Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 226 (2007) 131–155
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Modeling the growth and interaction of multiple dendrites in solidification using a level set method

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 15 December 2006; received in revised form 22 March 2007; accepted 29 March 2007
Available online 5 April 2007

Abstract

A level set method is presented to study the growth and interaction of multiple dendrites in solidification. The method couples thermal and solute diffusion with propagation of multiple interfaces. A single signed distance function is used to track the solid–liquid interface with the aid of markers, the value of which is the orientation angle, for identification of different crystals. The problem of evolving multiple crystal interfaces is reduced to two tasks: (1) tracking one level set variable (signed distance function) and (2) determination of the marker for a newly solidified finite element nodal point. Tracking a single level set variable is implemented by solving the level set equation with interface velocity computed from an extended Stefan equation using the marker information (crystal orientation). Determination of the marker for a newly solidified finite element nodal point is implemented by using an algorithm modified from the fast marching technique. Both of these two steps are computationally efficient and the approach is suitable for incorporating effects of multiple crystals. Convergence and accuracy of this approach are demonstrated by using different grid spacings and comparing with results obtained from the multi-phase level set method. A parametric study is performed to investigate the effects of solidification speed and thermal gradient on the resulting solidification microstructure pattern. Numerical results of columnar-to-equiaxed transition (CET) qualitatively agree with an analytical estimation and are similar to previous numerical results obtained using a phase field method. A convergence study is performed to determine the appropriate grid spacing for numerical simulation. At lower surface tension, CET occurs at a lower thermal gradient for a giving solidification speed. Secondary dendrite formation is more apparent with lower surface tension. The differences and similarities between the three-dimensional and two-dimensional growth results are analyzed. Randomness in crystal orientation and required undercooling for nucleation are modeled and found to have a great effect on the microstructure pattern. The efficiency of the present approach is finally demonstrated with an example that includes the growth of hundreds of crystals with consideration of randomness effects.

Keywords: Level set method; Multi-phase motion; Dendritic solidification; Crystal growth; Columnar-to-equiaxed transition

* Corresponding author. Tel.: +1 607 255 9104; 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 © 2007 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2007.03.023

1. Introduction

The formation of complex microstructure during solidification of pure materials and alloys has been of great interest for hundreds of years. Various numerical methods including cellular automata [1–4], front-tracking methods [5–8], phase field techniques [9–18] and level set methods [19–24] have been developed to study the growth of crystals and dendrites. Among these methods, the cellular automata technique is suitable for studying the interaction between multiple dendrites especially in three-dimensions (3D), because its computational requirement (memory and time) is low in comparison to other numerical methods. Cellular automata is widely used in many areas including biological systems and highway traffic modeling. In this technique, a collection of 'colored' cells on a grid of specified shape evolves through a number of discrete time steps according to a set of rules based on the states of neighboring cells. Cellular automata was first introduced into solidification systems in [25] with a growth kinetics model, which can reflect the growth along preferred directions (e.g. $\langle 1\,0\,0\rangle$ crystallographic orientations) during dendrite development. The basic idea of cellular automata for solidification is to mimic interface propagation by capturing nearby liquid cells to the solid body according to a certain criterion. In recent advances of this method, cellular automata is coupled with finite differences (CA-FD) [2,4] or the finite element (CA-FE) [1] by using a criterion based on numerical solution of temperature or solute concentration from finite differences or the finite element method. Curvature effects can also be taken into account [2]. Cellular automata for solidification is relatively easier to implement and requires less computational resources than most other numerical methods, including front-tracking, phase field method, and level set method. However, it has some deficiencies in its accuracy due to its discrete nature. For example, it is pointed out in [1] that cellular automata methods have a tendency to bias the results by introducing an anisotropy associated with the network of cells or sites. Although corrections can be introduced [1] to circumvent this problem, independence of the cellular automata results on the numerical grid size and mesh orientation is rarely demonstrated. It is also pointed out in [26] that cellular automata lacks the ability to accurately take into account the surface tension anisotropy effect, which is of great importance in dendritic growth.

Although cellular automata is very successful in predicting grain structures which account for interaction between many dendrites, many researchers are working on other more complicated and computationally more expensive methods since issues of accuracy are not sufficiently addressed by the cellular automata method. In fact, obtaining a converged solution for a single crystal independent of mesh orientation itself is a nontrivial task due to the existence of a moving interface during the solidification process [7,8,18,20–22]. During the last two decades, significant progress has been made in the simulation of single crystal growth using phase field methods [12,13,18]. These approaches, by considering a diffuse interface and a fixed-grid, avoid the need for applying temperature boundary conditions on the moving interface. A review of recent progress in phase field methods as applied to solidification processes is given in [18]. The basic idea of the phase-field method is to employ a phase-field variable Φ that varies smoothly from zero to unity between the two phases over the diffuse interface region, which has a small but numerically resolvable thickness. The phase field variable serves 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 [18]. 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. But on the other hand, this also leads to a drawback of the phase field methods since there are a large number of parameters involved in the solution of the evolution equations to be determined. Some of these parameters are difficult to determine for accurate physical crystal growth simulation of real world materials. The front-tracking method, however, can avoid the difficulty of determining parameters, since a sharp interface model is directly solved. Recently front-tracking techniques have been used successfully to reproduce the complex dendritic structure in crystal growth in undercooled melts including effects such as liquid trapping, tip-splitting, side branching and coarsening [5,6,27]. Successful 3D front-tracking implementations including the effects of melt flow have been presented [8]. The advantages of front-tracking methodologies lie in their ability to directly enforce the freezing interface temperature (Gibbs–Thomson) relation and energy balance (Stefan condition). Unfortunately, many of the current implementations of these conditions do not allow global energy conservation even though they may satisfy the Stefan condition pointwise.

