Available online at www.sciencedirect.com
ScienceDirect

Journal of Computational Physics 221 (2007) 9–40

JOURNAL OF COMPUTATIONAL PHYSICS

www.elsevier.com/locate/jcp

A level set simulation of dendritic solidification of multi-component alloys

Lijian Tan, Nicholas Zabaras *

Materials Process Design and Control Laboratory, Sibley School of Mechanical and Aerospace Engineering, 188 Frank H.T. Rhodes Hall, Cornell University, Ithaca, NY 14853-3801, USA

Received 19 December 2005; received in revised form 29 May 2006; accepted 2 June 2006
Available online 24 July 2006

Abstract

A level set method combining features of front tracking methods and fixed domain methods is presented to model microstructure evolution in the solidification of multi-component alloys. Phase boundaries are tracked by solving the multi-phase level set equations. Diffused interfaces are constructed from these tracked phase boundaries using the level set functions. Based on the assumed diffused interfaces, volume-averaging techniques are applied for energy, species and momentum transport. Microstructure evolution in multi-component alloy systems is predicted using realistic material parameters. The methodology avoids the difficulty of parameter identification needed in other diffused interface models, and allows easy application to various practical alloy systems. Techniques including fast marching, narrow band computing and adaptive meshing are utilized to speed up computations. Several numerical examples are considered to validate the method and examine its potential for modeling solidification of practical alloy systems. These examples include two- and three-dimensional solidification of a binary alloy in an undercooled melt, a study of planar/cellular/dendritic transition in the solidification of a Ni–Cu alloy, and eutectic and peritectic solidification of an Fe–C system. Adaptive mesh refinement in the rapidly varying interface region makes the method practical for coupling the microstructure evolution at the meso-scale with buoyancy driven flow in the macro-scale, which is shown in the solidification of a Ni–Al–Ta ternary alloy.

Keywords: Level set method; Dendritic growth; Solidification; Multi-component alloys; Multi-phase; Eutectic; Peritectic; Fluid flow

1. Introduction

Simulation of solidification and crystal growth processes has been of growing interest because of its importance in many technological applications. The classical Stefan problem with a sharp interface is a well accepted

* Corresponding author. Fax: +1 607 255 1222.
E-mail address: zabaras@cornell.edu (N. Zabaras).
URL: http://mpdc.mae.cornell.edu/ (N. Zabaras).

0021-9991/$ - see front matter © 2006 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2006.06.003

model of solidification of pure materials and alloys. Front-tracking techniques have been used successfully to reproduce the complex dendritic structure in crystal growth in undercooled melts [1–5]. Such methodologies directly enforce the freezing interface temperature (Gibbs–Thomson) relation and energy balance (Stefan) condition. However, the complexity for handling interfaces in all possible solidification conditions such as advancing/receding, merging/splitting and interface unit normal vector/curvature computations limit the applicability of front-tracking methods to single solid phase systems.

Phase field methods have been used extensively for modeling solidification processes for various systems including pure materials, binary alloys and multi-component alloys [6,7]. The basic idea is to employ a phase field variable Φ that varies smoothly from zero to unity between the two phases over the diffused-interface region, which has a small but numerically resolvable thickness. The phase field variable is used to distribute the interfacial forces over the diffused freezing region. It is governed by a phase field control equation derived from the thermodynamics of phase transition. Important physical mechanisms, such as curvature, anisotropy and kinetics effects, are implicitly incorporated in the phase field control equation. This leads to many computational advantages. For example, one does not need to compute interfacial geometric quantities such as interface curvature and outward normal vector.

The phase field method reduces to the standard sharp interface formulation in the limit of vanishing interface thickness. The quality of the solution deteriorates with increasing interface thickness. It has been shown that the interface thickness must be smaller than the capillary length for the solution to converge to the sharp interface limit [8,9]. This leads to significant computational effort especially for modeling dendritic growth in the presence of convection. However, the techniques of adaptive gridding and parallel computing have resulted in assuaging this drawback to some extent [10,11]. Another drawback of the phase field methods is related to the large number of parameters involved in the solution of the evolution equations that are rather difficult to compute for real materials.

The level set method handles the sharp interface front directly and efficiently, and avoids the asymptotic analysis needed in phase field models [12,13]. In this method, interfacial geometric quantities such as curvature and outward normal can be easily calculated using the level set variable ϕ, which is the signed distance from the interface. The method was first applied to Stefan problems in [14]. Level set calculations for dendritic growth were reported in [15]. In these works [14,15], the heat flux at nodes near the interface is interpolated in order to calculate the interface velocity via the Stefan condition. Direct application of temperature boundary conditions on the interface and the computation of heat fluxes from the temperature nodal values usually lead to energy conservation issues associated with discretization error. This may result in large variation of the computed crystal shapes if meshes of different sizes and orientations are used. In our previous work [16], an energy conserving implementation of the level set method was used to study pure material solidification without the need to apply the temperature boundary condition explicitly at the freezing interface. The basic idea is to track the interface using the level set method, but perform the heat and momentum transport computation in a diffused-interface, which is constructed from the level set computed values.

In this work, our earlier model is extended to binary alloy and multi-component alloy systems. For systems with multiple solid phases, phase boundaries are tracked by solving the multi-phase level set equations. Computations are performed on unstructured meshes (triangle meshes in 2D and tetrahedral meshes in 3D) using adaptive refinement. To the best of our knowledge, this computationally-intensive work provides the first reported development and finite element method implementation of level set methods for binary and multi-component alloy solidification in both two- and three-dimensions including the effects of melt flow. The plan of this paper is as follows. Section 2 introduces the sharp interface model including the effects of fluid flow. Section 3 discusses a diffused-interface approximation of the Stefan problem using volume-averaging techniques. Numerical techniques including multi-phase motion re-initialization, fast marching, adaptive meshing, narrow band computing, adaptive time stepping, finite element implementation and the overall solution procedure are discussed in Section 4. Section 5 presents applications to binary alloy crystal growth, a planar/cellular/dendritic transition study, three-dimensional crystal growth, eutectic/peritecitc multi-phase growth and ternary alloy solidification with melt convection effects. A mesh anisotropy study and a mesh adaptive refinement study are included in the binary alloy crystal growth example. Finally, Section 6 provides a brief summary of this work.

2. Sharp interface model for the solidification of multi-component alloys

Let us consider the solidification of an alloy system with n components that results in N phases. Each phase α is assumed to occupy a region Ω^α with temperature T^α and concentration C_i^α, for each component i. Each pair of these regions, Ω^α and Ω^β, are separated by the interface $\Gamma^{\alpha\beta}$, which is $\emptyset$ when the two phases are not neighboring each other. The phase domain Ω^α is time-dependent and its boundary Γ^α, which is a union of interfaces with all other phases ($\Gamma^\alpha = \cup_{\gamma\neq\alpha}\Gamma^{\alpha\gamma}$), is moving with normal velocity V^α. The unit normal vector $\boldsymbol{n}^\alpha$ is defined as pointing away from the phase region Ω^α. The total domain $\Omega = \cup_\alpha \Omega^\alpha$ containing all phases and its external boundary $\partial\Omega$ is assumed to be time-independent. On the interface of two phases $\Gamma^{\alpha\beta}$, which is a part of both Γ^α and Γ^β, the normal growth velocities and unit normal vectors are related as $V^\alpha = -V^\beta$ and $\boldsymbol{n}^\alpha = -\boldsymbol{n}^\beta$, respectively. In this work, we use the subscript i to denote different species with $i = 1$ indicating the major component and $i = 2, 3, \ldots, n$ the minor components. The superscript α is used to denote different phases with $\alpha = 1$ referring to the liquid phase and $\alpha = 2, 3, \ldots, N$ the various solid phases. To emphasize the differences between the solid phases and the liquid phase, we also use s to denote different solid phases with $s = 2, 3, \ldots, N$ and ℓ to denote the liquid phase with $\ell = 1$ (see Fig. 1).

To allow us to concentrate on key aspects of the solidification of multi-component alloy systems, the following assumptions are introduced for the transport of heat, momentum and solute:

(1) Constant density ρ, constant thermo-physical and transport properties for each phase including thermal conductivities k^α, heat capacities c^α, latent heats of transformation of the liquid phase ℓ to each solid phase s (denoted as L^s), partition coefficients of the liquid phase ℓ and solid phase s with respect to component i (denoted as $k_{p_i}^s$), solutal diffusion coefficients D_i^ℓ, and viscosity μ^ℓ.

(2) Zero solute diffusivity in all solid phases, $D_i^s = 0$, and negligible solid–solid phase transformation, $V^\alpha = 0$ on $\Gamma^{\alpha\beta}$, $\forall\alpha, \beta \neq \ell$.

Following standard notation, the governing equations in the presence of fluid flow are given as follows:

$$\nabla \cdot \boldsymbol{v}(\boldsymbol{x}, t) = 0, \quad \boldsymbol{x} \in \Omega^\ell, \tag{1}$$

$$\rho\left(\frac{\partial \boldsymbol{v}(\boldsymbol{x}, t)}{\partial t} + \nabla \boldsymbol{v}(\boldsymbol{x}, t)\boldsymbol{v}(\boldsymbol{x}, t)\right) = -\nabla p(\boldsymbol{x}, t)\boldsymbol{I} + \nabla \cdot \mu[\nabla \boldsymbol{v}(\boldsymbol{x}, t) + (\nabla \boldsymbol{v}(\boldsymbol{x}, t))^{\mathrm{T}}] + \boldsymbol{b}, \quad \boldsymbol{x} \in \Omega^\ell, \tag{2}$$

$$\rho c^s \frac{\partial T(\boldsymbol{x}, t)}{\partial t} = k^s \nabla^2 T(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Omega^s, \; s = 2, 3, \ldots, N, \tag{3}$$

$$\rho c^\ell \left(\frac{\partial T(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v} \cdot \nabla T(\boldsymbol{x}, t)\right) = k^\ell \nabla^2 T(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Omega^\ell, \tag{4}$$

$$\frac{\partial C_i^\ell(\boldsymbol{x}, t)}{\partial t} + \boldsymbol{v} \cdot \nabla C_i^\ell(\boldsymbol{x}, t) = D_i^\ell \nabla^2 C_i^\ell(\boldsymbol{x}, t), \quad \boldsymbol{x} \in \Omega^\ell, \; i = 2, 3, \ldots, n, \tag{5}$$

where $\boldsymbol{v}$ is the melt flow velocity and $\boldsymbol{b}$ is the buoyancy body force.

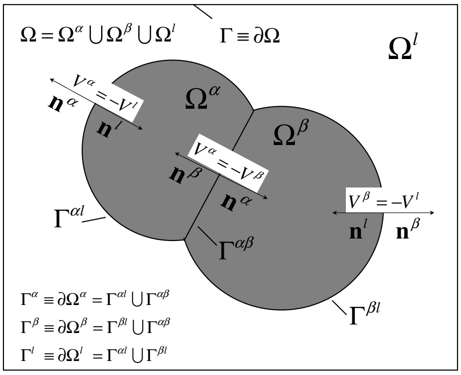

$\Omega = \Omega^\alpha \cup \Omega^\beta \cup \Omega^l$
$\Gamma \equiv \partial\Omega$
Ω^l
Ω^α
Ω^β
$V^\alpha = -V^l$
$\mathbf{n}^\alpha$
$\mathbf{n}^l$
$V^\alpha = -V^\beta$
$\mathbf{n}^\beta$
$\mathbf{n}^\alpha$
$V^\beta = -V^l$
$\mathbf{n}^l$
$\mathbf{n}^\beta$
$\Gamma^{\alpha l}$
$\Gamma^{\alpha\beta}$
$\Gamma^{\beta l}$
$\Gamma^\alpha \equiv \partial\Omega^\alpha = \Gamma^{\alpha l} \cup \Gamma^{\alpha\beta}$
$\Gamma^\beta \equiv \partial\Omega^\beta = \Gamma^{\beta l} \cup \Gamma^{\alpha\beta}$
$\Gamma^l \equiv \partial\Omega^l = \Gamma^{\alpha l} \cup \Gamma^{\beta l}$

Fig. 1. Schematic of the solidification of a multi-component alloy system with multiple solid phases (here phases α and β).

The temperature on the interface $\Gamma^{s\ell}$ (denoted as $T_I^{s\ell}$) is equal to the equilibrium temperature T_*^s. This equilibrium temperature is given from the Gibbs–Thomson relation as follows:

$$T_*^s = T^{s\ell}(C_1^\ell, C_2^\ell, \ldots, C_n^\ell) + \varepsilon_c^s \kappa^s + \varepsilon_V^s V^s, \quad s = 2, 3, \ldots, N, \tag{6}$$

where $T^{s\ell}(C_1^\ell, C_2^\ell, \ldots, C_n^\ell)$ is the liquidus temperature obtained from the phase diagram or thermodynamic databases, κ^s is the curvature of the freezing interface, and ε_c^s and ε_V^s are the curvature and kinetic undercooling coefficients, respectively.

With $C_i^s = k_{p_i}^s C_i^\ell$ on $\Gamma^{s\ell}$, solutal rejection at the solid/liquid interfaces is governed by the following equation:

$$D_i^\ell \frac{\partial C_i^\ell}{\partial \boldsymbol{n}^s} = -(1 - k_{p_i}^s) C_i^\ell V^s \boldsymbol{n}^s \quad \text{on } \Gamma^{s\ell}, \quad s = 2, 3, \ldots, N. \tag{7}$$

The interface velocity V^s on $\Gamma^{s\ell}$ is related to the heat flux jump at the freezing front by the classical Stefan equation (interfacial energy balance) as follows:

$$V^s = -V^\ell = (q^s - q^\ell)/(\rho L^s) \quad \text{on } \Gamma^{s\ell}, \; s = 2, 3, \ldots, N, \tag{8}$$

where q denotes the heat flux $k\nabla T \cdot \boldsymbol{n}^s$.

The level set function ϕ^α is defined to be the signed distance from the phase boundary Γ^α,

$$\phi^\alpha(\boldsymbol{x}, t) \equiv \begin{cases} +d^\alpha(\boldsymbol{x}, t), & \boldsymbol{x} \notin \Omega^\alpha, \\ 0, & \boldsymbol{x} \in \Gamma^\alpha, \\ -d^\alpha(\boldsymbol{x}, t), & \boldsymbol{x} \in \Omega^\alpha, \end{cases} \tag{9}$$

where $d^\alpha(\boldsymbol{x}, t)$ is the normal distance of a point $\boldsymbol{x}$ from Γ^α. The phase boundary position is thus implicitly stored in ϕ^α. The idea behind the level set method is to move ϕ^α with speed V^α at phase boundary Γ^α, which is extracted from Eq. (8). The equation of motion governing ϕ^α is given as follows:

$$\phi_t^\alpha + V^\alpha |\nabla \phi^\alpha| = 0, \quad \alpha = 1, 2, \ldots, N. \tag{10}$$

With the above introduction of the level set function, we will re-write the Stefan condition for the classical Stefan problem as follows:

$$V^s = -V^\ell = (q^s - q^\ell)/(\rho L^s), \quad q^s = \lim_{\phi^s \to 0^-} q, \quad q^\ell = \lim_{\phi^s \to 0^+} q \text{ on } \Gamma^{s\ell}, \tag{11}$$

where the notation 0^+ and 0^- is used here for notational simplification to denote the values of ϕ^s ($s = 2, 3, \ldots, N$) as we approach the freezing front from the solid and liquid sides, respectively.

The governing equations (1)–(5) in the phase domains $\Omega^\alpha, \alpha = 1, 2, \ldots, N$, the boundary conditions equations (6) and (7) on the solid–liquid interfaces $\Gamma^{s\ell}$, $\ell = 1$, $s = 2, \ldots, N$, the evolution of the phase boundaries given in Eq. (10) with velocity specified by the Stefan condition equation (11), and appropriate initial and boundary thermal, solute and velocity conditions, completely define what we here refer to as the *multi-component alloy solidification problem*.

