unstable solidification with isotropic surface tension. Stefan number $\mathrm{St} = 0.5$, surface tension parameter $\sigma = 0.002$, and interface kinetics parameter $\mu = 0.002$ (no anisotropy A_s, $A_k = 0$). (a) $k_l = 0.2$, $k_s = 1.0$, time from 0 to 1.0; (b) $k_l = 0.5$, $k_s = 1.0$, time from 0 to 1.0; (c) $k_l = 2$, $k_s = 1$, time from 0 to 0.3; (d) $k_l = 1.0$, $k_s = 0.01$, time from 0 to 0.25. All computations on a 400×400 grid.

shown in Fig. 21 as well as in Figs. 22b and 22c were allowed to grow in a larger domain this second tip-splitting event would occur at a later time.

The effect of surface tension on evolving interfaces is now demonstrated for unstable solidifying interfaces. The initial interface is as shown in Fig. 23. The initially flat interface is perturbed by imposing a four-wave sinusoidal perturbation of amplitude 0.1. The melt is initially undercooled to $\mathrm{St} = 0.8$. The boundaries of the domain of dimensions 4×8 units are insulated. Two cases are computed. For the case in Fig. 23a, $\sigma = 0.001$ and $\mu = 0$, and for that in Fig. 23b, $\sigma = 0.0002$ and $\mu = 0$. The role of the surface tension in setting the length scale of the instabilities is well demonstrated by the results. In the higher surface tension case, although the initial interface perturbation consisted of four wavelengths the final structure appears to tend toward the two outer fingers which grow at the expense of the two central fingers whose development has been suppressed by the latent heat released by the larger fingers. In contrast, by decreasing the surface tension by a factor of 5 in Fig. 23b, the

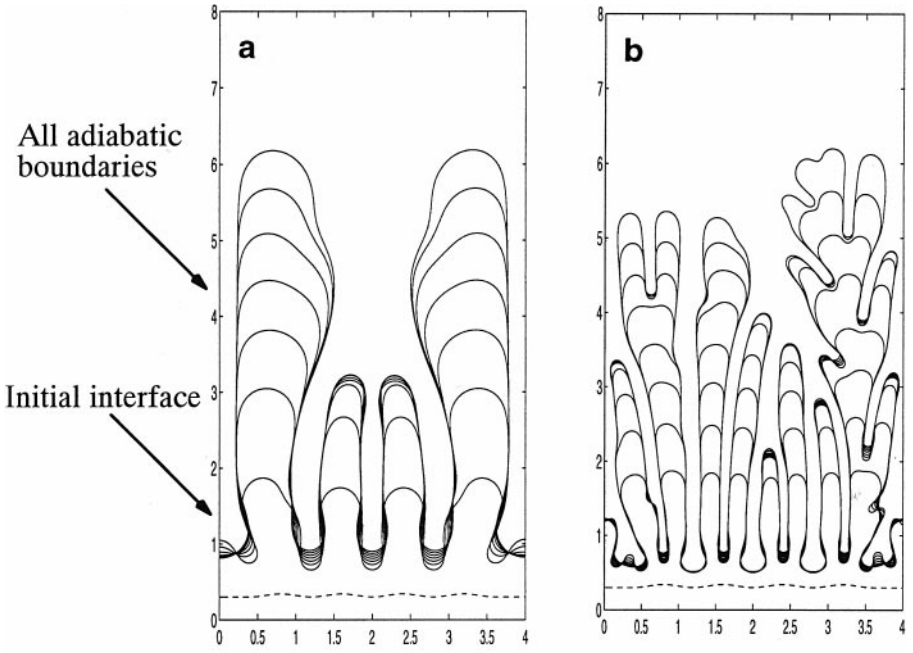

FIG. 23. Unstable solidification and finger formation (isotropic surface tension). Stefan number $= -0.5$. (a) Higher surface tension $\sigma = 1 \times 10^{-3}$. Interface shapes at equal intervals from $t = 0$ to 7. (b) Lower surface tension $\sigma = 2 \times 10^{-4}$. Interface shapes at equal intervals from $t = 0$ to 3.2.

initial four-wave perturbation is immediately converted into a finer scale instability. This selected fine spacing is maintained as the interface propagates by spacing readjustments. These readjustments result from the suppression of growth of some fingers and by repeated tip-splitting events which maintain the scale of the structure.