The level set method is an alternative method to handle the sharp interface front directly and to avoid the asymptotic analysis needed in phase field models. It has been shown to be a promising mathematical tool for tracking the interface with low computational cost. In this method, interfacial geometric quantities such as curvature and outward normal can be easily calculated using the level set variable ϕ, which is simply the signed distance to the solid/liquid interface with '+' sign in the liquid phase and '−' sign in the solid phase. The method was first applied to Stefan problems in [19]. Level set calculations for dendritic growth were reported in [20]. In these works [19,20], the heat flux at nodes near the interface is interpolated in order to calculate the interface velocity via the Stefan condition. During this interpolation, the temperature on the interface is fixed at the equilibrium temperature defined from the Gibbs–Thomson relation. However, like front-tracking methods, the 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 the discretization error [28]. This may result in large variation of the computed crystal shapes if meshes of different sizes and orientations are used.

One common drawback of both diffuse interface (phase field methods) as well as the interface tracking (front-tracking and level set methods) techniques is the significant computational effort required to perform any large scale realistic simulation. For instance, early implementations of the phase field method necessitated using grid spacing of the order of the interfacial thickness (the so-called sharp interface limit, see [12] for a detailed discussion). These issues were resolved to an extent in [12], where a "thin-interface" limit of the phase field equations was derived. Furthermore, developments in adaptive meshing have significantly improved the ability of such models to efficiently and accurately simulate phase transition behavior. Nevertheless, to obtain a fully converged solution for the growth of a single crystal using these methods still remains a non-trivial task. This point is particularly brought into context in [13], where it took approximately 10 CPU hours on a Sun UltraSPARC 2200 workstation to simulate the growth of a single crystal at dimensionless undercooling 0.55 by utilizing the improved phase field method [11] and adaptive meshing technique. On the other hand, since level set and other front tracking methods explicitly track the interface, they do not suffer from this particular problem. The major problems plaguing these explicit tracking techniques are energy conservation issues. This arises primarily due to the direct application of Gibbs–Thomson relation on the interface. These set of problems can only be resolved by using a fine mesh. As demonstrated in [5,19,20], a typical 2D mesh size to obtain a converged solution using the front-tracking method or level set method is of the size about 400×400. In essence, both phase field and level-set methods are bottle-necked by (different) issues that make realistic multiple dendrite simulation (while demonstrating convergence and mesh independence of the solution) a computationally daunting task requiring huge meshes and consequently enormous computational resources.

The importance of interaction between multiple crystals has been demonstrated in many studies [6,19,21,26]. Ref. [6] uses a front-tracking method to study the interaction between multiple crystals with the same initial perturbation in the seeds. Refs. [19,21] demonstrate the ability of using the level set method to capture the growth and merging of multiple dendrites with different initial perturbations. However, in these studies [6,19,21], nucleation is not modeled. Only recently, Ref. [26] gives the first phase field study of interaction between tens of crystals by fully utilizing symmetry in 2D with modeling of nucleation. Interaction between hundreds or even thousands of randomly nucleated crystals with a convergence study has not yet been demonstrated in the literature. In all of the above studies, convergence and mesh orientation independence are not demonstrated. More importantly, the randomness of crystal orientation is not considered in these studies. To efficiently track multiple interfaces with random orientations requires a significant departure from the current numerical algorithms. The emphasis of this work is using a level set approach to address this ability and to study the effects of randomness in crystal orientation.

In our previous work [22], we proposed a numerical scheme based on the level set method, which combines features of both front-tracking methods and fixed domain methods. It has been demonstrated that less computational resources are required to obtain a converged solution for pure material single crystal growth comparing with other numerical methods. Extension of this work to alloys is demonstrated in [24]. In this work, we will demonstrate the ability of using the level set method for tracking the growth of multiple dendrites in both 2D and 3D. We will first summarize a nucleation and growth model, followed with numerical techniques to solve the mathematical model. At the numerical examples section, we will perform a convergence study, and investigate the randomness effects on the growth pattern. The importance of this work is that the interaction

between multiple dendrites can be studied numerically with demonstration of convergence, which will serve as the foundation for multi-scale modeling of solidification processes.

2. Mathematical model

To focus on the modeling of nucleation, we take a few simplifications in our model.

(1) All material properties are assumed to be constant if not mentioned. These include density ρ, heat capacity c, latent heat L, heat diffusion coefficient k, solute diffusion coefficient D, liquidus slope m_l, and partition coefficient k_p.
(2) Fluid flow effects are not considered in this work.
(3) Solute diffusion in the solid phase is neglected.