3. Diffused-interfaces and modeling of transport processes

3.1. Important assumptions

Many of the difficulties in the implementation of the sharp-interface multi-component alloy solidification problem defined earlier are related to the application of the thermal essential boundary condition given in Eq. (6) and the solute rejection flux condition given in Eq. (7). To avoid directly applying these conditions as needed in front tracking methodologies, we introduce a diffused-interface approximation to allow convenient modeling of the underlying transport processes while still maintaining an explicit tracking of the phase boundaries. This model was applied earlier for the solidification of pure materials [16]. Volume-averaging approximations used extensively for macroscopic solidification process modeling will be used here for modeling dendritic alloy solidification. Using a diffused-interface model for the transport equations and volume-averaging to derive the corresponding equations is a well established approach for modeling solidification using the phase field method [17].

Assumption 1. We assume that solidification occurs in a diffused zone of width $2w$ that is symmetric around $\phi^\ell = 0$. The parameter w used here is defined in terms of the spatial discretization selected (usually taken as the size of one finite element) and it is not related to the physics of phase transformation which usually occurs on a thickness of the order of atomic distances. The choice of w in this model is comparable to the choice for the interface width in the phase field method, and can affect the accuracy of the results, just like in the phase field method.

Following ideas similar to those in diffused-interface models, let us define the function $\Phi^\alpha(\boldsymbol{x}, t)$ as follows:

$$\Phi^\alpha(\boldsymbol{x},t) \equiv \begin{cases} 0, & \phi^\alpha(\boldsymbol{x},t) > w, \\ 1, & \phi^\alpha(\boldsymbol{x},t) < -w, \\ -\phi^\alpha(\boldsymbol{x},t)/2w + 0.5, & \phi^\alpha(\boldsymbol{x},t) \in [-w, w]. \end{cases} \tag{12}$$

This function was used in our earlier work [16] for the solidification of pure materials to define the liquid volume fraction within the diffused-interface. However, this definition cannot be used for multi-component alloy solidification. Indeed, for $\boldsymbol{x}$ near the phase interface $\Gamma^{\alpha\beta}$ ($\phi^\alpha < w$, $\phi^\beta < w$) and far away from triple points ($\phi^\gamma > w$, $\forall \gamma \neq \alpha, \beta$), we have $\Phi^\alpha + \Phi^\beta = 1$ and $\Phi^\gamma = 0$, $\forall \gamma \neq \alpha, \beta$ and Φ can be used as phase volume fraction. However, for $\boldsymbol{x}$ near triple points (interface of three phases) α, β and γ ($\phi^\alpha < w$, $\phi^\beta < w$, $\phi^\gamma < w$), we will have $\Phi^\alpha + \Phi^\beta + \Phi^\gamma > 1$ and Φ cannot be used as phase volume fraction. We herein introduce another variable for phase volume fraction as follows:

$$\epsilon^\alpha \equiv \Phi^\alpha / \sum_\beta \Phi^\beta. \tag{13}$$

Assumption 2. The diffused-interface between a solid phase and the liquid phase is treated as a porous medium with Kozeny–Carman approximation for the permeability:

$$K(\epsilon^\ell) = \frac{K_0 \epsilon^{\ell 3}}{(1-\epsilon^\ell)^2}, \tag{14}$$

where K_0 is a permeability constant taken to be a small value of 10^{-5} in our computations.

The above assumption is of no physical significance in the present calculations. It is simply for the convenience of applying the no-slip boundary condition on the growing solid phases with a fixed-grid. This idea of using a thin region near the interface to apply the no-slip condition is a well accepted technique in the literature. The present Kozeny–Carman approach for enforcing the no-slip condition depends on the selection of w and is essentially equivalent to the one in [17]. In [18], another technique using variant viscosity in the diffused-interface was applied to account for the no-slip condition on the solid–liquid interface. In our previous work [16], we had shown that the variant viscosity and the Kozeny–Carman approach lead to nearly identical results with velocity of very small magnitude within the thin artificially-constructed diffused solid–liquid interfaces.

Assumption 3. The solid–liquid interface temperature $T_I^{s\ell}$ is allowed to vary from the equilibrium temperature T_*^s in such a way that

$$\frac{\mathrm{d}T_I^{s\ell}}{\mathrm{d}t} = -k_N(T_I^{s\ell} - T_*^s), \tag{15}$$

where k_N controls the rate with which $T_I^{s\ell}$ is designed to approach the desired equilibrium temperature.

With the assumption of Eq. (15), and by further assuming that the mean temperature in the diffused-interface (mushy) zone can be approximated as $T_I^{s\ell}$, the energy conservation for the diffused freezing zone leads to what we here refer to as *the extended Stefan condition* [16]:

$$V^s = \frac{q^s - q^\ell}{\rho L^s} + \frac{2\tilde{c}^s w}{L^s} k_N(T_*^s - T_I^{s\ell}), \quad q^s = \lim_{\phi^s \to -w^-} q, \quad q^\ell = \lim_{\phi^s \to w^+} q \text{ on } \Gamma^{s\ell}, \tag{16}$$

where $\tilde{c}^s \equiv 0.5(c^s + c^\ell)$ and the heat fluxes q^ℓ and q^s are computed at the boundaries of the diffused-interface. According to the stability analysis discussed in [16], k_N is selected based on the time step size as $k_N = 1/\Delta t$.

The extended Stefan equation is used to compute the interface velocity and evolve the interface. It weakly enforces the Gibbs–Thomson relation on the moving interface. The other two important conditions on the interface (the one governing solute rejection and the no slip condition for fluid flow) are also weakly enforced using the volume averaging techniques that are reviewed next.

3.2. Brief review of the volume-averaging transport equations

In order to set forward the notation and fundamental equations used for modeling the transport phenomena relevant to dendritic growth, in the section we will briefly summarize the volume-averaged transport equations applied to the diffused interface initially developed in [17] in the context of phase field solidification models. A more detailed presentation is given in [19,20].

The equations representing momentum, energy, mass and species transport in each phase α take the following general form:

$$\frac{\partial \Theta^\alpha}{\partial t} + \nabla \cdot (\Theta^\alpha \boldsymbol{v}^\alpha) = \nabla \cdot \mathbf{J}^\alpha + S^\alpha, \tag{17}$$

for appropriate selection of the field variable Θ, diffusion flux $\mathbf{J}$ and source term S.

A phase function v^α is introduced as taking value 1 in phase α and zero elsewhere. The volume-averaged quantity $\langle \Psi^\alpha \rangle$ of any quantity $\Psi(\boldsymbol{x}, t)$ in phase α over the entire averaging volume $\mathrm{d}V$ can now be introduced as:

$$\langle \Psi^\alpha \rangle = \frac{1}{\mathrm{d}V} \int_{\mathrm{d}V} \Psi^\alpha v^\alpha(\boldsymbol{x}, t)\, \mathrm{d}v. \tag{18}$$

Similarly, one can introduce the intrinsic volume-averaged quantity $\langle \Psi^\alpha \rangle^\alpha$ (averaged value of $\Psi(\boldsymbol{x}, t)$ in the control volume $\mathrm{d}V^\alpha$) as:

$$\langle \Psi^\alpha \rangle^\alpha = \frac{1}{\mathrm{d}V^\alpha} \int_{\mathrm{d}V} \Psi^\alpha v^\alpha(\boldsymbol{x}, t)\, \mathrm{d}v = \frac{\langle \Psi^\alpha \rangle}{\epsilon^\alpha}, \tag{19}$$

where ϵ^α was defined earlier in Eq. (13).

Finally, the fluctuating component $\hat{\Psi}^\alpha$ is commonly introduced to represent the deviation of Ψ^α from the intrinsic volume-averaged $\langle \Psi^\alpha \rangle^\alpha$. It is given by:

$$\hat{\Psi}^\alpha = (\Psi^\alpha - \langle \Psi^\alpha \rangle^\alpha) v^\alpha. \tag{20}$$

Multiplying each side of Eq. (17) by v^α, integrating it over the averaging volume $\mathrm{d}V$, one can obtain the following averaged transport equation for phase α [19]:

$$\begin{aligned} \frac{\partial \langle \Theta^\alpha \rangle}{\partial t} + \nabla \cdot \epsilon^\alpha \langle \Theta^\alpha \rangle^\alpha \langle \boldsymbol{v}^\alpha \rangle^\alpha = {} & \nabla \cdot \langle \mathbf{J}^\alpha \rangle + \langle S^\alpha \rangle + \nabla \cdot \frac{1}{\mathrm{d}V} \int_{\mathrm{d}V} (-\widehat{\Theta^\alpha}\, \widehat{\boldsymbol{v}^\alpha})\, \mathrm{d}v + \frac{1}{\mathrm{d}V} \int_{\mathrm{d}A^\alpha} \mathbf{J}^\alpha \cdot \boldsymbol{n}^\alpha\, \mathrm{d}A \\ & + \frac{1}{\mathrm{d}V} \int_{\mathrm{d}A^\alpha} \Theta^\alpha (\boldsymbol{V}^\alpha - \boldsymbol{v}^\alpha) \cdot \boldsymbol{n}^\alpha\, \mathrm{d}A, \end{aligned} \tag{21}$$

where $\mathrm{d}A^\alpha$ is the interfacial area of phase α with the other phases, $\boldsymbol{n}^\alpha$ is the outward unit normal of the infinitesimal element of area $\mathrm{d}A$ of phase α, and $\boldsymbol{V}^a$ is the growth velocity of the boundary of phase α.

In comparison with the original transport equation (17), three extra terms I_D^α, I_J^α and I_Q^α appear from the volume-averaging procedure of the form:

$$I_D^\alpha \equiv \nabla \cdot \frac{1}{\mathrm{d}V} \int_{\mathrm{d}V} (-\widehat{\Theta^\alpha}\, \widehat{\boldsymbol{v}^\alpha})\, \mathrm{d}v, \tag{22}$$

$$I_Q^\alpha \equiv \frac{1}{\mathrm{d}V} \int_{\mathrm{d}A^\alpha} \Theta^\alpha (\boldsymbol{V}^\alpha - \boldsymbol{v}^\alpha) \cdot \boldsymbol{n}^\alpha\, \mathrm{d}A, \tag{23}$$

$$I_J^\alpha \equiv \frac{1}{\mathrm{d}V} \int_{\mathrm{d}A^\alpha} \mathbf{J}^\alpha \cdot \boldsymbol{n}^\alpha\, \mathrm{d}A. \tag{24}$$

Almost all models reported in the literature neglect the I_D^α term [21–23]. The same approximation is considered here as well. The term I_Q^α accounts for the interfacial transfer due to phase change, whereas I_J^α represents the transport phenomena between phases within $\mathrm{d}V$ by diffusion and is related to the gradients of microscopic velocity, temperature and species concentration on each side of the solid/liquid interface $\mathrm{d}A^\alpha$ [24]. Since in the volume-averaging approach that we will follow, the averaged equations from different phases are added within the averaging volume $\mathrm{d}V$, detailed modeling of the interfacial transfer terms I_J^α and I_Q^α can be avoided [19]. The heat, mass or species lost from one phase is gained by other phases, i.e.

$$\sum_\alpha I_J^\alpha = 0 \quad \text{and} \quad \sum_\alpha I_Q^\alpha = 0. \tag{25}$$

For momentum, the interfacial momentum fluxes due to solidification also balance each other, that is $\sum_\alpha I_Q^\alpha = 0$. However, $\sum_\alpha I_J^\alpha$ in the diffused-interface (mushy) zone is modeled using Darcy's law [25]:

$$\sum_\alpha I_J^\alpha = -\frac{\epsilon^{\ell^2}\mu^\ell \langle \boldsymbol{v}^\ell\rangle^\ell}{K(\epsilon^\ell)}, \tag{26}$$

where $K(\epsilon^\ell)$ is the assumed permeability of the diffused-interface (see Assumption 2).

To arrive at a model tractable for computation, the variations of material properties in dV^α are neglected, although globally they may vary, that is $\langle\mu^\alpha\rangle^\alpha = \mu^\alpha$, $\langle k^\alpha\rangle^\alpha = k^\alpha$, $\langle D^\ell\rangle^\ell = D^\ell$, $D^s = 0$. We also assume that all phases in the averaging volume are in thermodynamic equilibrium, i.e. $\langle T^\alpha\rangle^\alpha = \langle T^\beta\rangle^\beta$ for any two phases α and β. The concentration in the averaging volume is assumed to be solutally well mixed, that is, $\langle C_i^\alpha\rangle^\alpha = C_i^\alpha$.

3.2.1. Mass conservation

For the derivation of the volume-averaged equation of mass conservation using Eq. (21), we substitute $\Theta = \rho$, $\mathbf{J} = 0$, and $S = 0$. By writing and adding the individual mass conservation equations for all phases, we obtain:

$$\nabla \cdot \boldsymbol{v} = 0, \tag{27}$$

where we have further defined:

$$\boldsymbol{v} = \epsilon^\ell \langle \boldsymbol{v}^\ell\rangle^\ell. \tag{28}$$

Note that in this work, we have assumed that $\langle \boldsymbol{v}^s\rangle^s = 0$ for all solid phases $s = 2, 3, \ldots, N$.

3.2.2. Momentum conservation

For deriving the averaged equation of momentum conservation from Eq. (21), we take $\Theta = \rho\boldsymbol{v}$ and $S = \mathbf{b}$. Furthermore, we assume a Newtonian fluid and hence the viscous-stress is given in terms of the rate of deformation as,

$$\boldsymbol{\sigma} = -p^\ell I + \mu^\ell[\nabla \boldsymbol{v}^\ell + (\nabla \boldsymbol{v}^\ell)^{\mathrm{T}}]. \tag{29}$$

Using the previous assumptions, and the definition of $\boldsymbol{v}$ given earlier, the final form of the averaged transport equation of momentum transport then yields [25]:

$$\frac{\partial(\rho\boldsymbol{v})}{\partial t} + \nabla\cdot\left(\rho\frac{\boldsymbol{v}\boldsymbol{v}}{\epsilon^\ell}\right) = -\epsilon^\ell\nabla\langle p^\ell\rangle^\ell + \nabla\cdot[\mu^\ell(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^{\mathrm{T}})] - \epsilon^\ell\mu^\ell\frac{\boldsymbol{v}}{K(\epsilon^\ell)} + \epsilon^\ell\rho^\ell\mathbf{g}, \tag{30}$$

where $\mathbf{g}$ is the gravity vector. The change in liquid density is here expressed using the Boussinesq approximation $\rho^\ell = \rho[(1 - \beta_c(C - C_0) - \beta_{\mathrm{T}}(T - T_0)]$ and it appears only in the body force term. In other places, ρ^ℓ is regarded to be the same constant as ρ.

