Available online at www.sciencedirect.com

SCIENCE DIRECT®

Journal of Computational Physics 194 (2004) 677–696

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

Numerical simulation of dendritic solidification with convection: Three-dimensional flow

Nabeel Al-Rawahi [a], Gretar Tryggvason [b,*]

[a] *Department of Mechanical and Industrial Engineering, Sultan Qaboos University, Al-khod, 123, Oman*
[b] *Department of Mechanical Engineering, Worcester Polytechnic Institute, Worcester, MA 01609, USA*

Received 11 March 2003; received in revised form 25 September 2003; accepted 25 September 2003

Abstract

A numerical method for the simulation of the effect of melt flow on the three-dimensional growth of a dendrite is described. The method is an extension of the technique for two-dimensional flow described in Al-Rawahi and Tryggvason [J. Comput. Phys. 180 (2002) 471] and is based on the explicit tracking of connected marker points that describe the liquid–solid interface. An explicit projection method is used to solve the energy and the Navier–Stokes equations on a regular stationary grid and the solidified region is represented by setting the velocities in the solid phase to zero. The latent heat released during solidification is calculated using the normal temperature gradient near the interface. The method is validated by a comparison with an exact solution for a Stefan problem and a grid refinement study. The simulations show that the speed of a dendrite arm growing into the flow is increased due to an increase in the temperature gradient on the upstream side and the formation of side branches is promoted, as in two-dimensions. The effect of the flow on the growth of dendrite arms growing in the downstream direction is smaller than in two-dimensions, due to a smaller wake.
© 2003 Elsevier Inc. All rights reserved.

1. Introduction

The relation between the properties of metals and their microstructure is one of the fundamental problems in material science. The identification and classification of microstructures is mostly based on observations and the dependency of the microstructure on the process parameters is still poorly understood. While dendritic microstructures generally form because of constitutional undercooling in alloys, pure materials can form transient dendritic microstructure during solidification. It is well known that fluid flow can have significant effect on the morphology of the microstructure and a number of experimentalists have quantified this effect (see Glicksman, [2], for example). In this paper we describe a numerical method to simulate the effect of flow on the growth of transient dendritic microstructures in pure materials.

* Corresponding author. Tel.: +1-508-831-5759; fax: +1-508-831-5680.
E-mail address: gretar@WPI.EDU (G. Tryggvason).

doi:10.1016/j.jcp.2003.09.020

Theoretical solutions for dendritic growth are limited to very simple situations and numerical simulations are playing an increasing role in studies of dendrites. Simulations of the growth of two-dimensional dendrites without flow are now common and a number of simulations of the growth of three-dimensional dendrites in the absence of fluid motion are available. Detailed review of simulations of two-dimensional systems can be found in Nabeel and Tryggvason [1] where two-dimensional simulations of the effect of flow are discussed, and in Zhao et al. [3] who present two-dimensional simulations of binary alloys. The first simulation of a three-dimensional dendrite was published by Kobayashi [4], who used a phase field method and who studied the effect of surface tension anisotropy on the morphology of the dendrite. Another phase field method was presented by Karma and Rappel [5,6], who compared their results to the predictions of solvability theory. Schmidt [7] used a finite element method and two independent grids for the melt and the solid. A combined phase field/Monte Carlo method was developed by Plapp and Karma [8] to simulate dendritic solidification in the low undercooling limit and simulations using level-sets to track the interface are presented by Gibou et al. [9,10].

Simulations that include fluid flow are limited to a few simulations of two- and fully three-dimensional dendrites. Phase field methods have been extended to simulate the growth of two-dimensional dendrites with melt flow by Tonhardt and Amberg [11] and Beckerman et al. [12]. Tonhardt and Amberg studied the effect of shear flow on dendritic growth from a solid wall by modeling the solid as a fluid that was 100 times more viscous than the melt. Beckerman et al. examined the growth of a single dendrite in a uniform flow and held the solid in place by a force field added to the momentum equations. Finite element simulations were presented by Bansch and Schmidt [13] who modeled the solid and the fluid domain with separate grids and explicitly enforced the no-slip boundary condition at the interface. A method using a regular grid that is modified locally to align with the interface has been introduced by Udaykumar et al. [14]. Juric [15] used a front tracking method to simulate the effect of shear flow on dendritic growth. He calculated the heat source by an iterative scheme and modeled the solid as a fluid with a viscosity that was 100 times that of the liquid. Al-Rawahi and Tryggvason [1] presented an improved front tracking method in which the heat source is found directly from the temperature gradient near the interface and the no-slip boundary condition is enforced by directly setting the velocity in the solid to zero. The studies described above have all been limited to two-dimensional systems. Methods for the simulations of fully three-dimensional growth of a dendrite in the presence of melt flow, using the approach of Beckerman et al. [12], were recently presented by Jeong et al. [16,17] and Lu et al. [18] (see also [19]). Jeong et al. pointed out that the flow structure behind a fully three-dimensional dendrite is quite different than behind a two-dimensional one.

In this paper we extend the front tracking method developed by Al-Rawahi and Tryggvason [1] – here referred to as part I – to fully three-dimensional flows. After reviewing briefly the governing equations we describe the numerical implementation, focusing on those aspects that are different for three-dimensional flow. Then we show a few examples of simulations done using the new method.

2. Formulation

The mathematical model solved here is essentially the same as in Part I. The problem setup is sketched in Fig. 1. Undercooled liquid enters through the left boundary of a cubical domain and flows past a dendrite growing in the center. As the liquid solidifies and the dendrite grows, the geometry of the solid/liquid boundary changes. The whole domain is resolved by a single fixed grid and one set of equations is used for both the liquid and the solid. The phase boundary is treated as an imbedded interface by adding the appropriate source terms as delta functions to the conservation laws. The temperature is found by solving the energy equation

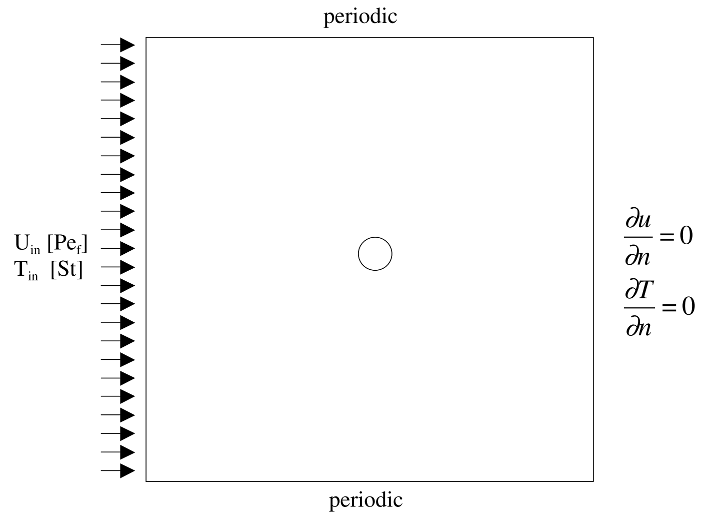
periodic
periodic
U_{in} [Pe_f]
T_{in} [St]
$\frac{\partial u}{\partial n} = 0$
$\frac{\partial T}{\partial n} = 0$

Fig. 1. The initial setup used for the simulations presented in this paper. A solid seed is located at the center of the cubical domain and uniform undercooled liquid enter through the left boundary with a velocity equal to the Peclet number. Weak conditions are imposed on the outlet flow and periodic boundary conditions are used for the front and back, as well as the top and bottom boundaries.

$$\frac{\partial \rho c_p T}{\partial t} + \nabla \cdot (\rho c_p T \mathbf{u}) = \nabla \cdot k \nabla T + \int_F \dot{q} \delta(\mathbf{x} - \mathbf{x}_\mathrm{f}) \mathrm{d}a, \tag{1}$$

where we have assumed that the fluid is incompressible and that viscous heating can be neglected. $\mathbf{u}$ is the fluid velocity (equal to zero in the solid), ρ and c_p are the density and heat capacity, respectively, k is the thermal conductivity and $\mathbf{x}_\mathrm{f}$ is the front location. The integral term is the heat liberated during solidification and the source strength $\dot{q}$ is related to the normal velocity of the phase boundary V_n and the latent heat L by

$$\dot{q} = \rho L V_\mathrm{n}. \tag{2}$$

The motion of the phase boundary is found by integrating

$$\frac{\mathrm{d}\mathbf{x}_\mathrm{f}}{\mathrm{d}t} = V_\mathrm{n} \mathbf{n}, \tag{3}$$

where $\mathbf{n}$ is the normal vector to the phase boundary. To find the heat source and thus the normal velocity, we integrate the energy equation across the phase boundary. The result is that the heat released by the motion of the front must be balanced by heat conduction from the front:

$$\dot{q} = k_\mathrm{solid} \left.\frac{\partial T}{\partial \mathbf{n}}\right)_\mathrm{solid} - k_\mathrm{liquid} \left.\frac{\partial T}{\partial \mathbf{n}}\right)_\mathrm{liquid}. \tag{4}$$