2.1. Previous model

In this section, we provide a brief review of our previous model with the mentioned simplifications. For more details of this model with consideration of convection, the interested reader can refer to [22].

The governing equations for modeling of heat transfer and solute transport during solidification are as follows:

$$\rho c \frac{\partial T(\mathbf{x},t)}{\partial t} = k\nabla^2 T(\mathbf{x},t), \quad \mathbf{x} \in \Omega,$$
$$\frac{\partial C^l(\mathbf{x},t)}{\partial t} = D\nabla^2 C^l(\mathbf{x},t), \quad \mathbf{x} \in \Omega^l,$$

where Ω is the total domain including both the liquid part Ω^l and the solid part Ω^s. The above two governing equations for temperature T and solute concentration in liquid C^l are simple diffusion equations and introduce a physical model that has been very well studied. However, due to the existence of the moving interface, Γ^{sl}, the boundary conditions listed below make the problem nontrivial.

Because of phase transformation, solute is rejected from the solid phase to the liquid phase leading to a solute rejection flux at the freezing interface:

$$D\frac{\partial C^l}{\partial \mathbf{n}} = -(1-k_p)C^l V\mathbf{n}, \tag{1}$$

where $\mathbf{n}$ is the normal direction of the solid–liquid interface pointing towards the liquid phase.

The temperature at the solid–liquid interface Γ^{sl}, T_I, equals the equilibrium temperature, T_*, given by the Gibbs–Thomson relation:

$$T_I = T_* \equiv T_m + m_l C^l - \epsilon_c \kappa - \epsilon_V V, \tag{2}$$

where T_m is the melting temperature of the solidifying material, ϵ_c is the curvature undercooling coefficient, κ is the curvature of the interface, ϵ_V is the kinetic undercooling coefficient, and V is the interface velocity.

The velocity of the solid–liquid interface is governed by the heat flux jump through the classical Stefan equation:

$$V = \frac{q^s - q^l}{\rho L}, \tag{3}$$

where q^s and q^l are the heat fluxes at the solid and liquid side of the interface, respectively.

We have introduced two assumptions for applying these boundary conditions indirectly.

(1) Solidification occurs in a diffused zone of width $2w$ that is symmetric around the zero level set. A phase volume fraction ε can be defined according to the signed distance to the interface, ϕ, which is simply the distance to the interface with '+' sign in the liquid phase and '–' sign in the solid phase:

$$\varepsilon(x,t) = \begin{cases} 0, & \phi(x,t) > w, \\ 1, & \phi(x,t) < w, \\ 0.5 - \phi/(2w), & \phi(x,t) \in [-w, w]. \end{cases}$$

(2) The solid–liquid interface temperature, T_{I}, is allowed to vary from the equilibrium temperature, T_*, in a way governed by

$$\frac{\mathrm{d}T_{\mathrm{I}}}{\mathrm{d}t} = -k_N(T_{\mathrm{I}} - T_*),$$

where k_N controls the rate with which T_{I} is designed to approach the desired equilibrium temperature T_*. In [22], for a given time step Δt, we selected $k_N = 1/\Delta t$ to guarantee that the interface temperature stably converges to the equilibrium temperature. The selection of Δt was based on a CFL condition for the level set function calculation [22].

From the second assumption, a so called 'extended Stefan equation' (see Section 2.3) can be introduced to compute the interface velocity of the diffuse interface by performing an energy balance at the diffuse interface [22]. This interface velocity is then utilized to evolve the solid–liquid interface by solving a level set equation. From the updated signed distance function, the first assumption can then be applied to find the liquid volume fraction, and solve for the temperature and concentration fields using volume averaging techniques.

2.2. Nucleation model

Nucleation is a very complicated phenomenon related with the type and amount of impurities in the material. In this work, we assume that the number of impurities per unit volume, i.e. the density of impurities, is ρ^{n}. Suppose that the domain of interested problem has a volume of V, then there will be $\rho^{\mathrm{n}}V$ potential nucleation sites, since each impurity serves as a potential nucleation site. For each potential nucleation site i, we model three random variables, location x_i, required undercooling for nucleation ΔT_i^{n} and orientation I_i. These random variables are independently sampled with the sampling scheme discussed below.