3.2.3. Energy conservation

For deriving the volume-averaged equation of energy conservation we take $\Theta = \rho h = \rho\sum_\alpha \epsilon^\alpha h^\alpha$, where h represents the total enthalpy. In addition, we consider in Eq. (21), $S = 0$ and utilize Fourier's law $\mathbf{J} = -k\nabla T$, with $k \equiv \sum_\alpha \epsilon^\alpha k^\alpha$. Eq. (21) then yields the following:

$$\frac{\partial(\rho h)}{\partial t} + \nabla \cdot (\rho h^{\ell} \boldsymbol{v}) = \nabla \cdot (k \nabla T). \tag{31}$$

We assume constant but unequal specific heat for all phases. Using linear relation between enthalpy and temperature for each phase, a governing equation for energy conservation based on temperature can be derived from Eq. (31) in the following form:

$$\left(\sum_{\alpha=1}^{N} \rho c^{\alpha} \epsilon^{\alpha}\right) \frac{\partial T}{\partial t} + \rho c^{\ell} \nabla \cdot (\boldsymbol{v} T) = \nabla \cdot (k \nabla T) + \sum_{s=2}^{N} \rho L^{s} \dot{\epsilon}^{s}. \tag{32}$$

3.2.4. Species conservation

For arriving at the volume averaged equation of species conservation, we note that for this case in Eq. (21), $\Theta = \rho C_i$, where C_i represents solutal concentration (per unit mass) for component i, and $S = 0$. Furthermore, we utilize Fick's law for species diffusion flux, that is, $\mathbf{J} = -\rho D_i \nabla C_i$. We also assume no diffusion in the solid phase $D_i^s = 0$. The macroscopic transport equation of species conservation can then be derived from Eq. (21) as follows:

$$\frac{\partial C_i}{\partial t} + \nabla \cdot (C_i^{\ell} \boldsymbol{v}) = \nabla \cdot (\epsilon^{\ell} D_i^{\ell} \nabla C_i^{\ell}), \tag{33}$$

where

$$C_i = \sum_{\alpha} \epsilon^{\alpha} C_i^{\alpha}. \tag{34}$$

Notice that C_i^{ℓ} is only defined in the region with liquid fraction greater than zero (the liquid phase and diffused interface). Particularly in the diffused-interface, the following holds $C_i^{\ell} = C_i^s / k_{p_i}^s$, $\forall s = 2, 3, \ldots, N$. Any definition of C_i^{ℓ} in the solid phase will not affect the validity of Eq. (33), since the two terms in Eq. (33) related with C_i^{ℓ} are multiplied with either $\boldsymbol{v}$ (which is zero in the solid) or ϵ^{ℓ} (also zero in the solid). However, for numerical reasons, C_i^{ℓ} is usually taken to be a continuous function across the mushy zone. For example, C_i^{ℓ} is taken as the eutectic concentration for the solid phase in [25]. In this work, we follow the idea of [26] by taking $C_i^{\ell} = C_i^s / k_{p_i}^s$ in the solid phase. The *activity* scalar field is defined following [26] as:

$$a_i(\boldsymbol{x}, t) \equiv \begin{cases} C_i^{\ell}(\boldsymbol{x}, t), & \boldsymbol{x} \in \Omega_t^{\ell}, \\ C_i^{s}(\boldsymbol{x}, t)/k_{p_i}^{s}, & \boldsymbol{x} \in \Omega_t^{s}, \end{cases} \tag{35}$$

where $a_i(\boldsymbol{x}, t)$ is continuous in the whole domain Ω. In particular within the diffused interface, we will have $C_i^{\ell} = C_i^s / k_{p_i}^s = a_i$. So Eq. (33) can be rewritten as

$$\frac{\partial C_i}{\partial t} + \nabla \cdot (\boldsymbol{v} a_i) = \nabla \cdot (\epsilon^{\ell} D_i^{\ell} \nabla a_i), \quad \boldsymbol{x} \in \Omega^{\ell}, \ \ i = 2, 3, \ldots, n, \tag{36}$$

where using the definition of mixture concentration, $C_i = \epsilon^{\ell} C_i^{\ell} + \sum_s \epsilon^s C_i^s$ and Eq. (35), the relation between concentration and activity is written as follows:

$$a_i = \frac{C_i}{\epsilon^{\ell} + \sum_s \epsilon^s k_{pi}^s}. \tag{37}$$

4. Numerical implementation

4.1. Multi-phase motion re-initialization technique

For a solidification system with only one solid phase, only one signed distance function is required since $\phi^s = -\phi^{\ell}$ everywhere. Correspondingly, only one level set equation (usually ϕ^s) needs to be solved. In this work, we consider the possibility of multiple solid phases with $s = 2, 3, \ldots, N$. A signed distance function ϕ^{α} is defined for each phase α. The velocity on the solid–liquid interface is computed using Eq. (16). The velocity on the solid–solid interface is zero, since solid–solid phase transformation is neglected in this work. With interface velocity defined on the boundary of each phase, a set of level set equations (Eq. (10)) can be solved to evolve the phase boundary for each phase. However, evolving each level set equation independently leads to gaps or overlaps of zero level sets due to numerical error. This is a difficult problem addressed in many earlier publications [27,28]. In this work, we use a reinitialization scheme based on the following property of signed distances for a multi-phase system.

Property. *Each point $\boldsymbol{x}$ in the domain Ω belongs to a phase α and has a closest phase β such that $\phi^{\alpha}(\boldsymbol{x}) = -\phi^{\beta}(\boldsymbol{x})$ and $\phi^{\alpha}(\boldsymbol{x}) \leqslant 0 \leqslant \phi^{\beta}(\boldsymbol{x}) \leqslant \phi^{\gamma}(\boldsymbol{x})$, $\forall\gamma$, $\gamma \neq \alpha, \beta$.*

In this work, we use a rather simple but efficient re-initialization scheme with the following three steps:

(1) For each node point $\boldsymbol{x}$, find the smallest two signed distance functions $\phi^{\alpha}(\boldsymbol{x}), \phi^{\beta}(\boldsymbol{x})$ such that $\phi^{\alpha}(\boldsymbol{x}) \leqslant \phi^{\beta}(\boldsymbol{x}) \leqslant \phi^{\gamma}(\boldsymbol{x})$, $\forall\gamma \neq \alpha, \beta$. α will be the phase containing node point $\boldsymbol{x}$ and β will be the nearest phase to $\boldsymbol{x}$.
(2) Compute err $= \frac{\phi^{\alpha}(\boldsymbol{x})+\phi^{\beta}(\boldsymbol{x})}{2}$. For all γ (including α, β), update $\phi^{\gamma}(\boldsymbol{x})$ as $\phi^{\gamma}(\boldsymbol{x}) -$ err.
(3) Use a fast marching technique to independently re-initialize each signed distance function.

This numerical scheme is very fast since the operation time for the three steps is only O(m), O(m) and O($m \log m$), respectively, where m represents the number of nodes in the finite element mesh.

4.2. Fast marching technique

Reinitialization techniques based on structured meshes are described with detail in [12,13]. In an effort to make our developments suitable for coupling with FEM-based solvers for heat transfer, fluid flow and solute transport, we extend the fast marching method to unstructured grids and use it to implement the reinitialization procedure and extension of interface velocity. For convenience, we will only emphasize extension to the positive region $\phi > 0$. Extension of fields to the negative region $\phi < 0$ follows similar methodology.

To simplify the presentation, only two-dimensional problems are addressed here. The fast marching method uses tags to indicate the type of each node: 'alive', 'band' and 'far away', which are shown as black, gray and empty circles, respectively, in Fig. 2. 'Alive' means that the value of a variable (e.g. of the level set variable ϕ or the interface velocity V) is already computed on the node and ready to be extended to other nodes. A node adjacent to 'alive' nodes is ready to have its value extended and is marked as 'band'. Other nodes are simply marked as 'far away'.

A very important idea in the fast marching method is that the level set function values should be extended from the nodes closest to the interface to the nodes farthest from the interface in a well-defined order. In each iteration, the 'band' node with smallest ϕ will have its value extended. Note that when extending ϕ itself, the ϕ value on a 'band' node is just an estimate which is updated whenever one of its neighbor nodes becomes 'alive'. After the closest 'band' node becomes 'alive', its 'far away' neighbors may be captured into 'band'. One continues with this process until all 'band' nodes become 'alive'.

How to find the closest 'band' node as fast as possible for extension of value is very crucial. A simple way is to check all of the nodal points, which leads to a method of operation time O(m) for each iteration with m the number of nodes. The fast marching method uses a balanced heap data structure to store all the 'band' nodes with the nearest node always at root. The balanced heap data structure will lead to a method of operation time only O(log(m_{b})) for each iter with m_{b} the number of 'band' nodes. Details of implementation and algorithms of heap data structures are given in [12,13].

Upwind element. When we estimate value at a 'band' node P, there can be more than one finite element, which has all nodes 'alive' except P (e.g. element (A, B, P) and (B, C, P) shown in Fig. 2). We define as *upwind element* of a 'band' node the element with the smallest average ϕ over all its 'alive' nodes. The upwind element is used to extend value to the 'band' node. For example, if $\phi_{\mathrm{A}} + \phi_{\mathrm{B}} < \phi_{\mathrm{A}} + \phi_{\mathrm{C}}$, then element (A, B, P) will be selected as upwind element to extend value to node P.

Extend values in the fast marching method. Application of the fast marching method involves estimating the level set variable for a node from its 'alive' neighbors. Previous work [13] describes how to apply this

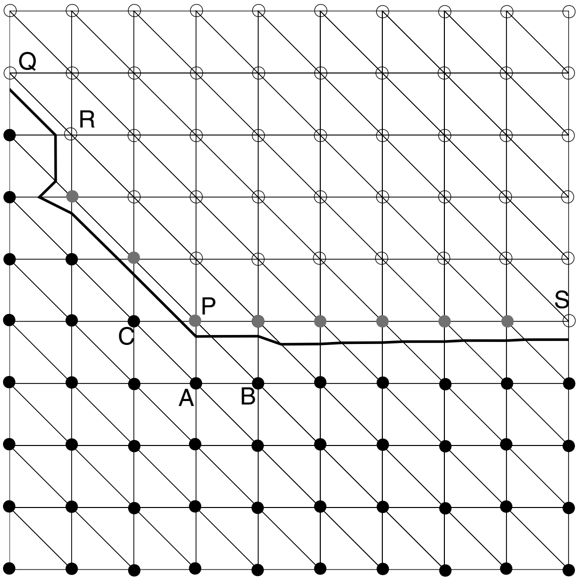


Fig. 2. Fast marching scheme for an unstructured finite element grid in two-dimensions.

extrapolation for structured grids. For unstructured grids, the problem boils down to estimating $\phi_{\rm P}$ on node P from the value at 'alive' nodes of its upwind element (e.g. $\phi_{\rm A}$ on node A and $\phi_{\rm B}$ on node B for the case shown in Fig. 2). In the following, we will derive the formulation of extending values for the two-dimensional case. Suppose, the other two 'alive' nodes of 'band' node P's upwind element are A, B. The normal direction of ϕ, $\nabla\phi = (\phi_{,1}, \phi_{,2})^{\rm T}$, in this upwind element will satisfy:

$$\begin{pmatrix} \Delta x_{\rm A} & \Delta y_{\rm A} \\ \Delta x_{\rm B} & \Delta y_{\rm B} \end{pmatrix} \begin{pmatrix} \phi_{,1} \\ \phi_{,2} \end{pmatrix} = \begin{pmatrix} \Delta\phi_{\rm A} \\ \Delta\phi_{\rm B} \end{pmatrix},$$

where $\Delta\phi_{\rm A} \equiv \phi_{\rm A} - \phi_{\rm P}$, $\Delta x_{\rm A} \equiv x_{\rm A} - x_{\rm P}$ (similarly for other variables). For convenience, we redefine $\phi_{\rm P}$, $\phi_{\rm A}$, $\phi_{\rm B}$ as ϕ, ϕ_1, ϕ_2 and define matrix $\mathbf{M}$ as

$$\mathbf{M} \equiv \begin{pmatrix} \Delta x_{\rm A} & \Delta y_{\rm A} \\ \Delta x_{\rm B} & \Delta y_{\rm B} \end{pmatrix}.$$

In order to calculate ϕ ($\phi_{\rm P}$), we take

$$\begin{pmatrix} \phi_{,1} \\ \phi_{,2} \end{pmatrix} = \mathbf{M}^{-1} \begin{pmatrix} \phi_1 - \phi \\ \phi_2 - \phi \end{pmatrix}$$

with each component $\phi_{,i} = \sum_j (\mathbf{M}^{-1})_{ij}(\phi_j - \phi) = -\sum_j (\mathbf{M}^{-1})_{ij}\phi + \sum_j (\mathbf{M}^{-1})_{ij}\phi_j$. According to the property of signed distance $\|\nabla\phi\| = 1$, we can write:

$$\sum_i \left(\sum_j (\mathbf{M}^{-1})_{ij}\phi - \sum_j (\mathbf{M}^{-1})_{ij}\phi_j \right)^2 = 1. \tag{38}$$

With the geometry-related matrix $\mathbf{M}$ and ϕ_j known, Eq. (38) becomes a quadratic equation for one single variable ϕ. Generally, Eq. (38) has two roots. If fast marching is used in the positive region ($\phi > 0$), then the larger root is used. Otherwise, the smaller root is used. Numerically, if the ϕ value near the interface deviates significantly from signed distance, the above quadratic equation may not attain any real root. Although this did not occur in any of the simulations reported in this paper, for such cases ϕ is taken to be $\frac{\phi_1+\phi_2}{2}$ to make the algorithm robust under any circumstances.

For extension of any other variable (denote as Θ here, e.g. of the interface velocity), we can use the following equation:

$$\nabla\Theta \cdot \nabla\phi = 0. \tag{39}$$

Similarly to extending the level set variable, we can write the following:

$$\begin{pmatrix} \Theta_{,1} \\ \Theta_{,2} \end{pmatrix} = \mathbf{M}^{-1}\begin{pmatrix} \Theta_1 - \Theta \\ \Theta_2 - \Theta \end{pmatrix}.$$

By defining the vector $b_{\Delta\Theta} = (\Theta_1 - \Theta,\ \Theta_2 - \Theta)^{\mathrm{T}}$ and $b_{\Delta\phi} = (\phi_1 - \phi,\ \phi_2 - \phi)^{\mathrm{T}}$, and using Eq. (39), we obtain

$$(\mathbf{M}^{-1}b_{\Delta\phi}, \mathbf{M}^{-1}b_{\Delta\Theta}) = 0 \Rightarrow b^{\mathrm{T}}_{\Delta\phi}\mathbf{M}^{-\mathrm{T}}\mathbf{M}^{-1}b_{\Delta\Theta} = 0.$$

Further by defining $s^{\mathrm{T}} \equiv b^{\mathrm{T}}_{\Delta\phi}\mathbf{M}^{-\mathrm{T}}\mathbf{M}^{-1} = (s_1, s_2)$, the variable Θ at node P can be derived as

$$\Theta = \frac{s_1\Theta_1 + s_2\Theta_2}{s_1 + s_2}.$$

Notice that in Fig. 2, the nodes Q, R and S are not ‘band’ nodes (thus are not stored in heap structure) even if they are neighboring one ‘alive’ node. This is because an upwind element with other two nodes ‘alive’ is required to extend values in our algorithm. Nodes Q, R and S are not captured into ‘band’ until there is at least one neighboring element with two other nodes (in 2D) ‘alive’.

Following the same procedure, the above fast marching techniques including reinitialization and interface velocity extension can easily be extended to three-dimensional problems using a tetrahedral mesh. For example, for extending values (level set variable or interface velocity) from A, B, C to P, the matrix **M** becomes,

$$M \equiv \begin{pmatrix} \Delta x_{\mathrm{A}} & \Delta y_{\mathrm{A}} & \Delta z_{\mathrm{A}} \\ \Delta x_{\mathrm{B}} & \Delta y_{\mathrm{B}} & \Delta z_{\mathrm{B}} \\ \Delta x_{\mathrm{C}} & \Delta y_{\mathrm{C}} & \Delta z_{\mathrm{C}} \end{pmatrix},$$

with $\Delta x_{\mathrm{A}}, \Delta y_{\mathrm{A}}, \ldots$ defined as before. The form of single variable quadratic equation (Eq. (38)) remains the same, except that the summation is done in three dimensions. The formula for extension of velocity becomes

$$\Theta = \frac{s_1\Theta_1 + s_2\Theta_2 + s_3\Theta_3}{s_1 + s_2 + s_3}.$$

4.3. Adaptive meshing technique

4.3.1. Mesh refinement procedure in two-dimensions

A uniform mesh is computationally inefficient since phase boundaries often require finer mesh density. For a given time step, a non-uniform mesh with finer mesh density near phase boundaries can greatly speed up the computation. However, the fact that phase boundaries are moving with time, requires fast adaptive meshing for each time step. In [11], a refinement procedure based on structured grids is used. Each box element, which needs refinement, is divided into a few smaller elements (four elements in 2D and eight elements in 3D). The adapted grid will have so-called ‘hanging’ nodes which appear whenever an element has a neighbor whose refinement level differs by one. Notice that ‘hanging’ nodes make the grid not conforming, thus not very convenient for finite element computation. To ensure continuity of the solution between elements, constraints are enforced for solution at each ‘hanging’ node [11]. In this work, we implemented a technique based on unstructured grids (triangles in 2D and tetrahedrons in 3D), which is more convenient for meshing with irregular geometry. A scheme with little computational cost (proportional to the number of ‘hanging’ nodes) is developed to make the grid conforming by linking the ‘hanging’ nodes in a compatible way.