The interface temperature satisfies the Gibbs–Thompson relation

$$T_\mathrm{f} = T_\mathrm{m}\left(1 - \frac{\gamma}{\rho L}\kappa\right) - \frac{V_\mathrm{n}}{\eta}. \tag{5}$$

Here, γ is the surface tension, η is the kinetic mobility and κ is the curvature. The surface tension and the kinetic mobility are generally anisotropic and we use

$$\begin{aligned}\gamma &= \gamma_0(1 - A_{\mathrm{s}}(4(n_1^4 + n_2^4 + n_3^4) - 3)),\\ \frac{1}{\eta} &= \frac{1}{\eta_0}(1 - A_{\mathrm{k}}(4(n_1^4 + n_2^4 + n_3^4) - 3)).\end{aligned} \tag{6}$$

In Eq. (6), A_{s} and A_{k} quantify the anisotropy of the surface tension and the inverse of the kinetic mobility, respectively, n_1, n_2 and n_3 are the components of the normal unit vector along the axes of growth, calculated at each point on the interface. This form of the anisotropy sets six preferred directions of growth, resulting in a six folds dendrite. For the simulations presented here the axes of growth are in the coordinate directions, so $(n_1, n_2, n_3) = (n_x, n_y, n_z)$. This form is similar to the one used by Schmidt [7] and Muschol et al. [20]. Other expressions have been used for the anisotropy, see [16], but as discussed by Juric and Tryggvason [21] the effects are generally similar.

The fluid motion is governed by the Navier–Stokes equations:

$$\frac{\partial \mathbf{u}_{\mathrm{liquid}}}{\partial t} + \nabla \cdot \mathbf{u}_{\mathrm{liquid}}\mathbf{u}_{\mathrm{liquid}} = -\nabla p + v\nabla^2 \mathbf{u}_{\mathrm{liquid}} \tag{7}$$

and the velocity in the solid is zero

$$\mathbf{u}_{\mathrm{solid}} = 0. \tag{8}$$

Here, v is the kinematic viscosity and p is the pressure. The fluid is assumed to be incompressible and the density of the solid and the liquid are taken to be the same. Volume contraction and expansion, as well as fluid convection due to buoyancy are thereby neglected. Since density is constant, we have redefined pressure as the pressure divided by the density. Since the governing equations are solved on a single grid, we define a velocity field everywhere in the domain by

$$\mathbf{u} = \phi \mathbf{u}_{\mathrm{liquid}}. \tag{9}$$

$\phi(x)$ is an indicator function defined by

$$\phi = \begin{cases} 0 & \text{in the solid,} \\ 1 & \text{in the liquid,} \end{cases} \tag{10}$$

and varies smoothly between zero and one in the interface region. The mass conservation equation for the whole domain reduces to:

$$\nabla \cdot \mathbf{u} = 0. \tag{11}$$

The thermal conductivity and the specific heat of each phase are assumed to be constant in each phase, but can be different in the solid and the liquid.

The nondimensional parameters governing this problem have been discussed in Part I and the definitions of the various nondimensional numbers are given in Table 1, along with the scaling for time and length.

3. Numerical implementation

The energy equation and the momentum equation are solved on a single regular stationary grid that covers both the liquid and the solid region. The phase boundary is represented by moving marker points, connected by elements to form a two-dimensional front that cuts through the three-dimensional stationary mesh. The front is used to advect the discontinuous material property fields and to calculate the heat source.

Table 1
Nondimensional parameters. Here, c_p is the heat capacity, ΔT is the undercooling, L the latent heat, k the heat conductivity, U the inflow velocity, μ the fluid viscosity, ρ the density, γ_0 the surface tension and T_M the melt temperature for a flat phase boundary

Nondimensionalization:	
Length: $d_0 = \frac{T_m \gamma_0 c_p}{\rho L^2}$; Time: $\tau = d_0^2 \frac{\rho c_p}{k}$; Velocity: $u_0 = \frac{d_0}{\tau} = \frac{k}{d_0 \rho c_p}$	
Nondimensional numbers:	
Stefan number $St = \frac{c_p \Delta T}{L}$	Prandtl number $Pr = \frac{k}{c_p \mu}$
Peclet number $Pe = \frac{kUd_0}{c_p}$	Stability parameter: $\sigma^* = \frac{2}{V_{\text{tip}} R_{\text{tip}}^2} \frac{kd_0^2}{\rho c_p}$

3.1. Solving the conservation equations

To discretize the energy equation and the momentum equations we use a conservative, second order, centered difference scheme for the spatial variables. For the simulations presented here we have used a first order explicit time integration method. As discussed in Part I, it is straightforward to extend the method to second order, using a predictor-corrector scheme. For details of the discretization, see Part I, as well as Tryggvason et al. [22]. At the inflow boundary (left) of the computational domain the velocity and a temperature equal to the undercooling temperature are specified, the top and bottom boundaries are periodic, and the fluid is allowed to flow freely out through the right boundary by putting the gradient of temperature and velocity to zero.

The main challenge in solving one momentum equation on a single grid resolving both the fluid and the solid is the enforcement of a zero velocity in the solid. As discussed in Part I, several authors have represented solids on fixed grids by finding the forces needed to hold them in place. Generally, it is necessary to use an iterative method to find the force. However, it has recently become clear that the computations can be simplified with minimal loss of accuracy (see Fadlun et al. [23], for example) by simply setting the velocity in the solid to zero (if the solid is stationary). This can be accomplished by a slight modification of the standard projection scheme. First we split the Navier–Stokes equations in the usual way; calculate the unprojected velocity, $\mathbf{u}^{\text{up}}$, set it equal to zero in the solid, and solve for the pressure. Then we add the pressure gradient, multiplied by ϕ to the unprojected velocity:

$$\begin{aligned} \mathbf{u}^{\text{up}} &= \phi(\mathbf{u}^n + \Delta t \mathbf{A}(\mathbf{u}^n)), \\ \nabla^2 P &= \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^{\text{up}}, \\ \mathbf{u}^{n+1} &= \mathbf{u}^{\text{up}} - \Delta t \phi \nabla P. \end{aligned} \tag{12}$$

Here, $\mathbf{A}$ is the discrete advection and diffusion terms. We have also found that instead of using the index function ϕ directly, using ϕ^2 helps keep the phase boundary sharper. This is similar to the use of non-symmetric weighting by Beckermann et al. [12]. A test of how well this approach works for a well-defined geometry is presented in Section 4.

The index function ϕ is constructed by distributing the jump in ϕ onto the fixed grid and integrating as described in Part I. Since we use explicit time integration for the energy and momentum equations, as well as for advecting the front, the size of the time step is limited by the criteria discussed in Part I. For the parameters used here, the most severe limitation comes from the diffusion part of the energy or the momentum equation.

3.2. The front

The connected marker particles – the front – are used to advect the discontinuous material property fields and to calculate the heat released during the solidification. Although the basic numerical implementation of the front is the same as in Part I, there are a few differences since the three-dimensional front is a surface instead of a curve. In the present simulations the front is represented by moving marker points, connected by triangular elements to form a two-dimensional front that lies on the three-dimensional stationary mesh. Fig. 2 shows the front structure and the different quantities stored for the front points and elements. Each element has pointers to the corner points and to the three adjacent elements. For each point its coordinates are stored. In addition, the elements and the points are logically connected by a linked list that identifies the previous and next element and points. The order of points and elements in the linked list is arbitrary. In addition to facilitate looping over the front points and elements, the linked list data structure simplifies the adding and deletion of points and elements as the front grows and deforms.

To compute the temperature at the center of each element (Eq. (5)) it is necessary to know the local mean curvature. To find the mean curvature for each element we first compute the total force on each element due to surface tension

$$\delta \mathbf{F} = \sigma \int_{\delta A} \kappa \mathbf{n} \, \mathrm{d}A. \tag{13}$$

Since the mean curvature of a surface can be written as

$$\kappa \mathbf{n} = (\mathbf{n} \times \nabla) \times \mathbf{n}, \tag{14}$$

Eq. (13) can be written as

$$\delta \mathbf{F} = \sigma \int_{\delta A} \kappa \mathbf{n} \, \mathrm{d}A = \sigma \int_{\delta A} (\mathbf{n} \times \nabla) \times \mathbf{n} \, \mathrm{d}A = \sigma \oint_{s} \mathbf{t} \times \mathbf{n} \, \mathrm{d}s. \tag{15}$$

Here, $\mathbf{t}$ is the vector tangent to the edge of the element, $\mathbf{n}$ is the normal vector to the element, and S is the contour bounding an element. The cross product is a vector that lies in the surface and is normal to the edge of the element. The product of the surface tension and this vector gives the pull on the element edge and the net force is obtained by integrating it around the perimeter of the element. To ensure that the total force is conserved, the vector obtained by the cross product for each edge is averaged between the two elements that