Next we examine the effects of anisotropy on the growth of interfaces. In Fig. 24 we compute the growth of an initially fourfold symmetric seed crystal in an undercooled melt

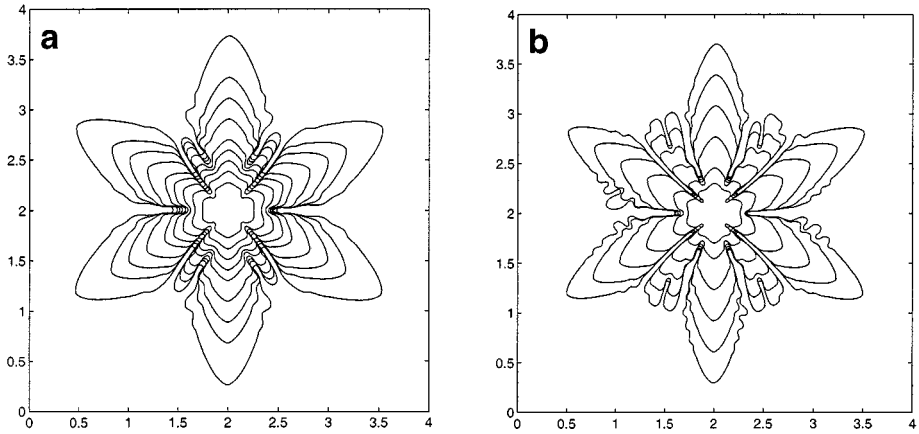

FIG. 24. Sixfold symmetric anisotropy in surface tension. Simulations using a 500×500 grid. St $= 0.8$. (a) Higher surface tension case. Surface tension parameter $\sigma = 0.002$, kinetic parameter $\mu = 0.002$, surface tension anisotropy $A_s = 0.4$, and kinetic anisotropy $A_k = 0.0$. t from 0 to 0.9. (b) Lower surface tension case. Surface tension parameter $\sigma = 0.001$, kinetic parameter $\mu = 0.001$, surface tension anisotropy $A_s = 0.4$, and kinetic anisotropy $A_k = 0.0$. t from 0 to 0.6.