At first, an initial coarse triangle mesh needs to be generated using any suitable mesh generator. If refinement is desired in a region, a triangle element may be subdivided into four small triangles with the same size and shape by adding middle points of the three edges. If the neighboring element is already refined then middle point of the neighboring edge does not have to be inserted since it is already there.

Information about the level of refinement is stored in a tree data structure, which is capable of tracking sons (refined elements) or ancestors (unrefined element) for each element. The refinement level of an element is defined according to its position in the tree data structure. Elements at the root (in the initial coarse mesh) have refinement level 0. Elements which are generated by refining elements with refinement level n are assigned refinement level $n + 1$. Refinement level of the mesh is defined as the maximum refinement level of elements in the tree data structure. A typical sequence of refinement is shown in Fig. 3, along with the associated data structure.

In general, the mesh will not be suitable for finite element analysis after refinement, since middle points may exist on element edges as shown in the refined mesh in Fig. 3. In order to make the mesh conforming, two more steps are taken as follows:

(1) Elements are further refined so that the refinement level for each two neighboring elements differs by at most 1 as shown in step (1) of Fig. 4.
(2) An element neighboring another element with higher refinement level is subdivided into a few connecting elements by connecting its nodes and middle points on edges as shown in step (2) of Fig. 4.

Using this adaptive meshing technique, the quality of the triangles (note that elements with all angles equal to 60° are considered to be of the best quality) will be exactly the same as the input mesh since the refinement does not change the element shape. This, however, is not the case for the connecting elements (elements generated by linking the hanging nodes). Moreover, linear interpolation of data only needs to be done for newly inserted middle points by just taking the average of data on end points. These advantages make the present adaptive meshing technique very suitable for transient problems.

4.3.2. Mesh refinement procedure in three-dimensions

The basic mesh refinement procedure in three-dimensions is similar to that used earlier in two-dimensions. However, there are a few additional challenges.

Each tetrahedral element is divided into eight child elements. The shape of child elements is not preserved as shown in Fig. 5. Moreover, there are three options to divide a tetrahedron into eight smaller tetrahedrons. We compare the length of the line segments 4–9, 5–8, 6–7 (Fig. 5), and choose the option corresponding to the shortest length. This ensures that the sum of edge lengths is minimized in refinement and consequently, the mesh quality is maintained.

As in two-dimensions, linking elements with ‘hanging’ nodes on some of the middle of their edges are generated after the first procedure of refinement. In order to make the mesh conforming, it is necessary to link the ‘hanging’ nodes in a compatible way such that (1) each linking element is divided into a few smaller ones without ‘hanging’ nodes as shown in Fig. 6, (2) the common faces of neighboring elements are divided in the same way to avoid the error shown in Fig. 7.

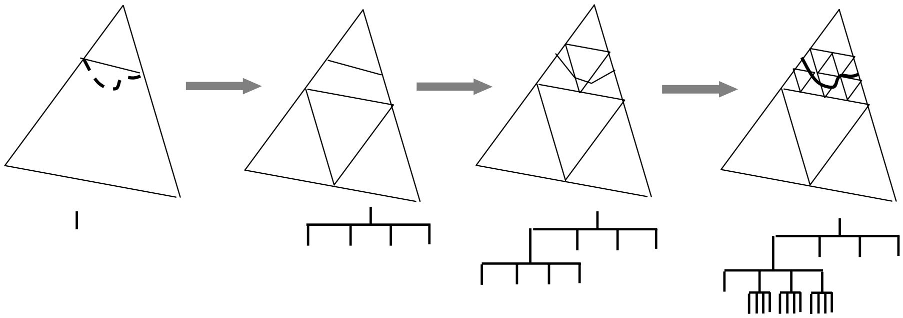


Fig. 3. Schematic of the refinement procedure for unstructured grids in two-dimensions. The broken line refers to the interface to be represented. Continuous line demonstrates the linear representation of the interface using the adaptive mesh.

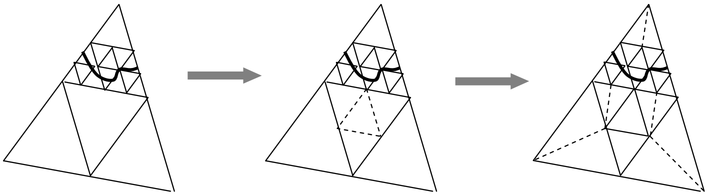


Fig. 4. Schematic of generating a conforming grid in two-dimensions.

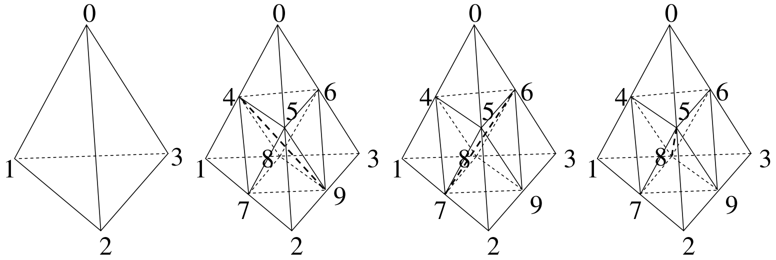


Fig. 5. Three options of refining a tetrahedral element in 3D.

4.3.3. Mesh refinement criterion

Different criterions can be used to determine whether an element needs to be further refined or not. For example, in [29] a refinement criterion was used based on the temperature gradient within the element e as follows:

$$\mathrm{Error}(e) \equiv \int_{\Omega^e} \|\nabla T\| \, \mathrm{d}\Omega < \mathrm{tol}_T. \tag{40}$$

Similarly, a criterion based on solute gradient can also be used as follows:

$$\mathrm{Error}(e) \equiv \int_{\Omega^e} \|\nabla C\| \, \mathrm{d}\Omega < \mathrm{tol}_C. \tag{41}$$

In this work, since we can determine how far away a node point is from an interface based on the computed signed distances, we select a refinement criterion according the signed distances as follows:

$$h^e < \max\left(\frac{|\phi|}{4}, w\right), \tag{42}$$

where h^e is a characteristic length of the element computed as $\sqrt{\mathrm{area}(e)}$, $|\phi|$ is the minimum distance to the interface of the element computed from the value of signed distances on each element node. Notice that

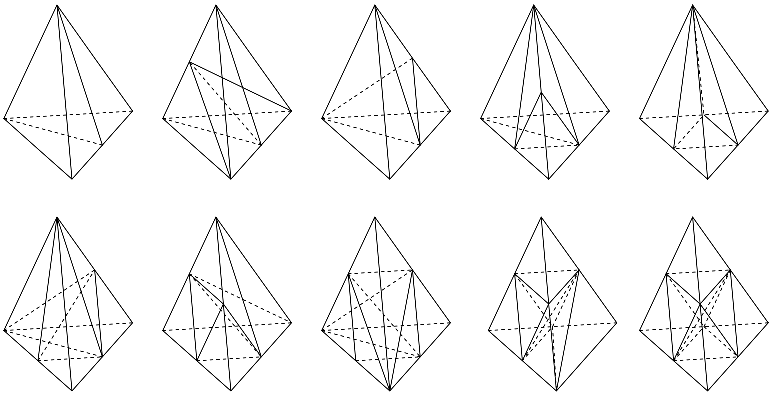


Fig. 6. Various types of linking element in 3D.

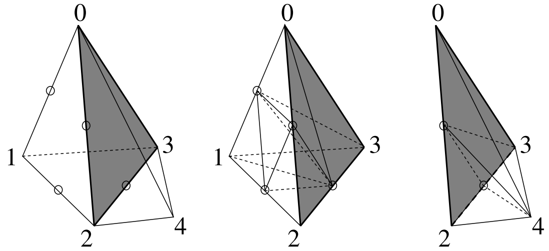


Fig. 7. Incompatible neighboring elements in three dimensions. Although both neighboring elements (0–1–2–3 and 0–2–3–4) are correctly divided into a few smaller elements by linking the 'hanging' nodes, the common face of 0–2–3 is divided into different ways for these two neighboring elements.

the value of w is preselected as a spatial discretization parameter. Based on this value of w and level set field ϕ, the mesh is refined with the refinement criterion to be sure that (1) fine mesh is obtained when near the interface and coarse mesh is sustained when far away from the interface, and (2) at least one or a few elements are contained within the artificial diffused-interface (mushy) zone. w serves as a lower bound for h^e in Eq. (42), since generally two elements in the artificial mushy zone are enough. If more elements (e.g. four elements) are desired to numerically resolve the artificial mushy zone, w in Eq. (42) needs to be replaced with a smaller value (e.g. $\frac{w}{2}$, so that the criterion in Eq. (42) becomes $h^e < \max(\frac{|\phi|}{4}, \frac{w}{2})$).

4.3.4. Narrow band computation

Adaptive meshing can drastically reduce the number of degrees of freedom, especially for computation of heat, momentum and solute transport. However, for the level set evolution, adaptive mesh computation on the whole domain is not very efficient. Level set computation can be performed on a narrow band near the interface, instead of on the whole domain [12,13]. An example of narrow band computing mesh (for level set only) is shown in Fig. 8.

4.4. Adaptive time step technique

For many dendritic solidification growth problems, initial conditions are given before the formation of thermal and solutal boundary layers. The initial interface velocity is large in comparison to the interface velocity after the build up of thermal and solutal boundary layers. Correspondingly, a small time step should be used initially. However, such small time step is not required in later stages. In order to speed up the computation, adaptive time step technique is adopted in this work. We select the time step size according to a CFL-type criterion of the form

$$\Delta t^{n+1} = \lambda_{\mathrm{CFL}} w / \max(|V^n_{\mathrm{Interface}}|), \tag{43}$$

where $\max(|V^n_{\mathrm{Interface}}|)$ is the maximum interface velocity at all phase boundaries. λ_{CFL} is a coefficient controlling the accuracy, and unless is otherwise stated it is selected as 0.6 in all numerical examples. Eq. (43) guarantees that the interface advances about $0.6w$ in each time step at places with maximum interface velocity (normally at tips).

Note that in all of the subproblems including fluid flow, heat transfer, solute transport and level set evolution, fully-implicit backward Euler scheme is adopted for time integration in order to achieve unconditionally stability regardless of the time step size. The selection of the time step size is mainly based on solution accuracy requirements and is not related with stability issues [16].

4.5. Finite element implementation

In this work, the adaptive meshing technique generates triangle/tetrahedral type unstructured grids. The finite element technique is thus adopted for computations including fluid flow, heat transfer, solute transport and level set evolution. For heat transfer and solute transport, the classical SUPG stabilizing technique is used

$$B_{\mathrm{stab}}(\boldsymbol{W}, \boldsymbol{V}) = L_{\mathrm{stab}}(\boldsymbol{W}), \tag{50}$$

where

$$B_{\mathrm{stab}}(\boldsymbol{W}, \boldsymbol{V}) = B(\boldsymbol{W}, \boldsymbol{V}) + \int_{\Omega} \mathscr{F}(\boldsymbol{v}, p) \cdot \mathscr{G}(\boldsymbol{w}, q)\, \mathrm{d}\Omega,$$

$$L_{\mathrm{stab}}(\boldsymbol{W}) = L(\boldsymbol{W}) + \int_{\Omega} \left\{ \frac{p}{\epsilon^{\ell}} \nabla \epsilon^{\ell} - \epsilon^{\ell} \rho [\beta_c (C - C_0) + \beta_T (T - T_0)] \boldsymbol{g} \right\} \cdot \mathscr{G}(\boldsymbol{w}, q)\, \mathrm{d}\Omega.$$

The following definitions have been introduced in the above equation:

$$\mathscr{F}(\boldsymbol{v}, p) = \rho \frac{\partial \boldsymbol{v}}{\partial t} + \frac{\rho \boldsymbol{v}_* \cdot \nabla \boldsymbol{v}}{\epsilon^{\ell}} + \nabla p + \frac{\epsilon^{\ell} \mu^{\ell}}{K(\epsilon^{\ell})} \boldsymbol{v} - \nabla \cdot [\mu^{\ell} (\nabla \boldsymbol{v} + \nabla^{\mathrm{T}} \boldsymbol{v})], \tag{51}$$

$$\mathscr{G}(\boldsymbol{w}, q) = \tau_1 \frac{\boldsymbol{v}_* \cdot \nabla \boldsymbol{w}}{\epsilon^{\ell}} - \frac{\tau_2 \epsilon^{\ell}}{K(\epsilon^{\ell}) \rho} \boldsymbol{w} - \frac{\tau_3}{\rho} \nabla \cdot [\mu^{\ell} (\nabla \boldsymbol{w} + \nabla^{\mathrm{T}} \boldsymbol{w})] + \tau_4 \nabla q \tag{52}$$

with $\boldsymbol{v}_*$ a divergence-free velocity obtained from the previous iteration. τ_1, τ_2, τ_3 and τ_4 are parameters for the advection (SUPG), Darcy drag (DSPG), diffusion and pressure (PSPG) stabilizing terms. The selection of their values is discussed in [25,30].

4.5.2. Stabilized finite element method for level set computation

The weak form of the level set equation (Eq. (10)) is

$$\int_{\Omega} \delta\phi^{\alpha} (\phi^{\alpha}_t + V^{\alpha} |\nabla \phi^{\alpha}|)\, \mathrm{d}\Omega = 0, \tag{53}$$

where $\delta\phi^{\alpha}$ is a test function in C^0. To avoid oscillations, GLS and diffusion stabilizing term are introduced as shown in the following formulation [31]:

$$\sum_{e=1}^{N_{el}} \int_{\Omega^e} \left[\delta\phi^{\alpha} + \left(\tau^e_{\phi^{\alpha}} \frac{V^{\alpha} \nabla \delta\phi^{\alpha} \cdot \nabla \phi^{\alpha}}{|\nabla \phi^{\alpha}|} \right) \right] (\phi^{\alpha}_t + V^{\alpha} |\nabla \phi^{\alpha}|) + (\upsilon \nabla \delta\phi^{\alpha} \cdot \nabla \phi^{\alpha})\, \mathrm{d}\Omega^e = 0, \tag{54}$$

which further leads to the following semi-discretized form:

$$(M_{\phi^{\alpha}} + M_{\mathrm{GLS}}) \dot{\phi}^{\alpha} + f^{\alpha}_{\phi} + f_{\mathrm{GLS}} + f_{\mathrm{SC}} = 0,$$

$$M^{\alpha}_{\phi} = \sum_{e=1}^{N_{el}} \int_{\Omega^e} N^{\mathrm{T}} N\, \mathrm{d}\Omega^e, \quad M_{\mathrm{GLS}} = \sum_{e=1}^{N_{el}} \int_{\Omega_e} (\nabla N^{\mathrm{T}} \cdot \frac{\nabla \phi^{\alpha}}{|\nabla \phi^{\alpha}|} V^{\alpha}) \tau^e_{\phi^{\alpha}} N\, \mathrm{d}\Omega^e,$$

$$f^{\alpha}_{\phi} = \sum_{e=1}^{N_{el}} \int_{\Omega^e} N^{\mathrm{T}} V^{\alpha} |\nabla \phi^{\alpha}|\, \mathrm{d}\Omega^e, \quad f_{\mathrm{GLS}} = \sum_{e=1}^{N_{el}} \int_{\Omega^e} \left(\nabla N^{\mathrm{T}} \cdot \frac{\nabla \phi^{\alpha}}{|\nabla \phi^{\alpha}|} V^{\alpha} \right) \tau^e_{\phi^{\alpha}} V^{\alpha} |\nabla \phi^{\alpha}|\, \mathrm{d}\Omega^e,$$

$$f_{\mathrm{SC}} = \sum_{e=1}^{N_{el}} \int_{\Omega^e} \upsilon \nabla N^{\mathrm{T}} \cdot \nabla \phi^{\alpha}\, \mathrm{d}\Omega^e.$$