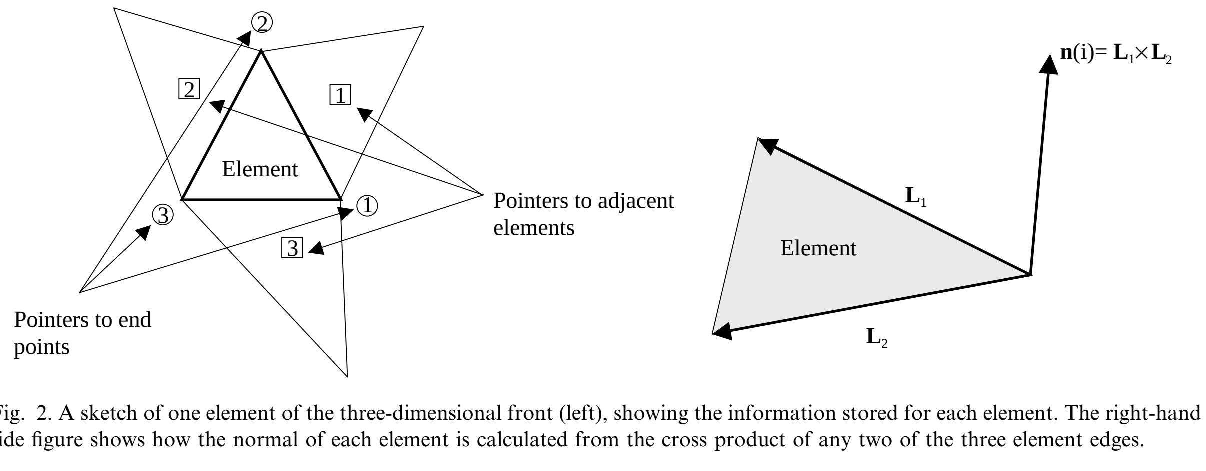

Fig. 2. A sketch of one element of the three-dimensional front (left), showing the information stored for each element. The right-hand side figure shows how the normal of each element is calculated from the cross product of any two of the three element edges.

share the edge. When the surface tension is constant, the net force is normal to the element. The net force on each element is also equal to the force calculated using the average mean curvature of each element:

$$\delta \mathbf{F} = \sigma\kappa\,\delta A\,\mathbf{n}. \tag{16}$$

Here, δA is the area of the element. Equating the force computed by Eqs. (15) and (16) we get

$$\kappa = \frac{1}{\delta A}\oint_s \mathbf{t}\times\mathbf{n}\,\mathrm{d}s\cdot\mathbf{n}, \tag{17}$$

where we have taken the dot product of Eq. (15) to get the normal component. This definition of the mean curvature (where the limit of $\delta A \to 0$, is taken) can also be found in Wetherburn [24]. The unit normal vector for each element is found by taking the cross product of any two edges of an element as shown in Fig. 2, scaled to unit length, and the area of each element is found as half the absolute value of the cross product of the edges. Once the curvature has been found, it is used to compute the temperature at the center of each front element using Eq. (5).

To determine the velocity of the front, we first need to find the heat source, using Eq. (4). We first find the heat source at the center of each element. To find the temperature gradients on either side of the front we follow Udaykumar et al. [14] and draw a normal line (or a "probe") from the element centers to points at $\omega * h$ away from the front:

$$\begin{aligned} x^{\pm} &= x_1 \pm \omega h n_x,\\ y^{\pm} &= y_1 \pm \omega h n_y,\\ z^{\pm} &= z_1 \pm \omega h n_z, \end{aligned} \tag{18}$$

where the plus (+) sign applies to the liquid side, and the minus (−) sign applies to the solid side, and n_x, n_y and n_z are the components of the unit vector normal to the element center. Here, h is the grid spacing and ω is an adjustable parameter. $\omega = 1.2$ was found to give the highest accuracy as discussed in Part I. The temperatures at $(x^{\pm}, y^{\pm}, z^{\pm})$ are then found from the temperatures of the surrounding grid points, T_{ijk}, using interpolation:

$$T_{\pm} = \sum_{ijk} T_{ijk} F_{ijk}\left(\mathbf{x}^{\pm}\right). \tag{19}$$

Here the weighing parameter $F_{ijk}(\mathbf{x}^{\pm})$ is given by

$$F_{ijk}\left(\mathbf{x}^{\pm}\right) = d(x^{\pm} - \mathrm{i}h)d(y^{\pm} - \mathrm{j}h)d(z^{\pm} - \mathrm{k}h), \tag{20}$$

where, $d(r)$ is a one-dimensional volume-weighing function

$$d(r) = \begin{cases} (h - |r|)/h, & |r| < h,\\ 0, & |r| \geqslant h. \end{cases} \tag{21}$$

Using the temperatures T_+ and T_- from Eq. (19) the heat source is found by

$$\dot{q} = \frac{1}{h}\left(k_{\mathrm{s}}(T_- - T_{\mathrm{f}}) - k_{\mathrm{l}}(T_{\mathrm{f}} - T_+)\right), \tag{22}$$

where, T_{f} is the temperature at the front, given by Eq. (5) if the inverse of the kinetic mobility equals to zero. If the inverse of the kinetic mobility does not equal to zero then T_{f}, V_{n} and $\dot{q}$ must be found by solving Eqs. (2), (5) and (22) simultaneously.

The heat source found by Eq. (22) is distributed onto the fixed grid by

$$\dot{q}_{ijk} = \sum_{\mathrm{el}} \dot{q}_{\mathrm{el}} F_{ijk}(\mathbf{x}_{\mathrm{el}}) \delta A_{\mathrm{el}} / h^3, \tag{23}$$

where $\mathbf{x}_{\mathrm{el}}$ is the coordinate of the element center, the weighing parameter F_{ijk} is defined by Eq. (20) and δA_{el} is the area of each front element. For the distribution of the heat source, the weighing parameter F_{ijk} is calculated using Peskin's [25] cosine weighting function, defined by

$$d(r) = \begin{cases} (1/4h)(1 + \cos(\pi r/2h)), & |r| < 2h, \\ 0, & |r| \geqslant 2h. \end{cases} \tag{24}$$

Once the heat source is found, the locations of the front points can be updated by integrating Eq. (3) using the same time integration scheme employed for the conservation equations. We must, however, first find the heat source and the normal vector, which are known at the element centers, at the front points. To do this we set up vectors containing the source strength and the associated area of each front point and initialize those vectors to zero. Then we loop over all the elements and add one third of the total source strength (the source strength per unit area times the area of each element) to the source strength at the corner points of each element. We also add one third of the element area to the area associated with the point and after the summing is done the total source strength at each point is divided by the total area associated with the point to give the source strength per unit area. For the normal vectors we simply add the element normal vectors of the elements connected to each point and divide by the number of elements. We have experimented with more elaborate averaging for the normal vectors but find essentially no effect on the results.

As the front topology changes, points and elements are added and deleted to keep the front smooth. As for the front in two-dimensions (Part I), points and elements are added or deleted when the length of any element edge exceeds a predefined maximum or minimum length. Here, we use a maximum length of h, which is the grid spacing of the fixed grid, and a minimum length of $h/3$. The front restructuring in three-dimensions is more involved than in two-dimensions and we must ensure that the addition and deletion of points and elements does not lead to poorly shaped elements or unacceptable connections. For a more detailed discussion of how the front is maintained, see Tryggvason et al. [22]. For more discussion about the use of the "one-field" formulation for computations, including extensive validation and grid refinement studies see Juric and Tryggvason [21] and Al-Rawahi and Tryggvason [1].

3.3. Parallelization

To make it possible to compute problems requiring fine grids and to improve performance, we have implemented the code on parallel computers, using the message-passing interface (MPI) library [26]. Fig. 3 illustrates how the computational tasks are distributed among the processors used. For the conservation equations we use a simple domain decomposition approach where the computational domain is divided into even-sized sub-domains that are assigned to separate processors. The front computations are assigned to one separate processor. This contrast to the approach used by Bunner [27] for his simulations of bubbly flows, where the front computations were divided between the processors used for the solution of the conservation equations. In that case, however, the front consisted of well-defined separate identities (the bubbles) whereas here the front is one continuous structure. The grid processors are also responsible for computing the heat sources for each element that lies in their domain. During each time step the front processor sends front information to the grid processors that in turn send the calculated heat sources of the elements lying in their domain to the front processor. The front processor receives the heat sources from each grid processor individually. Two methods were tested for sending information from the front processor to the grid processors. In the first approach the front processor sent information to each grid processor only for the elements lying in its domain. This requires individual messages from the front