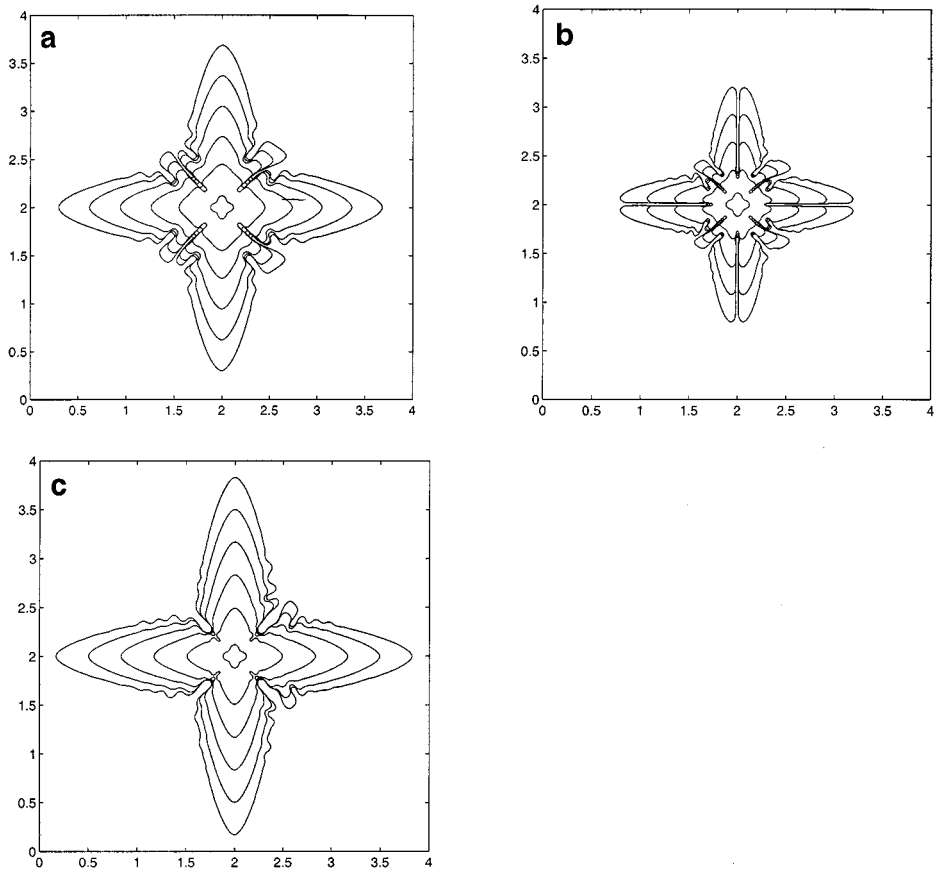

FIG. 25. Fourfold symmetric anisotropy in surface tension. St = 0.8. (a) Higher surface tension case. Surface tension parameter $\sigma = 0.002$, kinetic parameter $\mu = 0.0$, surface tension anisotropy $A_s = 0.4$, and kinetic anisotropy $A_k = 0.0$. $m = 4$. t from 0 to 0.05. 500×500 grid. (b) Lower surface tension case. Surface tension parameter $\sigma = 0.001$, kinetic parameter $\mu = 0.0$, surface tension anisotropy $A_s = 0.4$, and kinetic anisotropy $A_k = 0.0$. $m = 4$. t from 0 to 0.03. Simulations on an 800×800 grid. (c) Lower surface tension case with kinetic anisotropy. Surface tension parameter $\sigma = 0.001$, kinetic parameter $\mu = 0.001$, surface tension anisotropy $A_s = 0.0$, and kinetic anisotropy $A_k = 0.4$. $m = 4$. t from 0 to 0.05. 500×500 grid.

with a sixfold symmetric surface tension. The parameters used for the case in Fig. 24a are $\sigma = 0.002$, $\mu = 0.002$, $A_s = 0.4$, and $A_k = 0.0$. For case in Fig. 24b a lower surface tension and kinetic parameter was used; i.e., $\sigma = 0.001$ and $\mu = 0.001$. The undercooling was specified to be St = 0.8. Both calculations were performed on a 500×500 grid. The time step size was controlled based on the criterion in Eq. (33). The effect of the sixfold anisotropy is to promote growth in the preferred directions to yield a sixfold symmetric structure. The finer scales displayed by the more unstable lower surface tension case is consistent with the expected physics. Here again our results appear to differ in the details from those of JT96. Consistent with the observations made in connection with the test problem in Fig. 21, the present method appears to be less susceptible to small scale instabilities on the surface than that of JT96. The overall growth rate and pattern appear to agree with JT96 but the branching activity is less profuse in our case. A fourfold symmetric case is also computed, as shown in Fig. 25, by imposing fourfold symmetry in the surface tension. The parameters for the case in Fig. 25a are $\sigma = 0.002$, $\mu = 0.0$, $A_s = 0.4$, $A_k = 0.0$, and $m = 4$ and for that

in Fig. 25b, $\sigma = 0.001$, $\mu = 0.0$, $A_s = 0.4$, $A_k = 0.0$, and $m = 4$. For both cases $St = 0.8$. While the fourfold symmetry is reflected in the final shape for the higher surface tension case in Fig. 25a with four stably growing tips, the lower surface tension case seems to succumb to an early time tip split despite the imposed anisotropy. In the initial stage when the thermal boundary layer is still being formed the interface velocity is very high. It appears that for this case, in the initial stage the anisotropy is unable to completely stabilize the tip. In order to ensure that this behaviour is not grid-dependent we computed this case on a finer grid of 800×800 points. Note that after this initial tip split which leads to the formation of a groove in the tip of the dendrite in Fig. 25b, the tip propagates stably in the preferred growth directions. In Fig. 25c we impose the anisotropy in the kinetic anisotropy instead of the surface tension. This appears to stabilize the interface and prevent the early time split. This is perhaps because in the initial stage of rapid growth the kinetic anisotropy is perhaps more effective in providing tip stabilization than the surface tension anisotropy.