In our computation, the GLS stabilize term parameter τ^e_{ϕ} is selected as $10^{-3} \frac{h^e}{|V^{\alpha}|}$. Since sharp corners often exist near triple points, we use a larger diffusion stabilizing term at triple points than at other places:

$$\upsilon = \begin{cases} 10^{-4} h^e & \text{for triple points } (\exists \alpha \neq \beta \neq \gamma, \text{ s.t. } \epsilon^{\alpha}, \epsilon^{\beta}, \epsilon^{\gamma} > 0), \\ 10^{-6} h^e & \text{otherwise.} \end{cases} \tag{55}$$

4.6. Overall solution procedure

The various subproblems considered here include the thermal, flow, solute species problems as well as the interface evolution. The tolerance level used to define convergence in all four main solution steps (heat, solute, momentum and level set) is set at 10^{-12}. The error criterion is based on the relative error in the solutions obtained between Newton–Raphson iterations within a time step. For example, in the fluid flow solver, the error norm is defined as $\|\Delta U^{i+1}\|/\|U_n^{i+1}\|$ where $U = [\boldsymbol{v}, p]$. The overall algorithm is summarized below:

(1) At time t_n, all fields such as velocity $\boldsymbol{v}_n$, temperature T_n, concentrations C_{i_n}, level set variables ϕ_n^α, etc. are known.
(2) Determine Δt (adaptive time step) and generate the new mesh for this time level (adaptive meshing).
(3) Advance to time step $t_{n+1} = t_n + \Delta t$. Set $j = 0$, $\boldsymbol{v}_{n+1}^{j=0} = \boldsymbol{v}_n$, $T_{n+1}^{j=0} = T_n$, $C_{i_{n+1}}^{j=0} = C_{in}$, $\phi_{n+1}^{\alpha j=0} = \phi_n^\alpha$, etc., where j is an iteration index and the subscript denotes the time level.
(4) Start an inner loop to solve the coupled system till converged.
  (a) Compute velocity at phase boundaries, use fast marching to extend velocity to the whole domain.
  (b) Solve the level set equation for all phases to get the signed distance functions $\phi^{\alpha^j_{n+1}}$.
  (c) Solve the energy equation to obtain T^j_{n+1}.
  (d) Solve the solute equations to obtain $C^j_{i_{n+1}}$ for each component i.
  (e) Solve the momentum equation to obtain $\boldsymbol{v}^j_{n+1}$.
  (f) $j = j + 1$. If not converged, return to (a).
(5) Use the re-initialization scheme to remove gaps or overlapping between ϕ^α for different phases and ensure that all ϕ^α are signed distance functions.

5. Numerical examples

5.1. Crystal growth in an undercooled binary alloy

This example was originally addressed in [4] using a front-tracking method. The material parameters defining the problem are density $\rho^s = \rho^\ell = 1$, heat capacity $c^s = c^\ell = 1$, heat conductivity $k^s = k^\ell = 1$, latent heat $L = 1$, solute diffusivity $D^\ell = 0.1$, $D^s = 0$, liquidus slope $m =- 0.035$ (so that $T^{s\ell}(C^\ell) = mC^\ell$) and partition coefficient $k_\mathrm{p} = 0.312$. On the freezing interface $\Gamma^{s\ell}$, we consider the Gibbs–Thomson relation given in Eq. (6) with $\varepsilon_\mathrm{c} = \varepsilon_\mathrm{V} = -0.002$.

At time zero, we consider a small solid seed in the middle of the computational domain. Its geometry is described as follows:

$$\begin{aligned} x(s) &= (R + P\cos(8\pi s))\cos(2\pi s), \\ y(s) &= (R + P\cos(8\pi s))\sin(2\pi s), \end{aligned} \tag{56}$$

where $R = 0.1$ and $P = 0.02$. The initial concentration of the seed and the undercooled melt are $k_\mathrm{p}C_0$ and C_0, respectively with $C_0 = 2.2\%$. The initial temperature of the seed and the undercooled melt are taken as $T^{s\ell}(C_0)$ and $T^{s\ell}(C_0) - 0.5$, respectively.

Fluid flow is not considered in this example. The computational domains for the temperature and solute fields are taken as $[-10, 10]\times[-10, 10]$ and $[-2, 2]\times[-2, 2]$, respectively. Insulated boundary conditions are considered for both temperature and solute at all sides of the respective computational domains.

Based on the very coarse input mesh shown in Fig. 9a, an adaptive mesh is generated in each time level with a part of it used for the calculation of the solute field and the whole mesh used for temperature calculations as shown in Figs. 9c and d, respectively. At $t = 1.0$, the effect of the domain boundary on the temperature field is small since the temperature boundary layer is far away from the boundary as shown in Fig. 9f. The obtained micro-segregation pattern with maximum concentration about 5.0% is shown in Fig. 9e and it compares well with the results obtained in [4] using a sharp-interface (front-tracking) model. Comparison of the interface position at $t = 1$ with the results given in [4] is shown in Fig. 9b.

We also tested the effects of mesh orientation by rotating the initial crystal by 45°. The mesh for solute transport colored with solute concentration at time about 0.6 is shown in Fig. 10b. The similarity between Figs. 10a and b indicates that the obtained numerical results do not depend on the mesh orientation.

Using mesh refinements based on the level set, temperature gradient (tolerance $= 2\times10^{-5}$) and concentration gradient (tolerance $= 1\times10^{-5}$), the meshes shown in Fig. 11 were generated with a quarter of each mesh shown for easy comparison. It can be seen that all meshes provide similar results for the interface position,

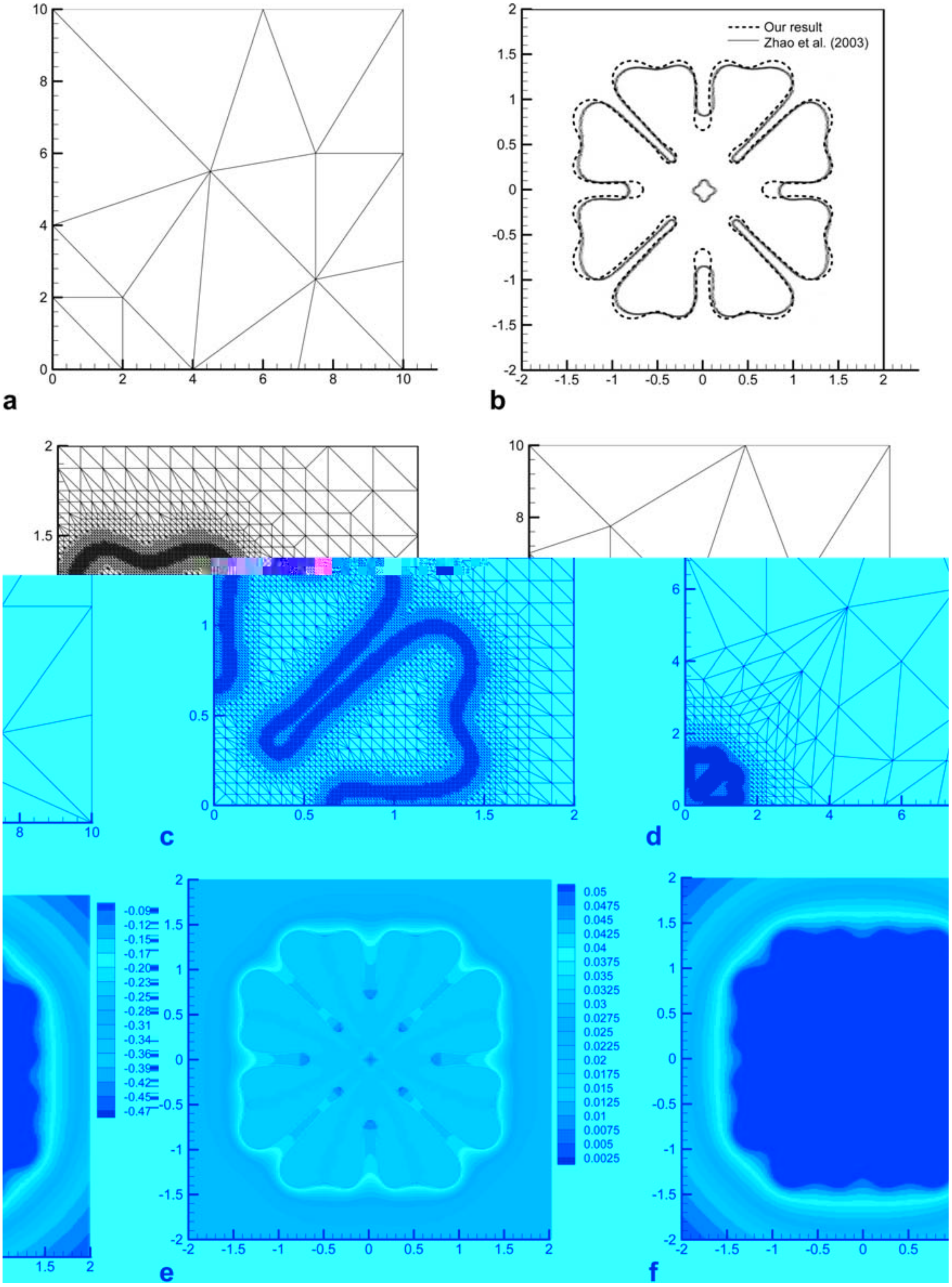


Fig. 9. (a) 1/4 of the input mesh. (b) The interface at $t = 1.0$. (c) 1/4 of the mesh used for solute computation ($t = 1.0$). (d) 1/4 of the mesh used for temperature computation ($t = 1.0$). (e) Solute concentration ($t = 1.0$). (f) Temperature ($t = 1.0$).

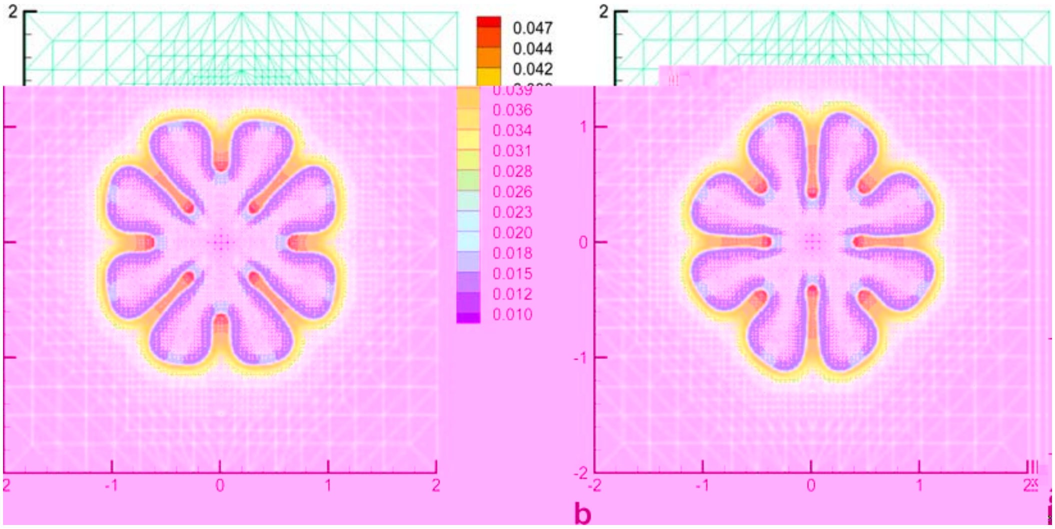


Fig. 10. Mesh for solute transport at time about 0.6 colored with the concentration field: (a) without crystal rotation; (b) with crystal rotation.

except that the interface position generated using the ∇T-based mesh criterion is a bit flattened at the dendrite 'valleys'. This is due to the small temperature gradient and correspondingly low mesh density at this location. With the criterion based on ∇C, many more elements were generated at the 'valleys'. This suggests that the criterion based on ∇T is not as good as the criterion based on ∇C or ϕ. The mesh generated with the criterion based on ∇C is very similar to the mesh generated with the criterion based on ϕ, except that it results in a higher mesh density at the middle of the crystal due to micro-segregation within the solid crystal.

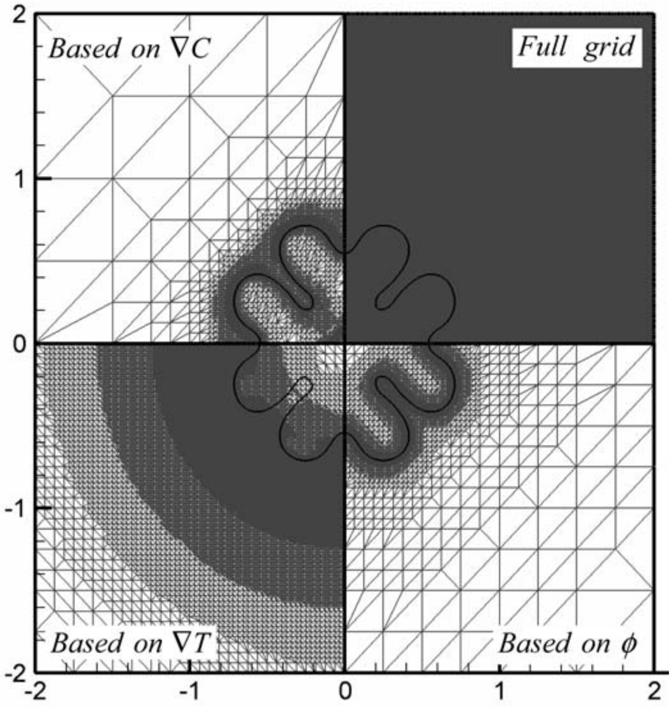


Fig. 11. Mesh generated using different mesh-refinement criteria.

In the above benchmark problem, we compared our results with the results obtained using a front tracking method. To allow a comparison with the phase field method for further validation of our algorithm, we briefly consider a problem similar to the one examined in [32] by choosing material properties $k_s = k_l = 1$, $c_s = c_l = 1$, $L = 1$, $\rho = 1$, $k_{\mathrm{p}} = 0.15$, $m_l = -1$, $C_0 = \frac{0.07}{1-0.15}$, $D_l = \frac{1}{40}$ and undercooling 0.55. We write the values of C_0 and D_l in this unfinished form, so that the reader can easily refer to both papers for the connection between the two models. The initial crystal used in our computation is a circular seed with radius 80 and without perturbation. The anisotropy in the Gibbs–Thomson relation is defined as $T^* = -d_0(1 - 15\varepsilon\cos 4\theta)\kappa$, $d_0 = 1$, $\varepsilon = 0.02$, where θ is the angle between the interface direction and the x-axis. We use an initial grid of 1 × 1 and 9 refinement levels to discretize a domain of 1500 × 1500. The results of this example are shown in Fig. 12. To facilitate comparison between the two models, we provide the computed dimensionless temperature at the dimensionless time 280,000. However, for concentration shown in the results, we use the concentration C instead of the definition in [32]. This is because the definition of dimensionless concentration in [32] requires a phase field variable, which is not applicable in our model. Comparing our results with those in Fig. 1 of [32], we note that the tip radius and position are practically the same. The solute boundary layer is also found to be much thinner than the thermal boundary layer due to the large Lewis number. However, there are some small differences of the interface shape at location around (300, 300), where perturbations are about to develop.

5.2. Planar/cellular/dendritic transition

The relationship between the solidification growth velocity and the primary arm spacing described by the Mullins–Sekerka theory has been numerically studied using phase field methods [33,34]. In these studies, the temperature gradient is assumed to be constant during directional solidification and the energy equation is not solved. Instead, the temperature field is obtained by moving the initial temperature field with a given velocity.