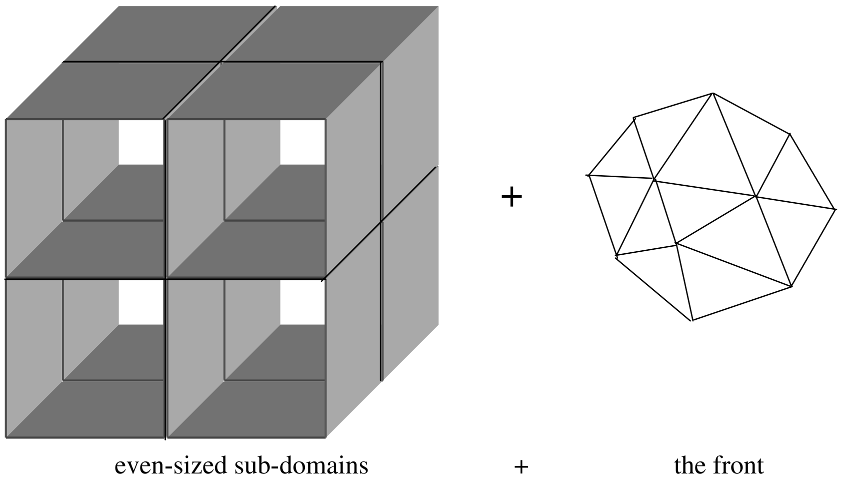

Fig. 3. Distribution of the computational load to the parallel processors. The implemented parallel computing strategy is to divide the whole domain into even-sized sub-domains and assign each sub-domain to a processor. A separate processor is assigned to the front. The total number of processors illustrated in the figure is $2 \times 2 \times 2 + 1 = 9$.

processor to each grid processor. In the second method the front processor broadcasted the entire front to all the grid processors in one message. Both methods were tested and it was found that the former method required more CPU time, especially when the front is very large. We have therefore used the second approach in most of our simulations. The grid processors also exchange boundary information for the temperature, velocities, and pressure. The pressure equation was solved by a parallelized multigrid code developed by Bunner [27], and only a few modifications were needed to accommodate the boundary conditions used in our code. More information about the multigrid code can be found in [27].

The simulations were done using the IBM SP2 machines at the Center for Parallel Computations (CPC) at The University of Michigan. The speedup (defined as the time needed to solve the problem using a single processor divided by the time needed using many processors) for a 128^3 grid and 1, 2, 4, 8, 16 and 32 grid processors is shown in Fig. 4. As the number of grid processors increases the domain computed by each processor becomes smaller, but the time consumed in exchanging boundary information increases. This is heavily influenced by the use of the multigrid method since it requires more exchange of boundary information then other iterative methods. The time consumed in advecting the velocity, including solving the pressure equation and finding the indicator function on the grid varies between 30% to 60% of the total time, depending on how many processors are used. The front processor is generally lightly loaded. However, as the front grows the time needed for the restructuring of the front increases and can consume more than 55% of the CPU time of the front processor.

3.4. The algorithm

Since the algorithm requires several steps, we list the various steps below in the order that they are carried out in the actual code. The simulation starts with a given flow field and an initial temperature field set equal to $T(x, y, z) = \phi St$. The integration of the governing equations is then carried out by the following algorithm:

1. Calculate the unit normal vector, the curvature, the surface tension anisotropy and the kinetic mobility anisotropy at the center of each front element.

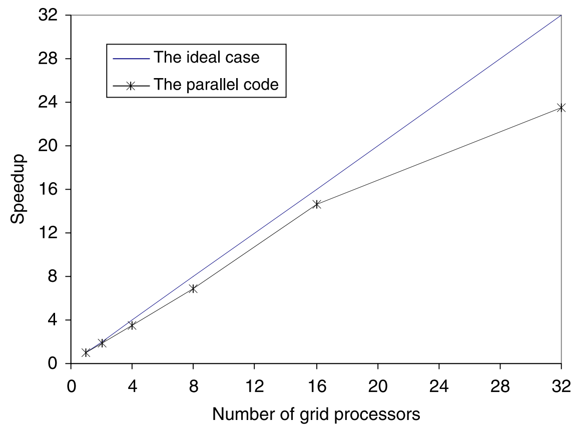

Fig. 4. The speedup (the time needed to solve the problem using a single processor divided by the time needed using many processors) versus the number of grid processors for the three-dimensional code. The simulations were carried out using a 128^3 grid for 100 time steps.

2. If the inverse of the kinetic mobility effect is zero, find the temperature at the center of each front element using Eq. (5). Otherwise, find the temperatures using Eqs. (2), (5) and (22).
3. Find the heat source at the center of each element using Eq. (22) and distribute the total heat sources for each element to its corner points.
4. Distribute the heat source to the fixed grid using Eq. (23).
5. Integrate the temperature field in time using Eq. (1).
6. Advance the front points using Eq. (3).
7. Calculate the new indicator function ϕ by distributing the jump in ϕ to the grid and integrate as described in Part I.
8. Advance the flow field by integrating the Navier–Stokes equation (7) in time, as described by Eq. (12), and return to step 1 to start the next time step.

4. Validation

To validate and check the code, we computed the solidification of a sphere and compared the computed results to analytical solutions for an spherically symmetric Stefan problem. The flow solver was validated by comparison of numerical results for flow through a square array of stationary spheres with analytical solutions for Stokes flow.

Numerical results from simulations of a three-dimensional Stefan problem are compared to the exact solution derived by Paterson [28] for the solidification of a sphere with a heat sink at the center. The strength of the heat sink increases with time as $Q = qt^{0.5}$, where t is time and q is a constant. The temperature in the liquid phase is given by

$$T(r,t) = St\left[1 - \frac{\left(\frac{\alpha_l}{\alpha_s}\right)^{0.5}\frac{t^{0.5}}{r}\exp\left(\frac{-r^2}{4t}\frac{\alpha_s}{\alpha_l}\right) - \frac{\pi^{0.5}}{2}\operatorname{erfc}\left(\frac{r}{2t^{0.5}}\left(\frac{\alpha_s}{\alpha_l}\right)^{0.5}\right)}{\left(\frac{\alpha_l}{\alpha_s}\right)^{0.5}\frac{1}{2\lambda}\exp\left(-\lambda^2\frac{\alpha_s}{\alpha_l}\right) - \frac{\pi^{0.5}}{2}\operatorname{erfc}\left(\lambda\left(\frac{\alpha_s}{\alpha_l}\right)^{0.5}\right)}\right] \tag{25}$$

and the temperature in the solid phase is given by

$$T(r,t) = \frac{q}{4\pi}\left[\frac{t^{0.5}}{r}\exp\left(\frac{-r^2}{4t}\right) - \frac{1}{2\lambda}\exp(-\lambda^2) + \frac{\pi^{0.5}}{2}\left(\mathrm{erfc}\left(\frac{r}{2t^{0.5}}\right) - \mathrm{erfc}(\lambda)\right)\right]. \tag{26}$$

The radius of the sphere as a function of time is

$$R_\mathrm{f}(t) = 2\lambda t^{0.5}, \tag{27}$$

where λ is the root of

$$\frac{q}{4\pi}\exp(-\lambda^2) = -4\lambda^3 - \frac{2\lambda\, St\, \frac{k_\mathrm{l}}{k_\mathrm{s}}}{1 - \left(\frac{\alpha_\mathrm{s}}{\alpha_\mathrm{l}}\right)^{0.5}\lambda\pi^{0.5}\exp\left(\lambda^2\frac{\alpha_\mathrm{s}}{\alpha_\mathrm{l}}\right)\mathrm{erfc}\left(\lambda\left(\frac{\alpha_\mathrm{s}}{\alpha_\mathrm{l}}\right)^{0.5}\right)}. \tag{28}$$

Here $\alpha = k/\rho c_p$ and the subscripts 's' and 'l' denote the solid and the liquid, respectively. The simulations were carried out in 2^3 domain using three grid resolutions of 30^3, 60^3 and 120^3. The heat sink was modeled in the simulations by putting the center of the sphere in the center of a mesh cell and keeping the temperature of the eight corner points equal to the analytical temperature. Fig. 5(a) shows the numerical and the analytical results for the radius of the sphere as function of time when $St = 0.5$, $q = -20$, and the material properties in the melt and the solid are equal (for those conditions, $\lambda = 0.4518075$). The numerical results are in good agreement with the analytical results and the computed results converge to the analytical solution as the grid resolution is increased. Fig. 5(b) and (c) shows the numerical and the analytical temperature profiles for a line passing through the center of the sphere at times 0.0276, 0.2976, 0.5676 and 0.8376, increasing from left to right, using the 120^3 grid. The numerical results accurately predict the analytical temperature profiles.

To evaluate the flow solver and to assess how accurately the flow over a solid object is captured on a fixed grid the Stokes flow over a simple cubic array of spheres was simulated and the results compared to the analytical solution of Sangani and Acrivos [29]. The simulations were done in a fully periodic cube and the flow was driven by an imposed pressure gradient. Fig. 6 shows the numerical results and the analytical solutions for the nondimensionalized mean velocity as function of $\chi = c/c_\mathrm{max}$, where c is the volume fraction. $c_\mathrm{max} = \pi/6$ is the maximum volume fraction possible for a simple cubic arrangement. Two grid resolutions were used, with 52^3 and 104^3 grid points. Both results are in a good agreement with the analytical solution and the numerical results converge to the analytical solutions as the grid resolution is increased.