(1) x_i is the location of the potential nucleation site. In [26], the potential nucleation sites are distributed uniformly with a certain spacing. The advantage of modeling uniformly spaced potential nucleation sites is that determination of their locations is very simple. This idea is followed in our computations in Section 4.2 in order to allow us to compare our results with those reported in [26]. A big disadvantage of modeling uniformly spaced potential nucleation sites is that it requires a structured mesh with proper grid spacing so that a potential nucleation site coincides with a finite element node. The nature of randomness in the location of potential nucleation sites is also lost by modeling them to be uniformly spaced. To overcome these disadvantages, we assume that x_i is equally probable at each point of the whole domain instead of deterministically at a particular mesh nodal point.In our computations with consideration of randomness (Sections 4.2.4 and 4.3), we apply the following sampling scheme element by element. For each element e of the initial coarse mesh, we assign a set of potential nucleation sites with the following steps:
   (a) Compute volume of element e, V_e. Since ρ^{n} is the number of potential nucleation sites in a unit volume, we will have $\rho^{\mathrm{n}}V_e$ potential nucleation sites inside element e. In general, $\rho^{\mathrm{n}}V_e$ will not be an integer. For example, if $\rho^{\mathrm{n}} = 100$, $V_e = 0.023$, then $\rho^{\mathrm{n}}V_e = 2.3$. We want to use a sampling scheme such that the expected number of potential nucleation sites assigned to the element is 2.3.
   (b) Let n_e be the integer part of $\rho^{\mathrm{n}}V_e$ (e.g. when $\rho^{\mathrm{n}}V_e = 2.3$, $n_e = 2$). Sample p from the uniform distribution with range $[0,1]$. If $p < \rho^{\mathrm{n}}V_e - n_e$ (i.e. $p < 0.3$ in our example), then set $n_e = n_e + 1$. This step guarantees that the expected number of potential nucleation sites inside element e is $\rho^{\mathrm{n}}V_e$. For the example of $\rho^{\mathrm{n}}V_e = 2.3$, with this step, the probability of $n_e = 2$ is 0.7, while the probability of $n_e = 3$ is 0.3. So the expected number of potential nucleation sites in the element becomes $\langle n_e \rangle = 0.7 \times 2 + 0.3 \times 3 = 2.3$, which is the value of $\rho^{\mathrm{n}}V_e$.

(c) For each potential nucleation site $i = 1, 2, \ldots, n_e$, sample its location x_i (uniformly distributed in element e), required undercooling ΔT_i^{n} ($N(\mu, \sigma^2)$), and orientation angle I_i (uniformly distributed in $[0, 2\pi]$).
After an element is refined into a few 'child' elements, the potential nucleation sites assigned to it are further assigned to its 'child' elements according to whether the location of the potential nucleation site falls inside the 'child' element. In this way, we only need to go through all elements to determine which potential nucleation sites become seeds for crystal growth.

(2) ΔT_i^{n} is the required undercooling for the potential nucleation site i to nucleate and become a crystal seed. Only if the undercooling at the potential nucleation site i is greater than ΔT_i^{n}, site i becomes a crystal seed. In our numerical example in Section 4.2.4, the required undercooling for nucleation ΔT_i^{n} is assumed to be at a fixed value of 8 K following [26]. In Section 4.3, we assume that ΔT_i^{n} follows a normal distribution with mean μ and variance σ^2.

(3) The other random variable, I_i, is the orientation angle at potential nucleation site i. The orientation angle is the preferred crystal growth direction. In two dimensions, the preferred crystal growth direction can be mapped to a point on the circumference of a unit circle. Any value between 0 and 2π corresponds to a preferred crystal growth direction. In other words, we can use one real number, I, to represent the preferred crystal growth direction. In three-dimensional growth, the preferred crystal growth direction can be mapped to a point on a unit sphere. Two angles of spherical coordinate system (zenith angle and azimuth angle) are often used to represent a point on a unit sphere. So for extension in 3D, the orientation angle I is taken as a vector with two components, one for zenith angle (angle from the positive z-axis) and one for azimuth angle (angle from the positive x-axis). In the numerical examples in Section 4.2.4, we only considered the randomness of orientation angle in two dimensions. I_i is sampled from a uniform distribution from 0 to 2π. After nucleation at location of potential nucleation site i, the crystal may in general rotate as it growths (e.g. as a result of convection). This leads to change in orientation angle for the crystal. This movement of crystal is not considered in this work. In other words, the orientation of the crystal, I, is assumed to be fixed at the value when it is nucleated (i.e. I_i if it is nucleated from potential nucleation site i) during growth of the crystal.

Using the above sampling technique, the potential nucleation sites do not need to be at the same location as the finite element nodes. This gives flexibility to select grid spacing. There are n_e potential nucleation sites (with expectation $\rho^{\mathrm{n}} V_e$) inside element e. In our implementation, we maintain a link list to contain information about these potential nucleation sites including x_i, ΔT_i^{n} and I_i for each element. For determining whether nucleation occurs at location of a potential nucleation site or not, we use data at nodes of the element to interpolate data at location of the potential nucleation site. After temperature T_i and concentration C_i is obtained at location of a potential nucleation site, x_i, using interpolation, we check whether $T_i \leqslant T_{\mathrm{m}} + m_{\mathrm{l}} C_i - \Delta T_i^{\mathrm{n}}$. If so, a small crystal seed will be put at location x_i.

2.3. Growth model

The velocity at the phase boundary of each crystal, V, is governed by the Stefan equation. Notice that for numerical convenience and making the scheme energy conserving, we have used an interface energy balance that is different from the Stefan equation [22],

$$V = \frac{q^{\mathrm{s}} - q^{\mathrm{l}}}{\rho L} + \frac{2cw}{L} k_N (T_* - T_{\mathrm{I}}). \tag{4}$$

The converged solution using this formula has been proved to be the same as using the Stefan equation [22].