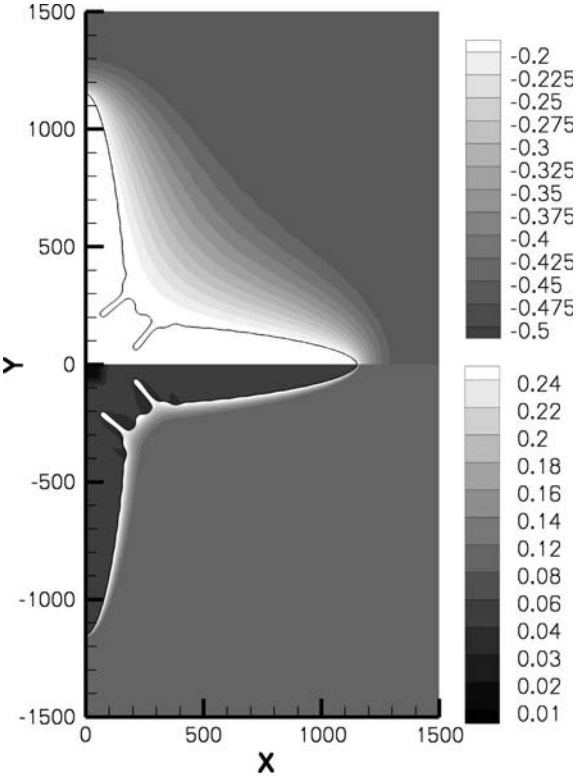


Fig. 12. The solid/liquid interface shape at time 280,000. The upper and lower boxes show the temperature and solute concentration fields, respectively.

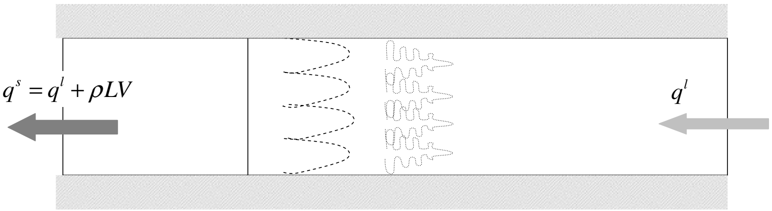


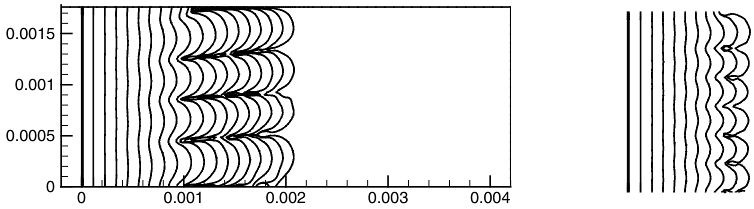
Fig. 13. Problem definition for study of planar/cellular/dendritic transition.

In this example, the simulations are performed under similar conditions to those of [33,34]. However, the simplifications on the temperature field are not taken here. The framework shown in Fig. 13 is adopted for a Ni–Cu alloy with the following important parameters [33,34]: melting temperature $T_m = 1728$ K, liquidus slope $m = 357$ K/at. frac., initial concentration $C_0 = 0.40831$ at. frac., latent heat $L = 2350$ J/cm^3, solute diffusivity $D^\ell = 10^{-5}$ cm^2/s, surface tension $\sigma = 3.7 \times 10^{-5}$ J/cm^2, Gibbs–Thomson coefficient $\varepsilon_c = \sigma T_m / L = 2.7207 \times 10^{-5}$ cm K, temperature gradient $G = q^\ell / k = 2.15 \times 10^4$ K/cm, partition coefficient $k_p = 0.86$, heat capacity $c = 0.46 \times 10^3$ J/kg K, density $\rho = 8.880 \times 10^{-3}$ kg/cm^3, heat diffusivity $k = 6.07 \times 10^{-1}$ W/cm K, and domain size is [$(1.76 \times 10^{-3}$ cm$) \times (4.40 \times 10^{-3}$ cm$)$]. No initial perturbation is introduced (an initially planar solid–liquid interface is assumed). The numerical error acts as a perturbation that leads to unstable growth pattern for the four considered growth velocity values (0.1 cm/s, 0.2 cm/s, 0.4 cm/s and 0.8 cm/s) as shown in Fig. 14.

We also plotted the obtained wavelengths for different growth velocities in Fig. 15 together with $\lambda^2 V = 1.6 \times 10^{-14}$ m^3/s and the MS (Mullins–Sekerka) loop. The MS loop is the zero contour plot of $\frac{\dot{\delta}}{\delta}$, where δ is the perturbation amplitude with wavelength λ [35]. Within the MS loop, $\frac{\dot{\delta}}{\delta} > 0.$, the perturbation with wavelength λ increases and therefore the interface is unstable. Outside the MS loop, $\frac{\dot{\delta}}{\delta} < 0$ and the perturbation with wavelength λ damps out. Planar interfaces are obtained with growth velocity 0.01 cm/s and 10 cm/s as demonstrated in Fig. 16. The upper limit of solidification speed for cellular growth is the absolute stability, V_a, velocity. Its approximate value is given in [36] as $V_a \approx \frac{-m_l C_0 (1-k_p) D}{k_p^2 \Gamma}$, which is about 10 cm/s in this problem.

In the above computation, we use an initial mesh of 50 × 20. Two levels of refinement for velocity 0.01 cm/s, 0.1 cm/s, 0.2 cm/s and three levels of refinement for velocity 0.4 cm/s, 0.8 cm/s, 10 cm/s are performed using the adaptive meshing technique. A typical mesh colored with solute concentration is shown in Fig. 17.

The computed solute boundary layer in front of the dendrite tips is shown in Fig. 18. Using the method discussed in [36], the equivalent solute boundary layers shown in Fig. 18 are about 2.2×10^{-4} cm,



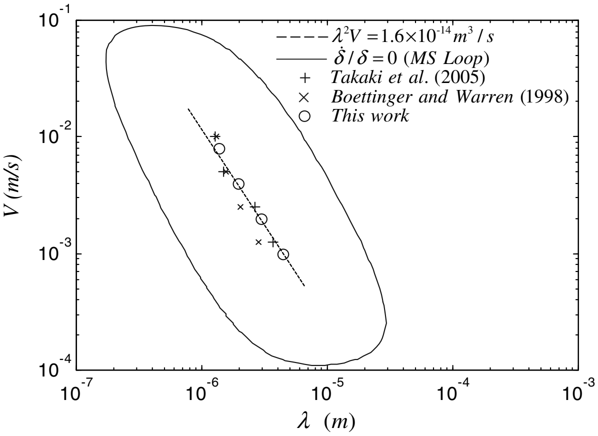


Fig. 15. Relationship between the solidification speed and arm spacing.

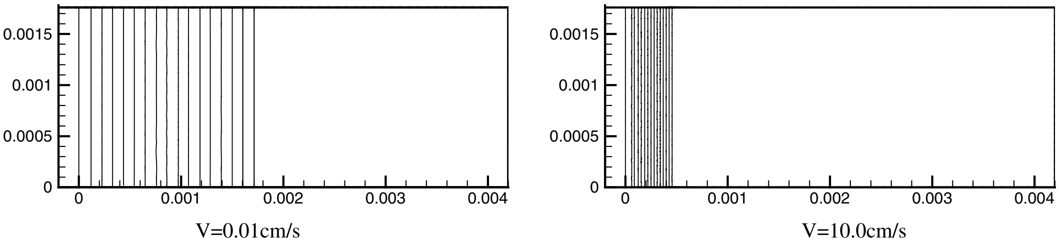


Fig. 16. Planar growth at small and large velocities (axes unit: cm).

1.2×10^{-4} cm, 0.6×10^{-4} cm and 0.3×10^{-4} cm for growth velocity values of 0.1 cm/s, 0.2 cm/s, 0.4 cm/s and 0.8 cm/s, respectively. These values agree with the analytical approximation of the solutal boundary layer ($\approx 2D_l/V \propto 1/V$).

For the initial solute concentration $C_0 = 0.40831$, we only observed planar and cellular growth patterns. The surface tension and thermal gradient prevent the transition from cellular to dendritic. Generally, the growth pattern is less stable for material with smaller surface tension, smaller partition coefficient and higher concentration. In this case, the surface tension and partition coefficient are assumed to be constant. In order to capture the dendritic growth pattern of Ni–Cu alloy, we increase the copper concentration to 0.6. With growth speed 0.2 cm/s, we observed three solidification stages: (1) planar growth; (2) cellular growth; and (3) dendritic growth (Fig. 19). Initially, the growth pattern is planar. Then about six cells are formed. Due to competition,

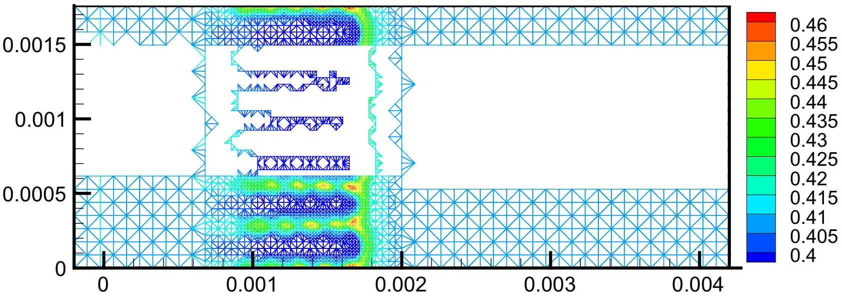


Fig. 17. Adaptive mesh colored with solute concentration.

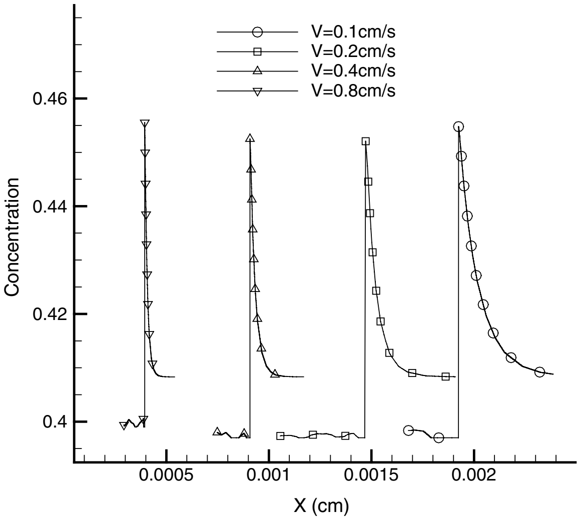


Fig. 18. Solute boundary layer for various growth velocities.

only three major cells are left in the computational domain. Secondary branches are observed in the segregation pattern (Fig. 19).

5.3. Alloy solidification with multiple solid phase growth

For many alloy solidification systems, multiple solid phases are formed. Eutectic and peritectic are the most common solidification microstructures with multiple solid phases. These solidification systems were addressed earlier using multi-phase field models with a comprehensive review given in [7]. Here we present the results obtained with the present level set model for an eutectic growth problem [37] and a peritectic growth problem [38].

5.3.1. Eutectic growth

For directional eutectic solidification with two solid phases α and β, we use the same framework shown in Fig. 13. A temperature gradient $G = 50$ K/cm is maintained at the right side and a cooling rate of $R = 0.005$ K/s is applied at the left side. Initially, a few β seeds are embedded in the α phase. Other important parameters are [37]: liquidus slops $m^{\alpha} = -10$ K/at.% and $m^{\beta} = +5$ K/at.%, partition coefficients $k_{\mathrm{p}}^{\alpha} = 0.1$ and $k_{\mathrm{p}}^{\beta} = 0.05$, diffusion coefficient $D^{\ell} = 10^{-5}$ cm^2/s, surface energy $\sigma = 10^{-5}$ J/cm^2, eutectic composition $C^{\mathrm{eut}}/C^{\beta} = 0.2$ with $C^{\beta} = 5$ at.%, latent heats $L^{\alpha} = L^{\beta} = 1000$ J/cm^3, heat capacity $c = 1 \times 10^3$ J/kg K, density $\rho = 1 \times 10^{-2}$ kg/cm^3, heat diffusivity $k = 1.0$ W/cm K, eutectic temperature $T_{\mathrm{e}} = 1000$ K and finally domain size [0.4 mm × 0.12 mm].

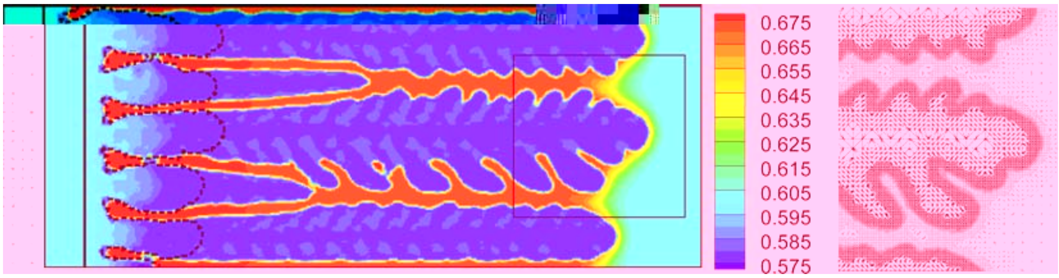


Fig. 19. Solute concentration and adaptive mesh near tips of dendritic growth at copper concentration 0.6 at. frac.

By putting initially four β seeds (red region shown in Fig. 20) into the α phase (blue region shown in Fig. 20), the eutectic growth follows a stable pattern as shown in Fig. 20 with a stable growth wavelength of about 30 μm. In [37], the stable growth wavelength reported was about 25 μm. If only two β seeds are inserted initially in the α phase, then the eutectic growth becomes unstable. Valleys with high solute concentration will form in the middle of the α–ℓ phase interface as shown in Fig. 21. This is because with only two β seeds, the solute rejected during the transformation of the liquid to the α solid phase cannot be easily absorbed by the β phase. The accumulation of solute will further slow down the growth rate of the α phase and form valleys. Since the high solute concentration favors the growth of β phase, the width of the β phase increases as shown in Fig. 21. This unstable eutectic growth pattern for fewer β phase seeds was previously pointed out in [37]. If ten β seeds are inserted initially in the α phase, half of the seeds stop growing after the tips are about 6 μm away from the left boundary. The growth of the remaining five seeds shows an oscillatory pattern. Finally, one more seed stops growing. The remaining four β seeds adjust their wavelength to about 30 μm as shown in Fig. 22.

5.3.2. Peritectic growth

Fe–0.3 wt%C alloy is considered with the following parameters: domain size 150 μm × 700 μm, cooling rate $R = -3$ K/s applied at the bottom, thermal gradient $G = 140$ K/cm maintained at the top, diffusion coefficient $D_l = 3 \times 10^{-5}$ cm^2/s and surface tension $\sigma = 2.04 \times 10^{-4}$ J/cm^2. The thermodynamic data are taken from the Fe–C phase diagram [38]. Initially, an initial small crystal is put in the bottom-left corner. The initial grid size is 3 × 14 with five levels of refinement. During the growth of the ferrite the redistribution of carbon leads to morphological instability and a dendrite evolves. Below the peritectic temperature the austenitic seeds are generated by nucleation. The austenite grows around the undercooled part of the ferritic dendrite as shown in Fig. 23. Before time 1.6 s, the solute concentration is less than 0.51 wt% and thus only the ferritic phase is formed. As the solute concentration accumulates to be more than 0.51 wt% behind the dendrite tips, the austenitic phase is nucleated and formed around the secondary dendrite arms. Table 1 provides a comparison of the computed results with those reported in [38].