As a final demonstration, in Fig. 26 we show the formation of a dendrite from an initial fourfold symmetric seed placed eccentrically in an insulated box. The dimension of the box is 4×8 units. A 500×1000 uniform grid system was used for the computations. The initial seed is shown in the figure. The radius was given to be that in Eq. (57). The interface parameters chosen were $\sigma = 0.002$, $\mu = 0.0$, $A_s = 0.4$, $A_k = 0.0$, and $m = 4$. The undercooling was given to be $St = 0.8$. The interface shape is plotted at equal intervals of time from $t = 0$ to 0.07. The initial seed grows without hindrance from the insulated

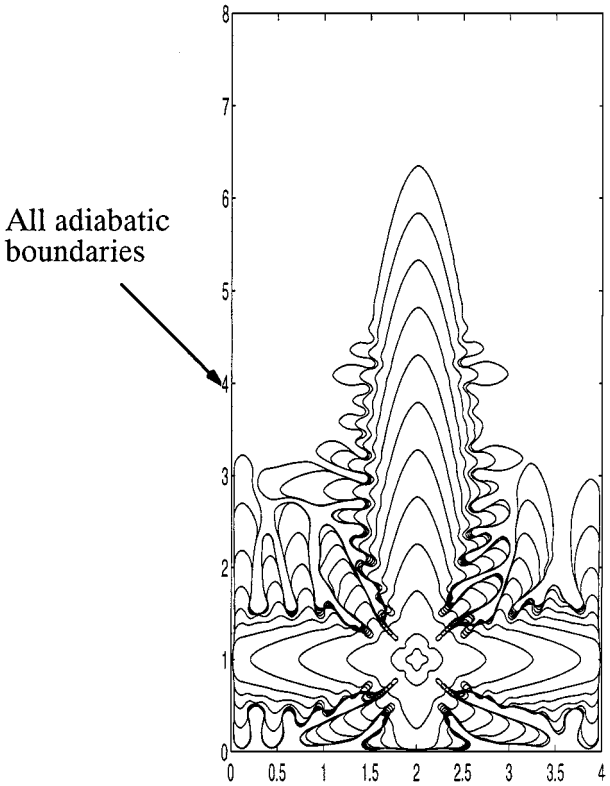

FIG. 26. Fourfold symmetric anisotropy in surface tension. $St = 0.8$. Simulations using a 500×1000 grid. Surface tension parameter $\sigma = 0.002$, kinetic parameter $\mu = 0.0$, surface tension anisotropy $A_s = 0.4$, and kinetic anisotropy $A_k = 0.0$. $m = 4$. t from 0 to 0.07.

boundaries for a short duration in a fourfold symmetric fashion. Three of the primary arms then feel the effects of the boundary and growth is inhibited. Thereafter, branches grow on these arms in the *y* direction where the melt undercooling persists. The primary arm is shown to grow stably with a parabolic tip and a uniform velocity. Intense sidebranching activity follows behind the stably growing tip. Asymmetry and branch competition are noticed in the development of the sidebranches on the main stem. Further coarsening effects can also be seen on the sidebranches on the horizontal primary branches where a final coarse structure in the secondary sidebranches seems to have emerged at the end of the calculation.