In our previous model, we have assumed a four-fold symmetry of the crystal structure. The Gibbs–Thomson relation coefficient ϵ_{c} is modeled as

$$\epsilon_{\mathrm{c}} = d_0 \{1 - 15\epsilon \cos[4\alpha(\mathbf{n})]\}, \tag{5}$$

where d_0 is the capillary length, ϵ is a coefficient describing the surface anisotropy extent, and $\alpha(\mathbf{n})$ is the angle from the positive x-axis to the normal direction $\mathbf{n}$. With the above formulation for the Gibbs–Thomson

relation coefficient ϵ_c, the crystal grows fastest at directions with angles 0°, 90°, 180° and 270° to the x-axis, and slowest at directions with angles 45°, 135°, 225° and 315° to the x-axis. In this work, we want to model the growth of multiple dendrites. Each dendrite is growing with different orientation angle I fastest at directions with angles $0° + I$, $90° + I$, $180° + I$ and $270° + I$ to the x-axis and slowest at directions with angles $45° + I$, $135° + I$, $225° + I$ and $315° + I$ to the x-axis. So the same 4-fold symmetry of the crystal structure is used, except with a '$-I$' term in the cosine function, as shown below:

$$\epsilon_c = d_0\{1 - 15\epsilon\cos[4(\alpha(\mathbf{n}) - I)]\}. \tag{6}$$

Since I varies for different crystals but is the same for the same crystal, different crystals will grow with different preferred orientations.

3. Numerical techniques

Comparing with single crystal growth, the additional difficulty of multiple crystal growth is to track multiple interfaces. We have implemented two numerical methods: (1) the multi-phase level set method by solving multiple signed distance functions, and (2) the level set method by solving a single signed distance function with the aid of markers. Emphasis of this work is on the second method due to its high efficiency and accuracy.

3.1. Multiple signed distance functions

For simulating the growth of multiple crystals, one way is to use multi-phase level set method with a signed distance function for each crystal. In the implementation of this method, a new level set solver is created to handle the evolution of a crystal when a potential nucleation site nucleates. Each level set solver is assigned with the crystal orientation I of the potential nucleation site at the time it is created. The following are the main steps of using this method at each time level:

(1) Compute undercooling at each potential nucleation site (interpolation will be required), and create new level set solvers to handle the newly created crystals if required undercooling is satisfied at the location of potential nucleation site.
(2) Compute the interface velocity on the interface of each crystal with information of crystal orientation I, and use the computed interface velocity to evolve the signed distance function of the corresponding crystal.
(3) Perform re-initialization of all signed distance functions.
(4) Solve for temperature and concentration fields using volume averaging techniques.
(5) Return to step 2 until convergence ($\|T_n^k - T_n^{k-1}\| \leqslant 10^{-3} \times \|T_n^k - T_{n-1}\|$) is achieved for this time level. Here n is the current time level, $n - 1$ is the previous time level and k is the iteration level.

Details of this method are provided in [24], where it is applied to study the growth of multiple phases instead of multiple crystals. More theoretical analysis of this multi-phase level set method can be found at [29,30]. In this work, this multi-phase level set method is not used as the main investigation tool but as a verification tool for the other method discussed next, because the multi-phase level set method requires the expensive solution of multiple level set equations. Solving multiple level set equations is only realistic for a small number of crystals. As the number of crystals increases, the multi-phase level set method becomes increasingly inefficient. If tens or even hundreds of crystals are present, this numerical method is impractical, since one cannot afford or does not want to store tens or hundreds of signed distance functions and solve tens or hundreds of level set equations.

3.2. Single signed distance function with markers

In this work, we do not model the evolution of the solid-solid phase transformation. Therefore, the solid/solid (crystal/crystal) interfaces are of no significant importance and are assumed to be static. The type of interface of great interest is the solid/liquid (crystal/liquid) interface. We can use a single signed distance function

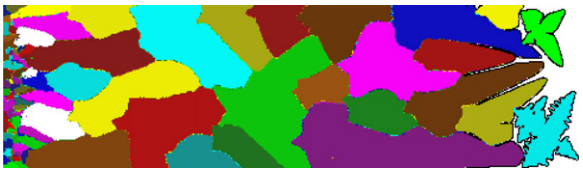


Fig. 1. Example of using a single signed distance function with markers.

to implicitly represent this interface. However, we also want to identify different crystals, because each crystal grows with different preferred orientations due to the underlying crystal structure. So we introduce another scalar (markers) to represent different crystals. The value of the markers is just the orientation of each crystal. As demonstrated in Fig. 1, different markers (shown with different colors) are used to identify different crystals. Although only one signed distance function is used, the growth of a crystal at its own preferred direction can be captured with the aid of markers. The problem of evolving multiple crystal interfaces is reduced to tracking one level set variable (signed distance function) and determining the marker of a newly solidified node point. Tracking a single level set variable is implemented by solving the level set equation as before, while determining the marker of a newly solidified node point is implemented by using an algorithm updated from fast marching as discussed below.