5.4. Ternary alloy with melt convection effects

Crystal growth for pure materials under convection has been well studied using phase field methods [17,18,39], front tracking methods [2] and level set methods [16]. In these studies, the growth of only one crystal is investigated with a prescribed inlet flow velocity. In [40], Ostwald ripening of Al–Cu particles in

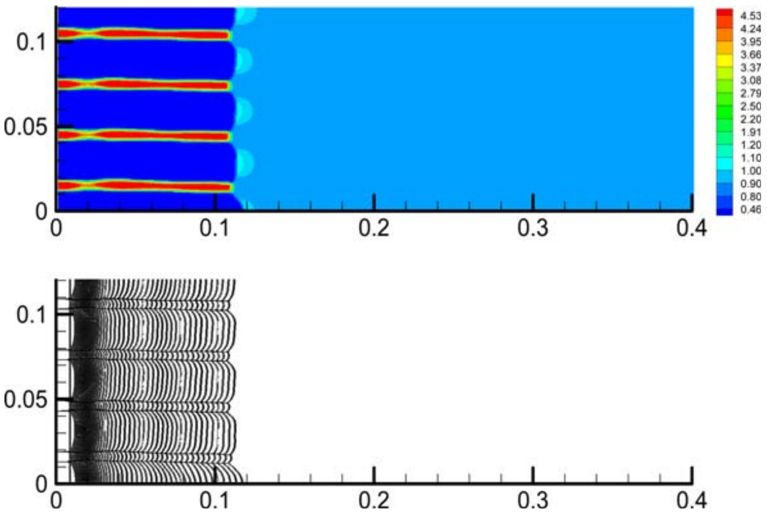


Fig. 20. Solute concentration at time about 80 s and the evolution of the interface position (with four β seeds). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

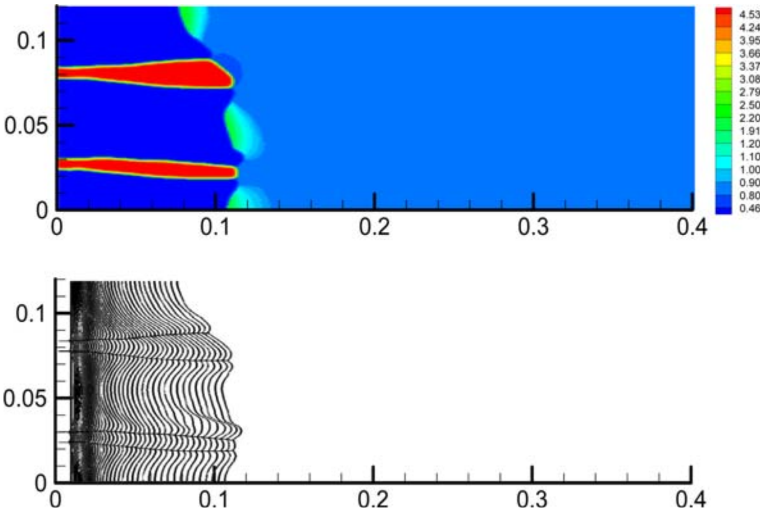


Fig. 21. Solute concentration at time about 80 s and the evolution of the interface position (with two β seeds).

the presence of melt convection was studied using the phase field method with a flow driven by a constant pressure drop. Fully coupled dendritic alloy solidification growth with heat, solute and momentum transport is computationally very difficult due to the various length scales involved. In order to resolve the dendrite tip or solute boundary layer, a small grid spacing is required. For the study of crystal growth of pure materials with a prescribed flow as in [2,16–18,39], a computation domain in the size of the thermal boundary layer is required. Developing substantial buoyancy driven flow requires even larger domain and correspondingly larger grid node/element number. The example in this section is the first study fully coupling dendritic growth with heat, solute and momentum transport for alloy solidification.

In this example, we consider a Ni based alloy with 5.8 wt%Al and 15.2 wt%Ta. The purpose of this example is to demonstrate the ability of the algorithms presented in this paper to couple solute diffusion of multiple components with convection. Only one solid phase is considered. Thermodynamic and transport properties for this alloy are obtained from [41] based on the one solid phase simplification.

Initially a cavity of size 0.025 m × 0.025 m is filled with alloy of Ni–5.8 wt%Al–15.2 wt%Ta at its liquidus temperature (without superheat or undercooling). A cooling rate of 0.28 K/s is applied at the bottom and

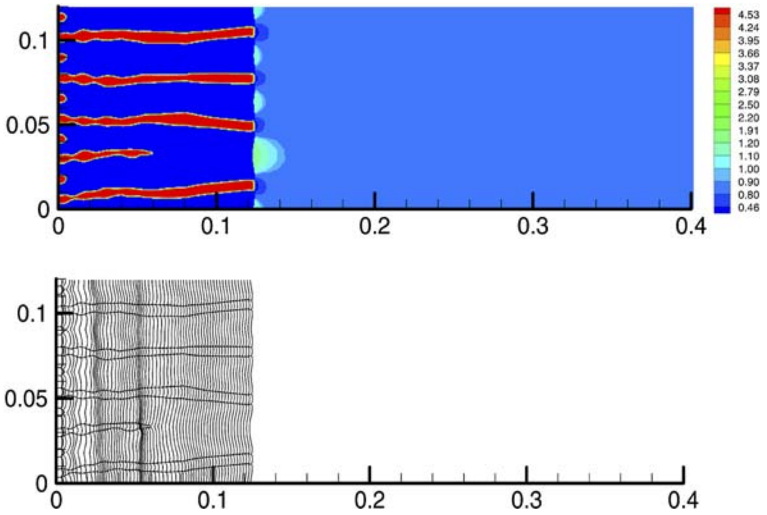


Fig. 22. Solute concentration at time about 80 s and the evolution of the interface position (with ten β seeds).

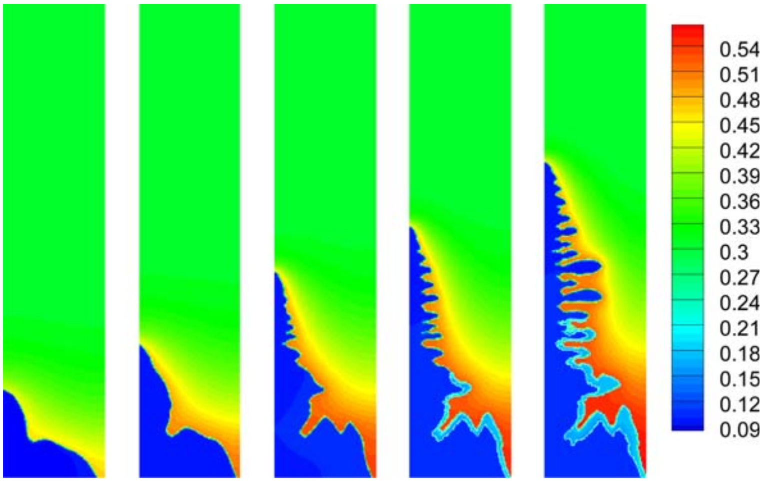


Fig. 23. Solute concentration for peritectic growth of an Fe–C alloy at time 0.6 s, 1.0 s, 1.5 s, 1.8 s and 2.4 s.

Table 1
Comparison of the results obtained with the current methodology for peritectic growth with those given in [38]

Quantity	Present calculations	From [38]
Tip velocity of ferritic dendrite (μm/s)	195	200
Distance of austenitic phase behind dendrite tip (μm)	300	350
Solute boundary layer (from contour plot) (μm)	≈50	≈50
Time of austenitic phase growth (s)	1.6	1.5–2.2

sides. The top side is assumed to be adiabatic. Utilizing symmetry, computation is done only in the left half of the domain. For adaptive meshing, we use an initial grid 1 × 2 and a refinement level of 10, which corresponds to a full mesh of 1024 × 2048. With this mesh, the width of the diffused interface is 12.2 μm, which is a reasonable value for meso-scale simulations. The CFL coefficient for adaptive time stepping is 0.5.

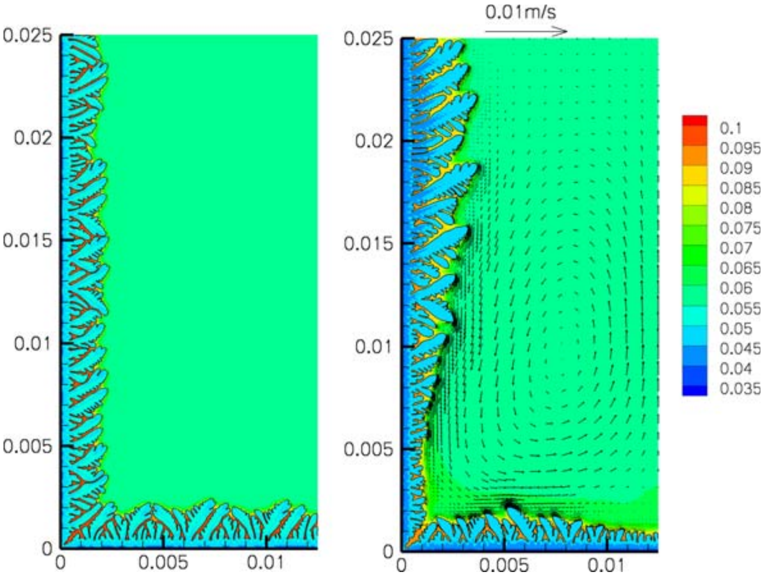


Fig. 24. Solute concentration of Al. On the left: without convection and on the right: with convection.

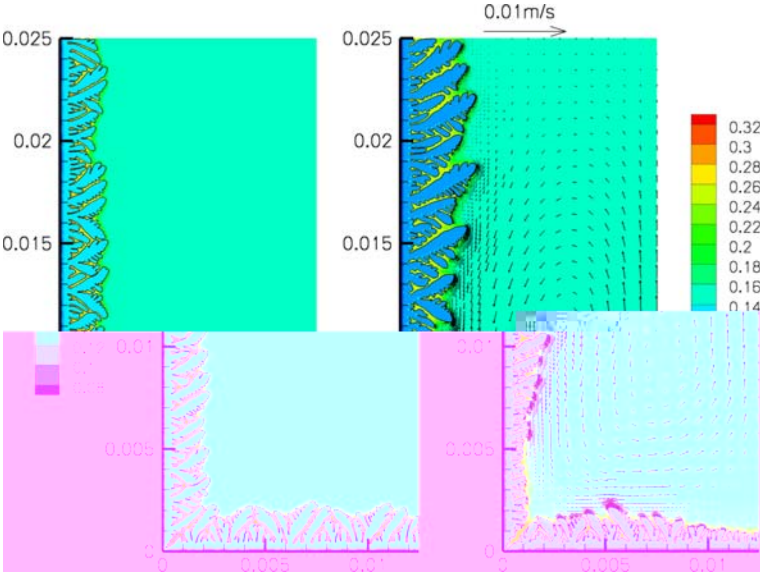


Fig. 25. Solute concentration of Ta. On the left: without convection and on the right: with convection.

In order to study the effects of convection, we carried two simulation runs: one without convection and one with buoyancy driven flow. With convection, the growth rate is about 22% faster than without convection. The dendrite tip front reaches roughly the same position as shown in Figs. 24 and 25 at time about 161 s in the case with convection and 196 s in the case without convection. Fig. 24 shows that the concentration of Al in the solid phase is slightly decreased due to convection. The underlying reason is that fluid flow enhances solute redistribution by taking away solute rejected from the dendrite tips into the liquid bulk. On the other hand, solute accumulated before the dendrite tips can only be taken away by diffusion. This phenomenon is more obvious for the Ta alloy component as shown in Fig. 25. The lower solute concentration in front of the dendrite tips explains why the growth speed is higher for the case with convection. Fluid flow also causes segregation. Solute concentration at the bottom of the cavity is higher than at the top of the cavity. This

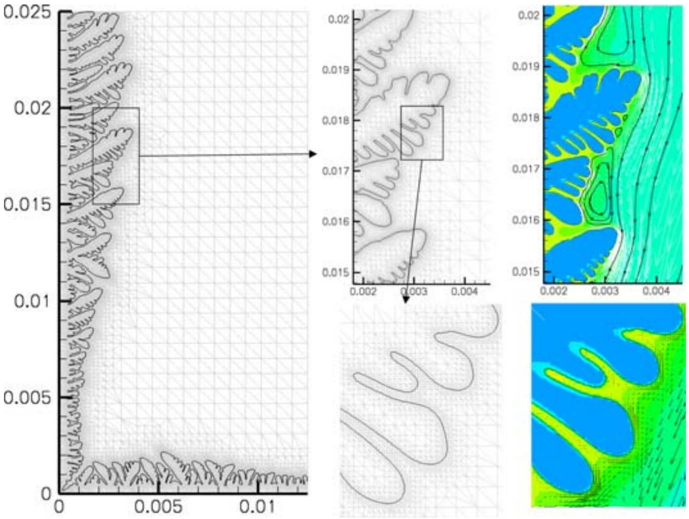


Fig. 26. Adaptive mesh and interface position. In the right picture, the domain is colored with concentration of Ta for better visibility of the flow field near the dendrite tips.

leads to a significant difference in the dendrite growth velocity resulting in a non-uniform pattern as shown in Figs. 24 and 25.

The adaptive mesh at time 161 s for the convection case is shown in Fig. 26. As shown in the magnified picture in Fig. 26, fluid flow passes through the dendrite tips and creates small eddies which enhance the mixing of solute.

5.5. Three-dimensional dendritic growth

Using Ni–Cu alloy with properties the same as in the example of Section 5.2, a simulation was presented earlier of 3D crystal growth using a phase field model assuming constant temperature (1356 K) in a cubic domain of size 3.5×10^{-3} cm [42]. An explicit scheme was used to perform computation on a mesh of 500^3 (125 million degrees of freedom). In this work, we use an adaptive mesh with initial grid 5^3 and six levels of refinement, which is equivalent to a uniform mesh of 320^3 (32 million degrees of freedom). With very high under-cooling (about 226 K), a high temperature gradient exists. So the simplification on the temperature considered in [42] (i.e. assuming constant temperature) is not taken here. As shown in Fig. 27, substantial secondary dendrite arms are formed behind the primary dendrite arms. Due to the significant under-cooling (226 K), the steady growth tip velocity is as high as 3000 cm/s. With such a high growth velocity, both the thermal and the solutal boundary layers are very thin as demonstrated in Fig. 28. The lowest concentration in the crystal is observed along the x, y and z axes, which corresponds to the growth of the primary dendrite tips.

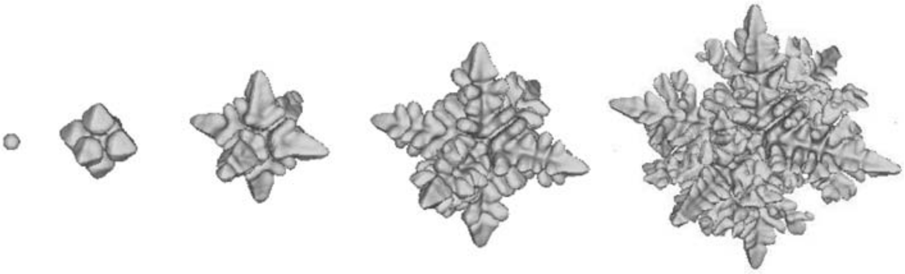


Fig. 27. Computed Ni–Cu crystal shape at steps 1, 51, 101, 151 and 211.

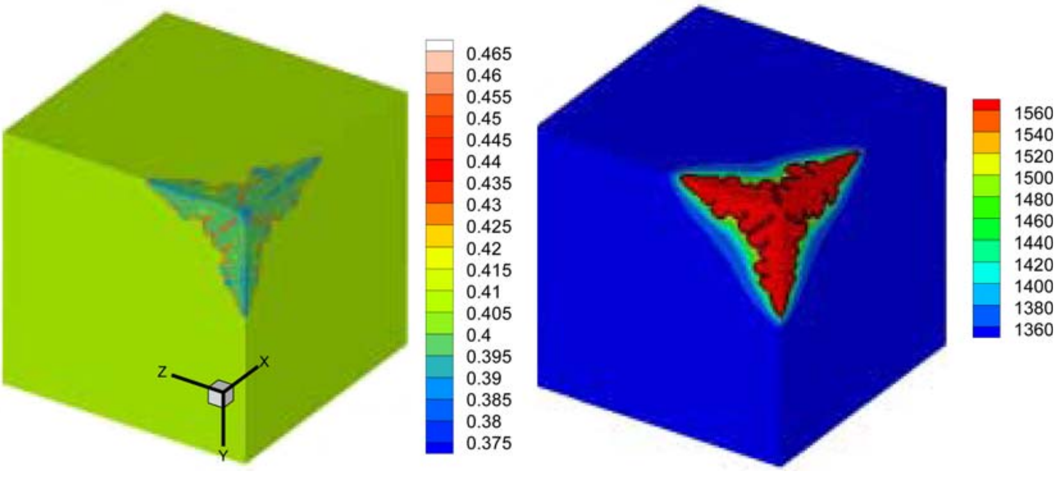


Fig. 28. Solute and temperature field at step 151 for Ni–Cu crystal growth.