4. CONCLUDING REMARKS

In this report we have described a Cartesian grid solver for PDEs in the presence of embedded moving boundaries. We first presented an account of the development of a finite-difference field equation solver for scalar transport. The application of Dirichlet as well as Neumann boundary conditions is shown to be accurate to second order. The method has been applied to treating the computation of solid–liquid phase fronts with a sharp interface representation of the interface. This work therefore differs from the majority of Cartesian grid methods for the solution of moving boundary problems where the effects of the interface are distributed over a region proportional to the grid size. The region of spread is usually assigned to be proportional to the strength of the discontinuity at the interface. In the present work we show how localized sources at the interface as well as jumps in properties across it can be treated in a simple finite-difference framework without the necessity of smearing such discontinuities over the grid. At each stage of the present effort the accuracy of the method has been ascertained and it is shown that the field variable is computed to second-order accuracy. The boundary position, which in the case of the solidification problem depends on the gradient of the temperature field, is obtained to first-order accuracy. The method is stable due to the fully coupled way in which the interface and field variable are updated. The numerical stability criterion for time stepping then takes the form of a convective criterion for interface advance. It is shown that complex interface morphologies can be tracked using the method presented in this work. However, in the unstable growth of the front there is a disparity in the results from the present method, the immersed boundary method of Juric and Tryggvason (1996), and Chen *et al.* (1997). The errors incurred in the discretization in these methods are different and it is possible that the differences in solutions in the noise-sensitive unstable case are induced by such numerical errors. Our attempt to benchmark the calculations for unstable growth shows that further efforts to apply algorithms to well-defined benchmark problems are necessary. Based on the framework developed here for the diffusion and convection diffusion equation the Navier–Stokes equations can also be solved. With particular regard to solid–liquid interface instabilities, the inclusion of solute transport to simulate interface propagation in alloys is straightforward. Work in these directions is in progress.

ACKNOWLEDGMENTS

This work was performed with support from the Air Force Research Laboratory, Eglin Air Base, Florida. Dr. Kirk Vanden was the project monitor.

REFERENCES

1. R. Almgren, Variational algorithms and pattern formation in dendritic solidification, *J. Comput. Phys.* **106**, 337 (1993).

2. R. P. Beyer and R. J. LeVeque, Analysis of a one-dimensional model for the immersed boundary method, *SIAM J. Numer. Anal.* **29**, No. 2, 332 (1992).
3. J. U. Brackbill, D. B. Kothe, and C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* **100**, 335 (1992).
4. G. Caginalp, in *Applications of Field Theory to Statistical Mechanics*, edited by L. Garrido, Lecture Notes in Physics, No. 216 (Springer-Verlag, Berlin, 1984), p. 216.
5. H. Carslaw and J. Jaeger, *Conduction of Heat in Solids* (Clarendon Press, Oxford, 1959).