To test the convergence of the full method for a complex dendritic structure over a range of resolutions is problematic. Finer grids are too costly and coarser grids fail to represent the solution in any reasonable way. To examine how the actual error behaves, we have therefore studied a simpler problem, selected to yield reasonably good solution on coarse grids. The problem that we examine is the growth of an initially spherical seed in an undercooled melt flow, where surface tension is sufficiently high so that no dendrites start to appear. We place an initially spherical seed of diameter 0.2 in the middle of a 2^3 domain resolved by 8^3, 17^3, 31^3, and 61^3 grids, take the material properties of the solid and the melt equal, set the anisotropy equal to zero, $Pr = 1.0$, and $St = -0.2$. This value of the undercooling is low enough so that we get essentially fully converged solution on the finest grid and high enough so that the effects of the boundaries are small. The seed grows more rapidly in the upstream direction, but remains nearly spherical and Fig. 7(a) shows the upstream location of the interface (measured from the left inflow boundary), as a function of time. Obviously the results are converging as the grid is refined. There is no exact solution for this problem and to estimate the convergence rate, we assume that for the three finer grids the solution has reached its asymptotic behavior and higher order terms can be neglected. If that is true, we can write $d = d_\mathrm{exact} + Ch^n$ for the position of the tip of the upstream growing dendrite. If we know the solution on two grids, d_h and d_{2h}, then the solution can be extrapolated to $h = 0$ by Richardson's extrapolation. For $n = 1.85$ any two of

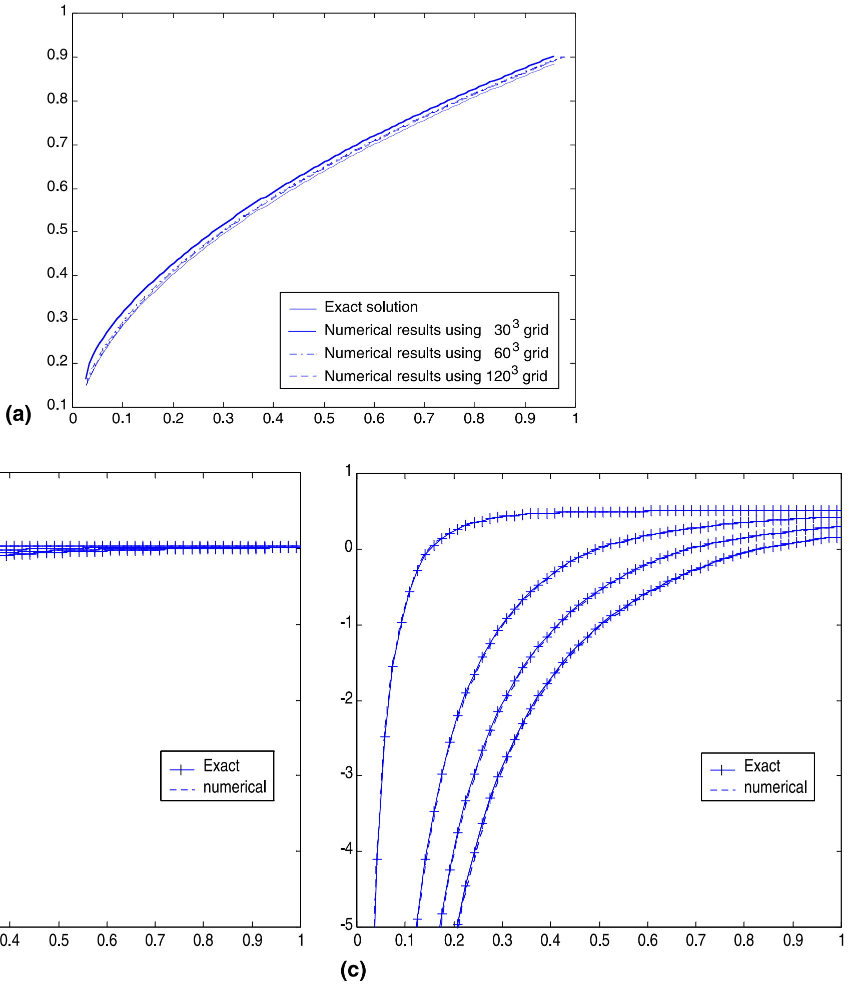

Fig. 5. Numerical and analytical results for the solidification of a sphere with a heat sink at the center. The parameters used are: $St = 0.5$, $q = -20$ in 2^3 domain. (a) The radius of the sphere as function of time using three grid sizes of 30^3, 60^3 and 120^3. (b) The temperature profiles along a line passing through the center at times 0.0276, 0.2976, 0.5676 and 0.8376, from left to right. (c) A blowup of the temperature profiles near the interface. The interface temperature equals 0.

the results for the three finer grids yield $d_{\text{exact}} = 0.3181$. In Fig. 7(b) the error is plotted versus the grid size along with a line with slope 2. We note that the nearly second order convergence rate found here is better than what is frequently observed for immersed boundary methods and we caution that in practical applications where the solution is obtained on a relatively coarse grid, the convergence is likely to be slower. These simulations were done on a 1 GHz PENTIUM III PC and took about 1 min for the 8^3 grid; 4 min for the 17^3 grid; 140 min for the 31^3 grid; and 68 h for the 61^3 grid.

5. Results

To show how the method described above handles complex interface shapes we have conducted two sets of simulations of unsteady three-dimensional dendritic growth, using the initial conditions and computa-

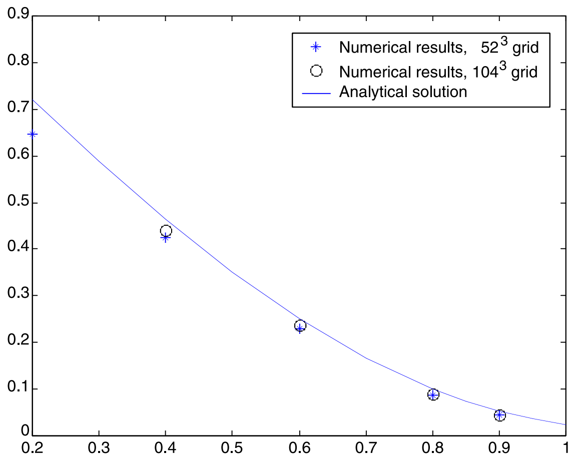

Fig. 6. Numerical and analytical results for the flow over an array of spheres. The nondimensional mean velocity is plotted as a function of $\chi = c/c_{\max}$, where c is the void fraction and $c_{\max} = \pi/6$ is the maximum void fraction for a cubical array.

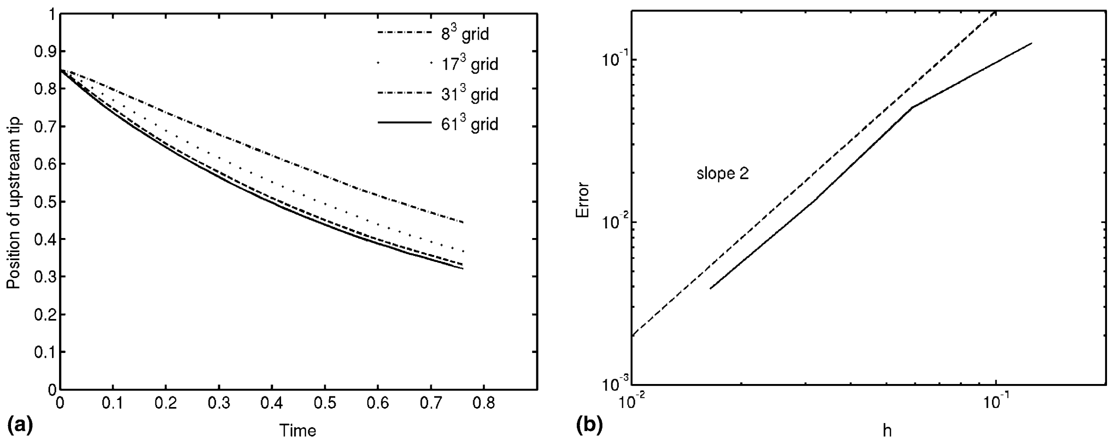

Fig. 7. The convergence of the location of the upstream tip for the test problem described in the text. (a) The location versus time, (b) the error versus the resolution.

tional setup shown in Fig. 2. For the first set we have attempted to assess the resolution needed for accurate simulations of complex dendritic growth by a resolution test and by a comparison with results for the growth of a two-dimensional dendrite.

For the first set of simulations we take $St = -0.15$, $Pr = 1.0$, $A_s = 0.3$, zero inverse kinetic mobility, and equal material properties. The flow Peclet numbers are $Pe_f = 0.0$ (no flow), $Pe_f = 0.002$, and $Pe_f = 0.006$. The dimensions of the computational domain, in the units defined in Table 1, are $10{,}200^n$, where n is the number of dimensions. The two-dimensional system was simulated using three grids with 128^2, 256^2 and 512^2 grid points and the three-dimensional system was computed on 128^3 and 256^3 grids.