The marker (orientation of the crystal, I) has physical meaning only in the solid phase. However, doing evaluation of the interface velocity requires that I is also defined in the liquid phase, since computation of the interface velocity is performed on both phases near the solid/liquid phase interface. In this work, we define I in the liquid phase as the orientation of the nearest crystal. With this definition, extension of I from the solid phase to the liquid phase can be implemented using the fast marching technique. We have a balanced heap data structure for easy tracking of the next closest nodal point to the solid–liquid interface. One important step of the fast marching algorithm is to extend values in an element (say A–B–C in the 2D case). Let values on two nodes (say A and B) be known, with the value on the other node (C) being unknown. In extending the interface velocity from nodes A and B to node C, interpolation is reasonable since one is interested in obtaining a smooth velocity field over the whole domain. However, for extending the crystal orientation angle from nodes A and B to node C, interpolation will not make much sense since node C belongs to either the same crystal as node A or the same crystal as node B. The following formula is thus used for extension of the orientation angle from nodes A and B to node C:

$$I_C = \begin{cases} I_A, & \text{if } \|AC\| \leqslant \|BC\|, \\ I_B, & \text{if } \|BC\| < \|AC\|. \end{cases} \tag{7}$$

Using this method, we can efficiently extend I from the solid phase to the liquid phase as demonstrated in Fig. 2. Notice that in Fig. 2, we have only extended I from the solid phase to a part of the liquid phase with $\phi < 3w$, which is far enough for computing the interface velocity. A newly solidified nodal point definitely falls in the region with $\phi < 3w$, since the CFL coefficient is selected to be less than 1. Therefore, the orientation angle I has already been extended to the newly solidified node.

The level set method uses a field (signed distance) to represent the interface. Let us suppose a new crystal is introduced into the solidification system by nucleation at location $\mathbf{x}_i$. Since the solid–liquid interface changes, the signed distance field needs to be updated with the following operation:

$$\phi(\mathbf{y}) \leftarrow \min(\phi^0(\mathbf{y}), \|\mathbf{x}_i - \mathbf{y}\| - R_0), \tag{8}$$

where ϕ^0 is the signed distance function before the potential nucleation site is nucleated at $\mathbf{x}_i$, R_0 is the size of the initial crystal seed at location $\mathbf{x}_i$ and $\mathbf{y}$ is the location of a node (see Fig. 3 for a related schematic). Notice that, we need to apply this operation on all nodes whenever there is a newly nucleated crystal. Fortunately, the number of operations is strictly proportional to the number of nodes. It is also very easy to parallelize, since no communication (except information of the newly created crystal) is required.

Fig. 2. Example of extending the orientation angle in the liquid phase. The orientation angle is extended to the liquid phase with $\phi < 3w$. The artifacts of interpolation lead to a color different from all nearby colors at some places of the mesh. The plotting software we utilized, Tecplot [31], automatically interpolates the orientation and plots the color corresponding to the interpolated value.

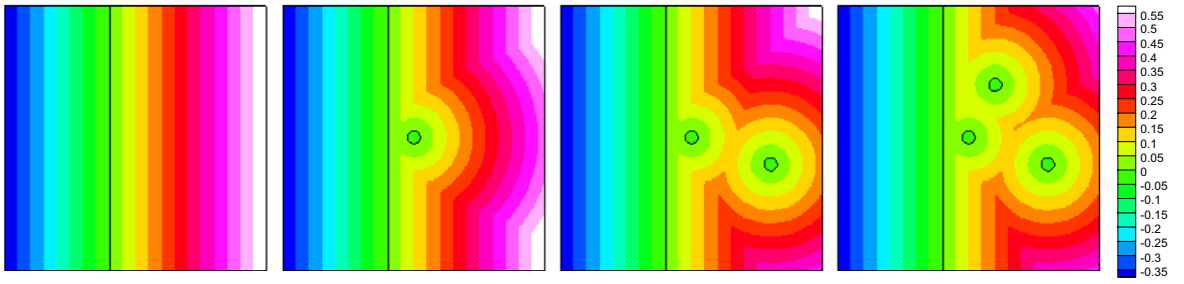


Fig. 3. Redefinition of the level set function when nucleation occurs in the domain $[0,1]\times[0,1]$: The initial level set field $\phi^0 = x - 0.4$ represents an interface at $x = 0.4$ (dark line). If nucleation occurs at (0.5,0.5) with initial seed radius 0.025, we redefine the level set field to be $\phi^1 = \min(\phi^0, \sqrt{(x-0.5)^2 + (y-0.5)^2} - 0.025)$. The contour value on the second figure is the value of ϕ^1 and the dark line shows the zero level set of ϕ^1 that appropriately captures the changes on the interface introduced by nucleation at $(0.5, 0.5)$. The remaining two figures demonstrate the ϕ^2, ϕ^3 fields when additional nucleation occurs at the locations $(0.8, 0.4)$ and $(0.6, 0.7)$.

The following are the main steps of using this method at each time level:

(1) Compute the undercooling at each potential nucleation site (interpolation will be required). For each newly nucleated crystal at location $\mathbf{x}_i$ apply the following operation on each node $\mathbf{y}$:

$$\phi(\mathbf{y}) \leftarrow \min(\phi^0(\mathbf{y}), \|\mathbf{x}_i - \mathbf{y}\| - R_0).$$

(2) Extend the orientation angle from the solid region to part of the liquid region with $\phi < 3w$ using the fast marching algorithm.
(3) Compute the interface velocity on the solid–liquid interface with information of orientation angle I, and use the computed interface velocity to evolve the signed distance function. Note that I plays a role in the computation of T_*, which plays a role in the computation of V using the extended Stefan equation.
(4) Perform re-initialization of the single signed distance function.