6. S. Chen, B. Merriman, S. Osher, and P. Smereka, A simple level set method for solving Stefan problems, *J. Comput. Phys.* **134** (1997).
7. A. J. Chorin, Hairpin removal in vortex interactions, *J. Comput. Phys.* **91**, 1 (1990).
8. J. Crank, *Free and Moving Boundary Problems* (Clarendon Press, Oxford, UK, 1984).
9. A. J. DeGregoria and L. W. Schwartz, A boundary integral method for two-phase displacement in Hele–Shaw cells, *J. Fluid Mech.* **164**, 383 (1986).
10. M. D. Dembo, D. C. Torney, K. Saxman, and D. A. Hammer, The reaction-limited kinetics of membrane-to-surface adhesion and detachment, *Proc. R. Soc. London B* **234**, 55 (1988).
11. C. Dong, R. Skalak, K.-L. P. Sung, G. W. Schmid-Schonbein, and S. Chien, Passive deformation analysis of human leukocytes, *J. Biomech. Eng.* **110**, 27 (1988).
12. L. J. Fauci and C. S. Peskin, A computational model of aquatic animal locomotion, *J. Comput. Phys.* **77**, 85 (1988).
13. J. Glimm, J. Grove, B. Lindquist, O. A. McBryan, and G. Tryggvason, The bifurcation of tracked scalar waves, *SIAM J. Sci. Statist. Comput.* **1**, 61 (1988).
14. R. Glowinski, T.-S. Pan, and J. Periaux, A fictitious domain method for Dirichlet problem and applications, *Comput. Methods Appl. Mech. Eng.* **111**, 283 (1994).
15. D. Goldstein, R. Handler, and L. Sirovich, Modeling a no-slip surface with an external force field, *J. Comput. Phys.* **105**, 354 (1993).
16. D. Goldstein, R. Handler, and L. Sirovich, Direct numerical simulation of turbulent flow over a modeled riblet covered surface, *J. Fluid Mech.* **302**, 333 (1995).
17. C. W. Hirt and B. D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, *J. Comput. Phys.* **39**, 201 (1981).
18. T. Y. Hou, J. S. Lowengrub, and M. J. Shelley, Removing the stiffness from interfacial flows with surface tension, *J. Comput. Phys.* **114**, 312 (1994).
19. T. Y. Hou, Z. Li, S. Osher, and H. Zhao, A hybrid method for moving interface problems with application to the Hele–Shaw flow, *J. Comput. Phys.* **134**, No. 2, 236 (1997).
20. V. Jayaraman, H. S. Udaykumar, and W. Shyy, Adaptive unstructured grid for three-dimensional interface representation, *Numer. Heat Transfer Part B* **32**, 247 (1997).
21. R. A. Johnson and D. M. Belk, Multigrid approach to overset grid comunication, *AIAA J.* **33**, 2305 (1995).
22. D. A. Jones, C. W. Smith, and L. V. McIntire, Effects of fluid shear stress on leukocyte adhesion to endothelial cells, in *Physiology and Pathophysiology of Leukocyte Adhesion*, Vol. 148 (Oxford Univ. Press, New York, 1995).
23. D. Juric and G. Tryggvason, A front tracking method for dendritic solidification, *J. Comput. Phys.* **123**, 127 (1996).
24. H.-C. Kan, H. S. Udaykumar, W. Shyy, and R. Tran-Son-Tay, Hydrodynamics of a compound drop with application to leukocyte modeling, *Phys. Fluids* **10**, No. 4, 760 (1998).
25. A. Karma and W.-J. Rappel, Numerical simulation of three-dimensional dendrite growth, *Phys. Rev. Lett.* **77**, No. 19, 4050 (1996).
26. D. A. Kessler, J. Koplik, and H. Levine, Pattern selection in fingered growth phenomena, *Adv. Phys.* **37**, 255 (1988).
27. R. Kobayashi, Modeling and numerical simulation of dendritic crystal growth, *Physica D* **63**, 410 (1993).
28. D. B. Kothe and R. C. Mjolsness, Ripple: A new model for incompressible flows with free surfaces, *AIAA J.* **30**, 2694 (1992).