Fig. 8 shows the solidification front, the temperature, and the velocity around both two- and three-dimensional dendrites at a relatively early time. The results for the three-dimensional systems are shown in the left column and for the two-dimensional ones in the column on the right hand side. In the top row $Pe_f = 0.0$ (no flow) and time $= 4.8 \times 10^6$. In the middle row $Pe_f = 0, 0.002$ and $Pe_f = 0.006$ in the bottom

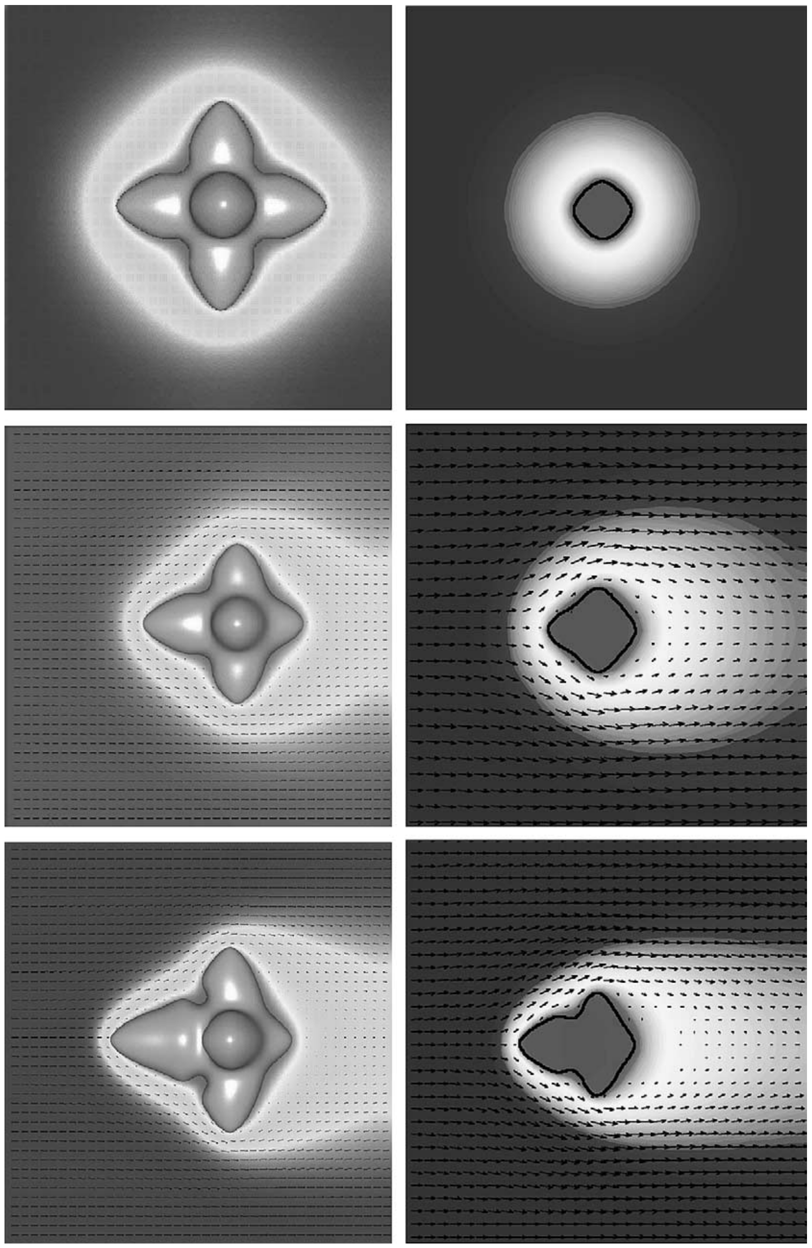

Fig. 8. The solidification front, the temperature, and the velocity around both two- and three-dimensional dendrites at a relatively early time. Three-dimensional results are shown in the left column and two-dimensional results in the right column. In the top row $Pe_{\mathrm{f}} = 0.0$ (no flow) and time $= 4.8 \times 10^6$. In the middle row $Pe_{\mathrm{f}} = 0$, 0.002 and 0.006 in the bottom row. Time is equal to 3.2×10^6 for the middle and the bottom row. Temperature is shown in gray, with the darker shades denoting colder material.

row. Time is equal to 3.2×10^6 for the middle and the bottom row. Temperature is shown in gray, with the darker shades denoting colder material. The radius of the initial seed is the same for both the two- and the three-dimensional case and both are perturbed by a wave with amplitude that is 20% of the radius of the seed. It is clear in all cases that the three-dimensional dendrites grow significantly faster than their two-dimensional counterparts. A careful examination of the wake (for the middle and the bottom row) also shows that the wake is smaller for the three-dimensional dendrites since the flow can go around the side of the arms growing perpendicular to the flow, instead of having to go around the tip, as in two-dimensions. Indeed, as already observed by Jeong et al. [16], the strong recirculating wakes behind two-dimensional dendrites at late times (see Part I) are not seen for the three-dimensional ones (see Fig. 12).

The velocities of the dendrite tips are plotted versus time in Fig. 9 for a flow Peclet number of $Pe_f = 0.002$. Fig. 9(a) shows the tip velocities for the two-dimensional simulations computed using the three grids. The tip velocity of the arm growing in the upstream direction, predicted using the 128^2 grid, is about 13% lower than the prediction using the 256^2 grid. The results from the 256^2 and the 512^2 grids are very

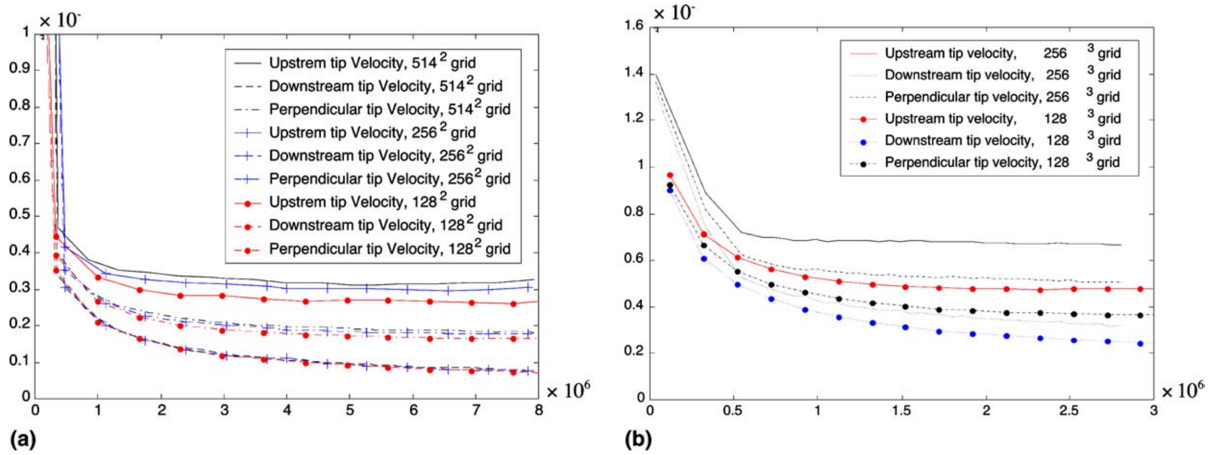

Fig. 9. The velocities of the tips of the dendrite arms for with the following parameters: $St = -0.15$, $Pr = 1.0$, $Pe_f = 0.002$, equal material properties and domain size of $(10.24 \times 10^3)^n$. Fig. 9(a) shows the tips velocities for two-dimensional systems using 128^2, 256^2 and 514^2 grids. Fig. 9(b) shows the tips velocities for three-dimensional systems using 128^2 and 256^2 grids.