(5) Solve for the temperature and concentration fields using volume averaging techniques.
(6) Return to step 3 until convergence is achieved for this time level.

Comparing the above method with the multi-phase level set method, multiple signed distance functions are reduced to just one signed distance function and an orientation angle. Moreover, we do not need to solve the multiple level set equations. So this method is much more efficient than the multi-phase level set method. However, as shown in Eq. (7), the extension of the orientation angle is not smooth at liquid nodes, which are close to multiple crystals. Fortunately, for most extension cases of the orientation angle from values at nodes A and B, $I_A = I_B$. In other words, $I_A \neq I_B$ occurs only at locations close to multiple crystals. Note that the interface velocity is only computed at nodes adjacent to zero level set (i.e. nodes of elements cut by the zero level set). At other nodes, the interface velocity is extended using fast marching. So assuming that I is correct for the nodes of the elements cut by the zero level set, the above approximation will not make any difference in the calculation of the interface velocity. This means that I_C will make a difference in the interface velocity only when $I_A \neq I_B$ and C is a node adjacent to zero level set, which means that two crystals are already very close to each other. We will demonstrate that the difference between the results obtained with this method and the results obtained from the multi-phase level set method is actually negligible.

3.3. Other techniques to speed up computation

In this work, the finite element method is used to solve for the temperature, solute concentration and level set function with implementation details discussed in [23,24]. Comparing with the combined finite element method (for governing equations of temperature and solute) and finite difference method (for level set computation) [22], there is flexibility in the current implementation to use adaptively refined unstructured mesh, which is shown to be much more efficient than a uniform mesh in [24]. Since the level set variable gives information about how far away each node or element is from the interface, a refinement criterion based on the level set variable [24] is very convenient for generation of adaptive mesh locally refined near the solid–liquid interface. The effect of adaptive meshing with one level of refinement near the interface can be seen in Fig. 2. Significantly higher levels of refinement are usually used in the numerical examples discussed later. Other than adaptive meshing, domain decomposition (implemented with the aid of graph partition package ParMetis [32]) is also used for parallel computing to allow investigation of interaction between hundreds of crystals. Fast marching technique [23] is utilized to perform re-initialization of the level set variable and extension of the interface velocity. For solving the linear systems in the finite element method, we use the Krylov subspace method in the matrix solver package PETSc [33].

4. Numerical examples

4.1. Interaction between nine crystals

In this example, we consider the interaction between nine crystals with orientations $I = 0°$, $10°$, $20°$, $30°$, $40°$, $50°$, $60°$, $70°$, $80°$ as shown in Fig. 4. The domain size of interest is $[-1200, 1200] \times [-1200, 1200]$. The nine crystals are uniformly spaced with distance Δ. The initial shape of each crystal is circular with radius 30. A pure material is considered for this example with all parameters including ρ, c, k, L normalized to 1. The temperature on the interface satisfies the following Gibbs–Thomson relation: $T_* = -d_0\{1 - 15\epsilon\cos[4(\theta - I)]\}\kappa$ with $d_0 = 0.5$ and $\epsilon = 0.05$. Initially, the domain is undercooled at temperature -0.55, while inside the nine initial crystals, the initial temperature is taken as 0. The boundary of the domain is kept at temperature -0.55. We want to study how the nine crystals interact with each other during their growth.

4.1.1. Effects of crystal spacing

In this computation, we use an adaptive mesh equivalent to a uniform mesh of grid spacing 1. Here equivalent means that the smallest grid spacing in the adaptive mesh is equal to the grid spacing in the uniform mesh. We use the term 'equivalent', because both meshes lead to the same numerical solution as was demonstrated in our earlier work [24]. The CFL coefficient for adaptive time stepping is selected to be $\frac{1}{3}$. When the

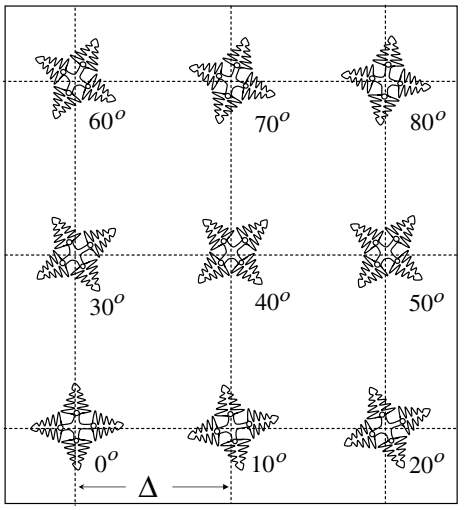


Fig. 4. Schematic of the growth of nine crystals.