29. J. S. Langer, Instabilities and pattern formation in crystal growth, *Rev. Mod. Phys.* **52**, No. 1, 1 (1980).
30. J. S. Langer, in *Directions in Condensed Matter Physics*, edited by G. Grinstein and G. Mazenko (World Scientific, Philadelphia, 1986), p. 164.
31. R. J. LeVeque and Z. Li, The immersed boundary method for elliptic equations with discontinuous coefficients and singular sources, *SIAM J. Numer. Anal.* **31**, No. 4, 1019 (1994).
32. J. E. Melton, F. Y. Enomoto, and M. J. Berger, 3D automatic grid generation for Euler flows, AIAA Paper 93-3386-CP (1993).
33. S. Osher and J. A. Sethian, Fronts propagating with curvature dependent speed: Algorithms based in Hamilton–Jacobi formulations, *J. Comput. Phys.* **79**, 12 (1988).
34. P. Pelce, *Dynamics of Curved Fronts* (Academic Press, New York, 1988).
35. R. B. Pember, J. B. Bell, and P. Colella, An adaptive Cartesian grid method for unsteady compressible flow in irregular regions, *J. Comput. Phys.* **120**, 278 (1995).
36. C. S. Peskin, Numerical analysis of blood flow in the heart, *J. Comput. Phys.* **25**, 220 (1977).
37. J. J. Quirk, An alternative to unstructured grids for computing gas dynamic flows around arbitrarily complex two-dimensional bodies, ICASE Report No. 92-7 (NASA Langley Research Center, Hampton, VA, 1992).
38. Y. Saito, G. Goldbeck-Wood, and H. Muller-Krumbhaar, Numerical simulation of dendrite growth, *Phys. Rev. A* **38**, No. 4, 2148 (1988).
39. R. Scardovelli and S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, *Annu. Rev. Fluid Mech.* **31**, 567–584 (1999).
40. A. Schmidt, Computation of three dimensional dendrites with finite elements, *J. Comput. Phys.* **125**, 293 (1996).
41. J. A. Sethian, *Level Set Methods: Evolving Interfaces in Geometry, Fluid Mechanics, Computer Vision, and Materials Science* (Cambridge Univ. Press, Cambridge, UK, 1996).
42. J. A. Sethian and J. Strain, Crystal growth and dendritic solidification, *J. Comput. Phys.* **98**, 231 (1992).
43. W. Shyy, *Computational Modelling for Fluid Flow and Interfacial Transport* (Elsevier, Amsterdam, The Netherlands, 1994).
44. W. Shyy, H. S. Udaykumar, M. M. Rao, and R. W. Smith, *Computational Fluid Dynamics with Moving Boundaries* (Hemisphere, Washington, DC, 1996).
45. R. W. Smith and W. Shyy, Computation of aerodynamic coefficients for a flexible membrane airfoil in turbulent flow: A comparison with classical theory, *Phys. Fluids* **8**, 3346 (1996).
46. J. Snyder and A. R. Woodbury, Interval methods for multi-point collisions between time-dependent curved surfaces, in *Comput. Graphics Proceedings, Annual Conference Series, SIGGRAPH 93* (1993), p. 321.
47. J. L. Steger, Thoughts on the Chimera grid method of simulation of three-dimensional viscous flow, in *Proceedings, Computational Fluid Dynamics Symposium on Aeropropulsion*, NASA CP-3078 (Natl. Aeronautics & Space Admin., Washington, DC, 1991), p. 1.
48. C. Tu and C. S. Peskin, Stability and instability in the computation of flows with moving immersed boundaries: A comparison of three methods, *SIAM J. Sci. Statist. Comput.* **13**, 1361 (1992).
49. H. S. Udaykumar and W. Shyy, A grid-supported marker particle scheme for interface tracking, *Numer. Heat Transfer B* **27**, 127 (1995a).
50. H. S. Udaykumar and W. Shyy, Simulation of morphological instabilities during solidification; Part I: Conduction and capillarity effects, *Int. J. Heat Mass Transfer* **38**, 2057 (1995b).
51. H. S. Udaykumar, W. Shyy, and M. M. Rao, *ELAFINT*—A mixed Eulerian–Lagrangian method for fluid flows with complex and moving boundaries, *Int. J. Numer. Methods Fluids* **22**, 691 (1996).
52. H. S. Udaykumar, H.-C. Kan, W. Shyy, and R. Tran-Son-Tay, Multiphase dynamics in arbitrary geometries on fixed Cartesian grids, *J. Comput. Phys.* **137**, 366 (1997).
53. S. O. Unverdi and G. Tryggvason, A front tracking method for viscous, incompressible, multifluid flows, *J. Comput. Phys.* **100**, 25 (1992).
54. V. Venkatakrishnan, Perspectives on unstructured grid flow solvers, *AIAA J.* **34**, 533 (1996).
55. V. R. Voller and C. Prakash, A fixed grid numerical modeling methodology for convection-diffusion mushy region phase-change problems, *Int. J. Heat Mass Transfer* **30**, 1709 (1987).

56. A. A. Wheeler, W. J. Boettinger, and G. B. McFadden, Phase field model for isothermal phase transitions in binary alloys, *Phys. Rev. A* **45**, 7424 (1992).
57. D. P. Young, R. G. Melvin, M. B. Bieterman, F. T. Johnson, S. S. Samant, and J. E. Bussoletti, A locally refined rectangular grid finite element method: Application to computational fluid dynamics and computational physics, *J. Comput. Phys.* **92**, 1 (1992).
58. W. J. Goirier and K. G. Powell, "Solution-adaptive Cartesian cell approach for viscous and inviscid flows," *AIAA J.* **34**, 938–945 (1996).
59. H. Zhang, L. L. Zheng, V. Prasad, and T. Y. Hou, A curvilinear level set formulation for highly deformable free surface problems with application to solidification, *Numer. Heat Transfer Part B* **34**, 1 (1998).