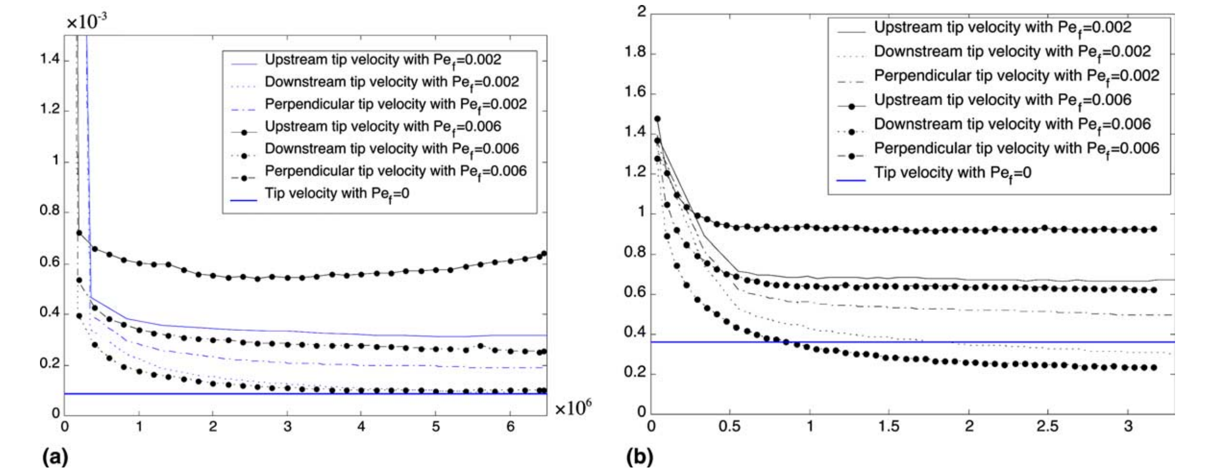

Fig. 10. The effect of the Peclet number on the velocities of the tips of the dendrite arms. The parameters used in the simulations are: $St = -0.15$, $A_s = 0.3$, $Pr = 1.0$, zero inverse kinetic mobility, and equal material properties. Fig. 10(a) shows results for two-dimensional systems and Fig. 10(b) shows results for three-dimensional systems.

similar and the difference between the upstream tip velocities predicted on the two grids is less than 4%. Fig. 9(b) shows the tip velocities for the three-dimensional dendrites, computed on both the 128^2 and the 256^2 grids. The difference between the upstream tip velocities predicted by the two grids is about 29%. If we assume a similar convergence rate for the three-dimensional case as the two-dimensional one, we would expect that the upstream tip velocity computed on a 512^3 grid would probably result in less than 9% difference from the results on the 256^3 grid. We have not, however, done a simulation using such a fine grid.

Fig. 10 shows the tip velocities versus time for $Pe_f = 0$, 0.002 and 0.006. The result for the two-dimensional system are shown in Fig. 10(a) and the results for the three-dimensional dendrite are shown in Fig. 10(b). The tip velocities without flow take much longer time to converge to a steady state value, especially in two-dimensions; therefore, only the steady state value of the tip velocity without flow are shown. The slow convergence to the steady state value is probably because the initial conditions lead to large heat release and since the Stefan number is low the heat diffuses away very slowly. In three-dimensions more melt is available to take up the heat. For nonzero Peclet number the flow advects the heat away from the growing dendrite, resulting in a faster convergence to steady state for the tip velocities. For the nonzero Peclet number simulations, all the tip velocities have converged to essentially steady state values except the velocities of the downstream tips in the three-dimensional simulations.

Table 2 shows the tip velocities of each dendrite arm, the radii of the tips, the Peclet numbers based on the tip velocity, and the stability parameter, σ^*, for $Pe_f = 0.0$ (no flow), 0.002 and 0.006 at the end of the simulations. With the exception of the velocities of the downstream tips in the three-dimensional simulations these are essentially steady-state values. Generally, the results show that increasing the melt flow results in a higher velocity of the tip of the arm growing in the upstream direction (and therefore the tip Peclet number) and a decrease in the tip radius. The results for the three- and the two-dimensional systems both show the same qualitative dependency of the velocities of the dendrite tips on the flow Peclet number – although the percentage changes are different – with the exception of the downstream arm. For the three-dimensional flow the downstream tip velocity decreases with increasing flow Peclet number, reflecting the

Table 2
Average tips velocities, radii, Peclet numbers and stability parameters (σ^*) as function of flow Peclet number in three and two-dimensional simulations

	Three-dimensional results			Two-dimensional results		
	$Pe_f = 0$	$Pe_f = 0.002$	$Pe_f = 0.006$	$Pe_f = 0$	$Pe_f = 0.002$	$Pe_f = 0.006$
Up-stream tip						
$V \times 10^3$	0.358	0.67	0.935	0.08	0.317	0.55
R	240	171	163	450	228	180
Pe_t	0.086	0.114	0.153	0.04	0.072	0.096
σ^*	0.10	0.12	0.08	0.12	0.12	0.115
Down-stream tip						
$V \times 10^3$	0.358	0.33	0.25	0.08	0.08	0.1
R	240	186	227	450	320	320
Pe_t	0.086	0.06	0.06	0.04	0.026	0.032
σ^*	0.10	0.15	0.16	0.12	0.24	0.21
Perpendicular tip						
$V \times 10^3$	0.358	0.5	0.625	0.08	0.186	0.268
R	240	182	175	450	268	242
Pe_t	0.09	0.09	0.11	0.04	0.05	0.07
σ^*	0.10	0.12	0.105	0.12	0.15	0.12

The parameters used in these simulations are: $St = -0.15$, $Pr = 1.0$, $A_s = 0.3$, zero inverse kinetic mobility, and equal material properties.

increased flow of heat from the upstream side and the reduced undercooling in the wake. The two-dimensional results are, however, strongly influenced by a recirculation in the wake and, as shown in Part I, increased flow rate results in a jet carrying flow from downstream toward the dendrite tip. This behavior is unlikely to be seen in real experiments and must therefore be regarded as an anomaly of two-dimensional systems. The dependency of the growth of the upstream arms on the fluid Peclet number for two-dimensional systems were also examined in Part I where it was shown that these trends agree with experimental and theoretical results. The stability parameter, σ^*, of the upstream and the perpendicular tips are almost the same, but usually slightly higher for the downstream one, possibly because the arm has not reached a steady-state velocity. The stability parameter for the upstream tip is essentially independent of Pe_f in our results. The dependency of σ^* on the flow remains a somewhat unresolved issue with computations and experiments giving the opposite trend [17]. Jeong et al. [17] found a slight decrease of σ^* for the upstream tip with the incoming flow velocity, but the dependency was week.

The three- and the two-dimensional results in Table 2 show the same trend for the effect of the flow on the behavior of the dendrite tips, but the percentage effect of the flow is quite different. In the three-dimensional simulations, introducing a melt flow of $Pe_f = 0.002$ and 0.006 result in an increase of the upstream tip velocity by 90% and 160%, over a tips velocity in the absence of flow. In the two-dimensional simulations, on the other hand, the introduction of melt flow results in an increase of the upstream tip velocity by a factor of 3.0 and 5.9 for $Pe_f = 0.002$ and 0.006, respectively. The much larger influence of the flow on the tip velocity for the two-dimensional system than the three-dimensional one is presumably due to the much lower tip velocity without flow in the two-dimensional case. In two-dimensions there is much smaller volume of undercooled liquid available to receive the latent head released by the solidification than in three-dimension and the growth is therefore slower. When flow is added and undercooled liquid is made available to absorb the heat, the difference between the flow and the no-flow case is greater in two-dimensions. The tip velocity without flow in the three-dimensional simulation is about 4.5 times higher than the corresponding tip velocity in the two-dimensional simulation. In three-dimensions the ratios of the inflow velocity of the undercooled liquid to the tip velocity without flow ($U_{in}/V_{tip,U=0}$) corresponding to $Pe_f = 0.002$ and 0.006 are about 5.5 and 16.5, respectively; while in two-dimensional simulations these ratios are 25 and 75, respectively. Similarly the effect of the flow on the upstream and perpendicular tips radii and Peclet numbers is stronger in the two-dimensional simulations.

The focus of the example above was on the relatively early growth. We conclude this section by showing two examples of simulations where we have followed the growth to much larger amplitude. Here $St = -0.25$, $A_s = 0.4$, $Pr = 0.1$, zero inverse kinetic mobility, equal material properties, and the computational domain is $28{,}000^3$ nondimensional units, resolved by a 256^3 grid. The two flow Peclet numbers are $Pe_f = 0.0015$ and 0.003. Although results for two-dimensional systems at the same parameters indicates that the overall shape of the dendrite is well predicted we expect that we would need about twice the resolution used here to obtain fully converged tip velocities. In Fig. 11 the $Pe_f = 0.0015$ dendrite is shown at times = 0, 0.9, 1.8, 2.7, 4.5, 5.4, 6.12 and 7.2×10^6. Fig. 12 shows the temperature and the flow fields around the $Pe_f = 0.003$ dendrite at time = 6.83×10^6. On the left the velocity and temperature are shown in a plane through the center of the computational domain and in the frame on the right in a plane cutting through a side branch of the dendrite, growing toward the viewer. In addition to leading to significant asymmetry in the growth of the main dendritic arms, the flow strongly affects the growth of side branches on the arm growing perpendicularly to the flow. Essentially all the side branches grow on the upstream side, some of them sufficiently fast to prevent growth of side branches from the upstream arm. This figure illustrates the importance of three-dimensional simulations since in two-dimensional simulations the flow has to go around the perpendicular arms and therefore has less effect on the side branches of these arms. See also Jeong et al. [16], where the difference is discussed in more detail. The temperature and the flow fields around the side branches of the perpendicular arms looks similar to the fields around the upstream arm. The temperature of this flow is, however, affected by the heat released from the upstream arm.