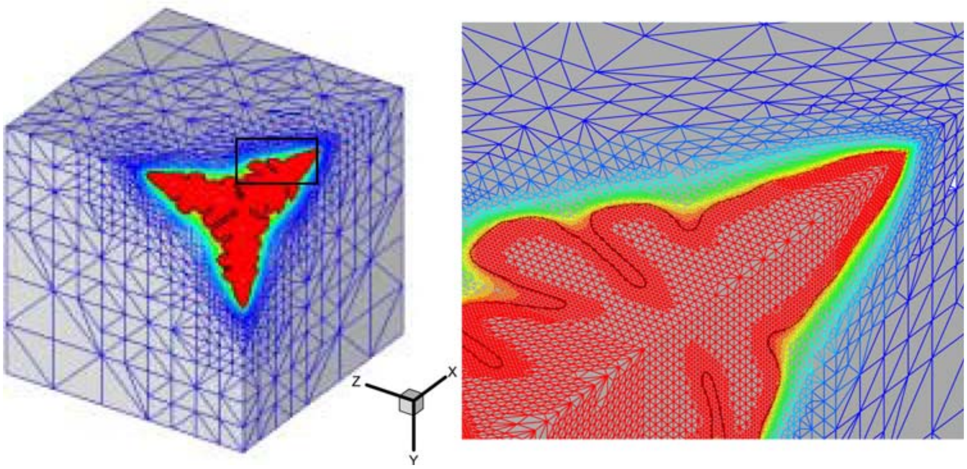


Fig. 29. Adaptive mesh colored with temperature at step 151 for Ni–Cu crystal growth. The figure on the right shows the magnified mesh near the primary dendrite in the x direction.

Computation for three-dimensional problems is often very intensive. With 16 nodes (each node with two 2048 MHz CPUs) in the Cornell Theory Center, this computation takes about 12 h. Using the adaptive meshing technique, the number of nodes and elements are only about 0.5 million and 3.0 million, respectively, at step 211. Without adaptive meshing, 32 million nodes and 192 million elements will be required to give an equivalent resolution. At previous time steps, the speed up of adaptive meshing is more obvious since the crystal is small. A typical adaptive mesh used in the computation is shown in Fig. 29.

In order to study the convergence of our numerical algorithm for this problem, we use an adaptive mesh with the same initial grid 1^3 and different refinement levels from 6 to 9 as shown in Fig. 30. Results show that

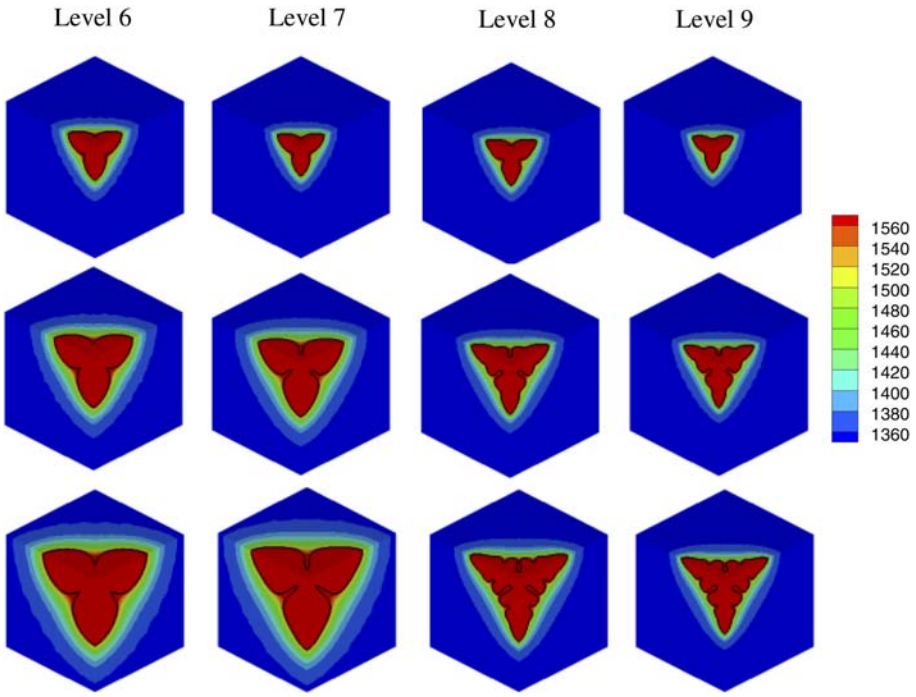


Fig. 30. Interface position and temperature field at times about $2e-7$ s, $7e-7$ s and $1.5e-6$ s (rows 1, 2 and 3), with the same initial grid 1^3 but different refinement levels from 6 to 9 (columns 1, 2, 3 and 4).

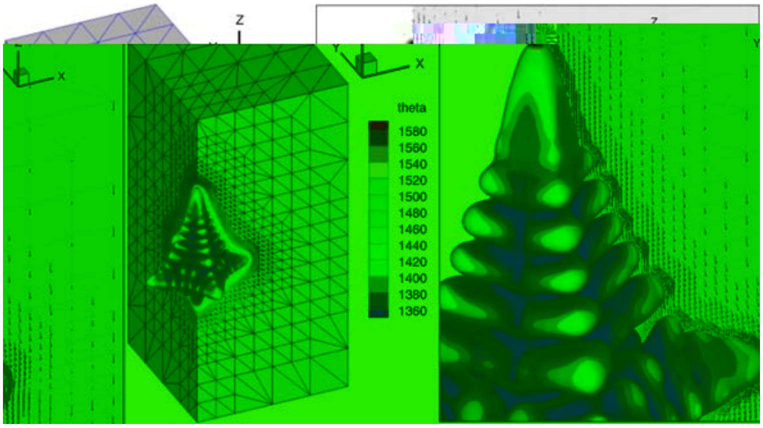


Fig. 31. Ni–Cu crystal growth with inlet flow from the top. Left: dendrite colored with interface velocity, mesh colored with temperature; right: flow passing by the upstream and perpendicular stream.

at least eight refinement levels (corresponding to a uniform mesh of 256^3) are required to numerically resolve the secondary dendrite arms.

The ability to study three dimensional pure material crystal growth under convection has been demonstrated in our previous work [16]. The three-dimensional free dendritic growth computation results compared very well with other numerical studies using phase field methods [29,43] at a high undercooling and front tracking techniques [5] at a low undercooling. Here, we present the three-dimensional results of coupling fluid flow for a binary alloy, which has not been previously reported in the literature. The same alloy Ni–Cu is used with the same conditions except that an inlet flow with velocity 0.1 m/s is applied at the top of the domain. In this case, we use an initial grid of $1 \times 1 \times 2$ with refinement level 8. This corresponds to a full mesh of $256 \times 256 \times 512$. Only 1/4 of the total domain is used due to symmetry. Similar to the pure material free dendritic growth case [16,29,43], crystal growth at the upstream arm is enhanced. However, unlike the pure material case in which the crystal growth is relatively smooth [16,29,43], perturbations quickly develop into secondary dendrite arms as shown in Fig. 31.

6. Conclusions

A level set method combining features of front-tracking and fixed domain methods is presented to model dendritic solidification of alloy systems. Some of the key features of the presented method include: (a) accurate tracking of the interfaces using the level set method; (b) heat/mass/momentum transport computation by avoiding direct application of interface conditions (solid/liquid boundaries); (c) 2D/3D finite element computation with fast adaptive meshing; and (d) multiple solid phase capability. The method is substantially simpler to implement relative to front tracking models and can serve as an efficient alternative approach to multiphase field methods.

The numerical investigations shown here have demonstrated that the present method can serve as an accurate and computationally effective alternative tool for modeling multi-phase multi-component alloys. Current work is in progress to apply this methodology to multi-scale alloy solidification.

Acknowledgements

The work presented here was funded by the University-Industry Partnerships for Aluminum Industry of the Future Program of the Office of Industrial Technologies of the US Department of Energy (DE-FC07-02ID14396). The computing was performed using a Beowulf cluster in our laboratory acquired with resources provided by the Air Force Office of Scientific Research. Additional computing resources were provided by the Cornell Theory Center.

References

[1] D. Juric, G. Tryggvason, A front-tracking method for dendritic solidification, J. Comput. Phys. 123 (1996) 127–148.

[2] N. Al-Rawahi, G. Tryggvason, Numerical simulation of dendritic solidification with convection: two-dimension geometry, J. Comput. Phys. 180 (2002) 471–496.

[3] P. Zhao, J.C. Heinrich, Front-tracking finite element method for dendritic solidification, J. Comput. Phys. 173 (2001) 765–796.

[4] P. Zhao, M. Venere, J. Heinrich, D. Poirier, Modeling dendritic growth of a binary alloy, J. Comput. Phys. 188 (2003) 434–461.

[5] N. Al-Rawahi, G. Tryggvason, Numerical simulation of dendritic solidification with convection: three-dimension flow, J. Comput. Phys. 194 (2004) 677–696.

[6] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Annu. Rev. Mater. Res. 32 (2002) 163–194.

[7] U. Hecht, L. Granasy, T. Pusztai, B. Bottger, M. Apel, V. Witusiewicz, L. Ratke, J. De Wilde, L. Froyen, D. Camel, B. Drevet, G. Faivre, S.G. Fries, B. Legendre, S. Rex, Multiphase solidification in multicomponent alloys, Mater. Sci. Eng. R 46 (2004) 1–49.

[8] A. Wheeler, W. Boettinger, G. Mcfadden, Phase-field model for isothermal phase transitions in binary alloys, Phys. Rev. A 45 (1992) 7424–7440.

[9] S.L. Wang, R.F. Sekerka, A.A. Wheeler, B.T. Murray, S.R. Coriell, Thermodynamically consistent phase field models for solidification, Physica D 69 (1993) 189–200.

[10] J.A. Warren, W.L. George, A parallel 3D dendritic growth simulator using the phase field method, J. Comput. Phys. 177 (2002) 264–283.

[11] N. Provatas, N. Goldenfeld, J.A. Dantzig, Adaptive mesh refinement computation of solidification microstructures using dynamic data structures, J. Comput. Phys. 148 (1999) 265–290.

[12] J.A. Sethian, Level Set Methods (Evolving Interfaces in Geometry, Fluid mechanics, Computer Vision, and Material science), Press Syndicate of the University of Cambridge, Cambridge, 1996.

[13] S. Osher, R. Fedkiw, Level Set Methods and Dynamic Implicit Surfaces, Springer, New York, 2003.

[14] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, J. Comput. Phys. 135 (1997) 8–29.

[15] Y.T. Kim, N. Goldenfeld, J. Dantzig, Computation of dendritic microstructures using a level set method, Phys. Rev. E 62 (2000) 2471–2474.

[16] L. Tan, N. Zabaras, A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods, J. Comput. Phys. 211 (2006) 36–63.

[17] 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.

[18] R. Tonhardt, G. Amberg, Phase-field simulation of dendritic growth in a shear flow, J. Crystal Growth 194 (1998) 406–425.

[19] W. Shyy, H.S. Udaykumar, M.M. Rao, R.W. Smith, Computational Fluid Dynamics with Moving Boundaries, first ed., Tailor & Francis, Washington, DC, 1996.

[20] Y. Sun, C. Beckermann, Diffuse interface modeling of two-phase flow based on averaging: mass and momentum equations, Physica D 198 (2004) 281–308.

[21] F.P. Incropera, W.D. Bennon, A continuum model for momentum, heat and species transport in binary solid–liquid phase change systems – I. Model formulation, Int. J. Heat Mass Transfer 30 (1987) 2161–2170.

[22] D.R. Poirier, P. Nandapurkar, S. Ganesan, The energy and solute conservation equations for dendritic solidification, Metall. Trans. B 22 (1990) 889–900.

[23] C. Beckermann, R. Viskanta, Mathematical modeling of transport phenomena during alloy solidification, Appl. Mech. Rev. 46 (1993) 321–336.

[24] J. Ni, C. Beckermann, A volume-averaged two-phase model for transport phenomena during solidification, Metall. Trans. B 22 (1990) 349–361.

[25] N. Zabaras, D. Samanta, A stabilized volume-averaging finite element method for flow in porous media and binary alloy solidification processes, Int. J. Numer. Method Eng. 60 (2004) 1103–1138.

[26] N. Palle, J. Dantzig, An adaptive mesh refinement scheme for Mullins Sekerka theory [solidification problems], Metal. Mater. Trans. A 27 (1996) 707–717.

[27] H. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.

[28] B. Merriman, J. Bence, S. Osher, Motion of multiple junctions: a level set approach, J. Comput. Phys. 112 (1994) 334–363.

[29] J. Jeong, N. Goldenfeld, J. Dantzig, Phase field model for three-dimensional dendritic growth with fluid flow, Phys. Rev. E 64 (2001) 0416021–0416024.

[30] T.E. Tezduyar, Stabilized finite element formulations for incompressible flow computations, Adv. Appl. Mech. 28 (1992) 1–43.

[31] J. Chessa, P. Smolinski, T. Belytschko, The extended finite element method (XFEM) for solidification problems, Int. J. Num. Method Eng. 53 (2002) 1959–1977.

[32] J.C. Ramirez, C. Beckermann, Examination of binary alloy free dendritic growth theories with a phase field model, Acta Mater. 53 (2005) 1721–1736.

[33] W.J. Boettinger, J.A. Warren, Simulation of the cell to plane front transition during directional solidification at high velocity, J. Crystal Growth 200 (1999) 583–591.

[34] T. Takaki, T. Fukuoka, Y. Tomita, Phase field simulation during directional solidification of a binary alloy using adaptive finite element method, J. Crystal Growth 283 (2005) 263–278.

[35] W. Mullins, R. Sekerka, Stability of a planar interface during solidification of a dilute binary alloy, J. Appl. Phys. 35 (1964) 444–451.

[36] W. Kurz, D.J. Fisher, Fundamentals of Solidification, third ed., Trans Tech Publications Ltd., Switzerland, 1989.

[37] M. Apel, B. Boettger, H.J. Diepers, I. Steinbach, 2D and 3D phase field simulations of lamella and fibrous eutectic growth, J. Crystal Growth 237–239 (2002) 154–158.

[38] J. Tiaden, Phase field simulations of the peritectic solidification of Fe–C, J. Crystal Growth 198/199 (1999) 1275–1280.

[39] J. Jeong, N. Goldenfeld, J. Dantzig, Phase field model for three-dimensional dendritic growth with fluid flow, Phys. Rev. E 64 (2001) 041602.

[40] H.J. Diepers, C. Beckermann, I. Steinbach, Simulation of convection and ripening in a binary alloy mush using the phase field method, Acta Mater. 47 (1999) 3663–3678.

[41] S.D. Felicelli, D.R. Poirier, J.C. Heinrich, Macrosegregation patterns in multicomponent Ni-base alloys, J. Crystal Growth 177 (1997) 145–161.

[42] W.L. George, J.A. Warren, A parallel 3D dendritic growth simulator using the phase field method, J. Comput. Phys. 177 (2002) 264–283.

[43] Y. Lu, C. Beckermann, J.C. Ramirez, Three-dimensional phase field simulations of the effect of convection on free dendritic growth, J. Crystal Growth 280 (2005) 320–334.