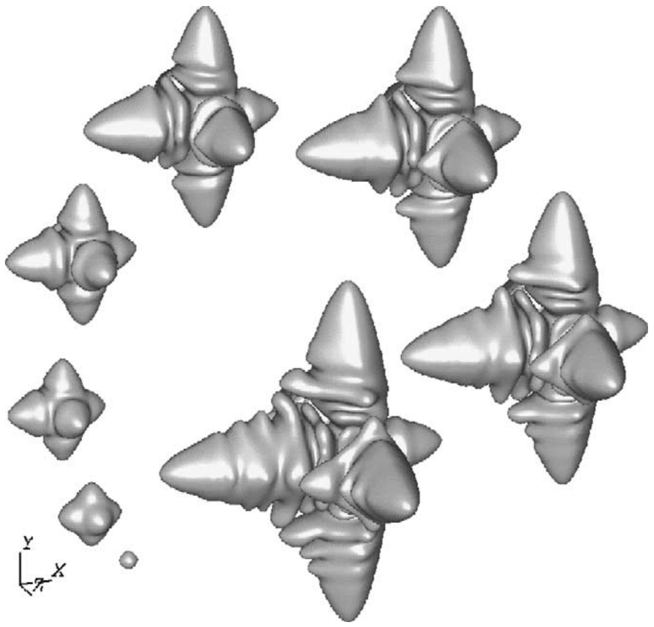

Fig. 11. Growth of a dendrite with $Pe_f = 0.0015$. The dendrite is shown at times = 0, 0.9, 1.8, 2.7, 4.5, 5.4, 6.12 and 7.2×10^6. The parameters used here are: $St = -0.25$, $A_s = 0.4$, $Pr = 0.1$, zero inverse kinetic mobility, and equal material properties.

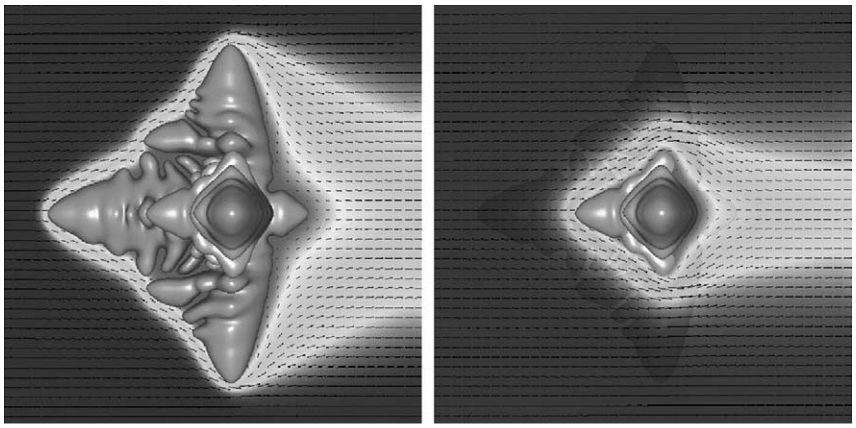

Fig. 12. The temperature and the velocity fields around a dendrite with $Pe_f = 0.003$ at time = 6.83×10^6. The parameters used in the simulation are: $St = -0.25$, $A_s = 0.4$, $Pr = 0.1$, zero inverse kinetic mobility, and equal material properties. The left frame shows the temperature and the velocity in a plane through the center of the domain, the right frame shows the temperature and the velocity in a plane cutting through the arm growing perpendicular to the flow, toward the reader.

6. Conclusions

We have presented a front tracking method for the simulations of three-dimensional dendritic growth with convection. The method is an extension of the method for two-dimensional systems presented by Al-Rawahi and Tryggvason [1] and the results are similar. The introduction of melt flow results in an increase of the growth rate of the arm growing into the flow, promoting more side branch formation on the upstream side. The flow also results in a reduction of the growth rate of the downstream arm. However, the effect of the flow on the downstream arms is smaller here than in two-dimensions, since the fluid can go around the side of the dendrite arms growing perpendicular to the flow, instead of having to go around the tip of these arms. The results also show that a fairly fine grid resolution is needed for accurate results. Although we believe – based on both the tests presented here as well our experience in simulating two-dimensional systems – that the results presented here accurately represent the behavior of a growing dendrite for the parameters used, it is clear that a 256^3 uniform grid is near the acceptable minimum needed and a finer grid would be desirable for more detailed studies. Although we did not examine the same system as Jeong et al. [16,17], our results show the same trend, including the smaller influence of the flow on the downstream arms as compared with two-dimensional flows.

Acknowledgements

Al-Rawahi would like to thank Sultan Qaboos University for financial support. This research was also supported in part by NASA grants NAG3-2162 and NAG3-2583. We would like to thank Dr. Bernard Bunner for many helpful discussions and for the use of his multigrid code to solve the pressure equation.

References

[1] N. Al-Rawhai, G. Tryggvason, Numerical simulation of dendritic solidification with convection: Two-dimensional flow, J. Comput. Phys. 180 (2002) 471–496.
[2] M.E. Glicksman, S.C. Huang, Convective heat transfer during dendritic growth, in: J. Zierep, H. Ortel (Eds.), Convective Transport and Instability Phenomena, Braun, Karlsruhe, 1982, p. 557.
[3] P. Zhao, M. Venere, J.C. Heinrich, D.R. Poirier, Modeling dendritic growth of a binary alloy, J. Comput. Phys. 188 (2003) 434–461.
[4] R. Kobayashi, A numerical approach to three-dimensional dendritic solidification, Exp. Math. 3 (1994) 59.
[5] A. Karma, W.-J. Rappel, Phase field method for computationally efficient modeling of solidification with arbitrary interface kinetics, Phys. Rev. E 53 (4) (1996) 3017.
[6] A. Karma, W.-J. Rappel, Numerical simulation of three-dimensional dendritic growth, Phys. Rev. Lett. 77 (19) (1996) 4050.
[7] A. Schmidt, Computation of three-dimensional dendrites with finite element, J. Comput. Phys. 125 (1996) 293.
[8] M. Plapp, A. Karma, Multiscale finite-difference-diffusion Monte-Carlo method for simulating dendritic solidification, J. Comput. Phys. 165 (2000) 592.
[9] F. Gibou, R. Fedkiw, L.-T. Cheng, M. Kang, A second order accurate symmetric discretization of the Poisson equation on irregular domains, J. Comput. Phys. 176 (2002) 205–227.
[10] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Sci. Comput. 19 (2003) 183–199.
[11] R. Tonhardt, G. Amberg, Phase field simulation of dendritic growth in a shear flow, J. Crystal Growth 194 (1998) 406.
[12] C. Beckermann, H.-J. Diepers, I. Steinbach, A. Karma, X. Tong, Modeling melt convection in phase-field simulations of solidification, J. Comput. Phys. 154 (1999) 468–496.
[13] E. Bansch, A. Schmidt, Simulation of dendritic crystal growth with thermal convection, Interf. Free Boundaries 2 (2000) 95.
[14] H.S. Udaykumar, R. Mittal, W. Shyy, Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids, J. Comput. Phys. 153 (1999) 535–574.
[15] D. Juric, Presented at the Molding of Casting, Welding and Advanced Solidification Processes VIII Workshop, San Diego, CA, 1998.
[16] J.-H. Jeong, N. Goldenfield, J.A. Dantzig, Phase field model for three-dimensional dendritic growth with fluid flow, Phys. Rev. E 64 (2001) 041602.
[17] J.-H. Jeong, J.A. Dantzig, N. Goldenfeld, Dendritic growth with fluid flow in pure materials, Metall. Mater. Trans. A 34 (2003) 459–466.
[18] Y. Lu, C. Beckermann, A. Karma, Convection effects in three-dimensional dendritic growth, in: Proceedings of the 2001 Fall MRS Meeting in Boston, MA, 2001 (available at http://www.engineering.uiowa.edu/~becker/phasefield.htm).

[19] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Ann. Rev. Mater. Res. 32 (2002) 163–194.
[20] M. Muschol, D. Liu, H.Z. Cummins, Surface-tension-anisotropy measurements of succinonitrile and pivalic acid: comparison with microscopic solvability theory, Phys. Rev. A 46 (1992) 1038–1050.
[21] D. Juric, G. Tryggvason, A front tracking method for dendritic solidification, J. Comput. Phys. 123 (1996) 127–148.
[22] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[23] E.A. Fadlun, R. Verzicco, P. Orlandi, J. Mohd-Yusof, Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations, J. Comput. Phys. 161 (2000) 35.
[24] C.E. Weatherburn, Differential Geometry of Three Dimensions, vol. I, Cambridge, 1927.
[25] C.S. Peskin, Numerical-analysis of blood-flow in heart, J. Comput. Phys. 25 (1977) 220.
[26] W. Gropp, E. Lusk, A. Skjellum, Using MPI: Portable Parallel Programming with the Message-Passing Interface, The MIT Press, London, 1995.
[27] B. Bunner. Numerical Simulations of Gas–Liquid Bubbly Flows. Ph.D. Dissertation. The University of Michigan, 2000.
[28] S. Paterson, Propagation of boundary of fusion, Proc. Glasgow Math. Assoc. 1 (1952) 4247.
[29] A. Sangani, A. Acrivos, Slow flow through a periodic array of spheres, Int. J. Multiphase Flow 8 (1982) 343–360.