spacing between the crystals is 800, i.e. $\Delta = 800$, the interaction between the crystals is weak as shown in Fig. 5. The closest distance between crystals is the distance between the right dendrite tip of the crystal with $I = 0°$ and the left dendrite tip of the crystal with $I = 10°$. The distance between any other two crystals is greater than the thermal boundary layer size. Therefore, each crystal grows almost independently, except that the thermal boundary layers of the two crystals $I = 0°$ and $I = 10°$ overlap at places around their closest tips. The independence and similarity of the thermal boundary layers for the nine crystals actually demonstrate that the computed solution is independent of mesh orientation. When the spacing between the crystals reduces to 600 or 400, the interaction between the crystals is very obvious as shown in Fig. 5. However, in all cases, we observe that the shapes of all 'free' dendrite tips are very similar. Here 'free' means that the dendrite tip is far away from other dendrites. More interestingly, the solution (interface position and temperature field) keeps partial symmetry in all three cases with different crystal spacing. For example, the two crystals with $I = 30°$ and $I = 60°$ are symmetrical to each other at places close to their center line $y = \frac{4}{2}$. Similarly, the two crystals with $I = 40°$ and $I = 50°$ are symmetrical to each other at places close to their center line $x = \frac{4}{2}$. This partial symmetry comes from the symmetry in the crystal orientations as shown in Fig. 4.

4.1.2. Comparison with the multi-phase level set method

In the above computation of studying crystal spacing effects, we have used only one signed distance function and nine markers to identify the interface of the nine crystals. It is more common to use multiple signed distance functions for tracking multiple interfaces with the level set method [29,30]. Here we study the interface position using both methods with grid spacing 1 and CFL coefficient $\frac{1}{3}$. As shown in Fig. 6, both methods predict almost the same interface position when the dendrite tips are close to the domain boundary. This computation gives us confidence on the accuracy of using only one signed distance function with markers, which is

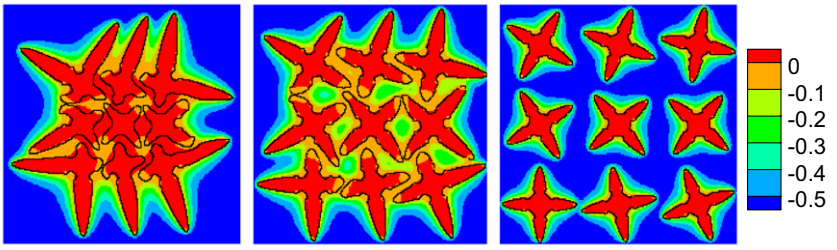


Fig. 5. Temperature field for the interaction between nine crystals with spacing 400, 600 and 800.

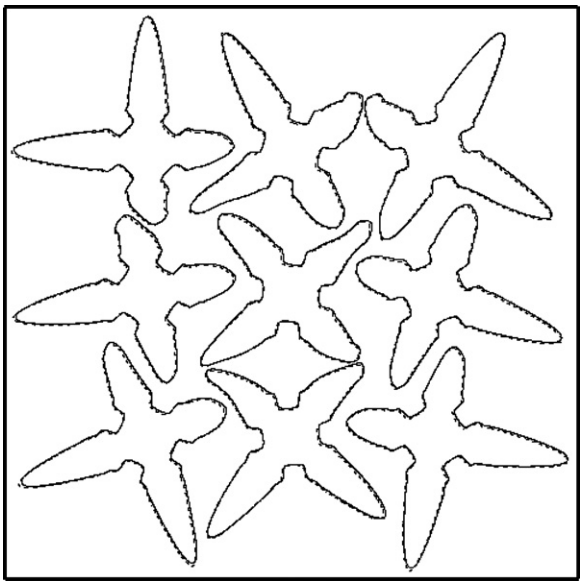


Fig. 6. Interface position when the dendrite tips are close to the domain boundary. Solid line: computed interface position using single signed distance function with markers. Dash dot line: computed interface position using multiple signed distance functions.

significantly much more efficient than using multiple signed distance functions, since only one level set equation needs to be solved. In all the other computations in this work, one signed distance function with markers will be used.

4.1.3. Convergence study

In the previous computations of studying crystal spacing effects, we have used an adaptive mesh (initial coarse mesh 75×75, refinement level 5) equivalent to a uniform mesh of grid spacing 1. However, in [13,20], a grid spacing of 0.4 was used for a similar problem with just one crystal. In order to demonstrate that grid spacing 1 already leads to a converged solution, we also computed the numerical solution for crystal spacing 600 using various grid spacings 1, 2, 4, 8, 16, 32 (corresponding to refinement levels 5, 4, 3, 2, 1 and 0), as demonstrated in Fig. 7. Notice that in the studies here, adaptive meshing is used. Thus the grid spacing actually refers to the smallest grid spacing of the adaptive mesh. To use roughly the same time discretization, the CFL coefficient for adaptive time stepping is selected to be $\frac{1}{3}, \frac{1}{6}, \frac{1}{12}, \frac{1}{24}, \frac{1}{48}, \frac{1}{96}$ corresponding to the selection of grid spacings 1, 2, 4, 8, 16, 32, respectively. As the grid spacing decreases, the computed dendrites become slimmer. To quantitatively study convergence, we define the error as the root mean square (rms) of the difference in the temperature and the "true" temperature (obtained with refinement level 5) at time 1.05×10^4 on nodes of a 75×75 uniform mesh, i.e. $\text{err} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{T_i - T_i^{\text{true}}}{0.55}\right)^2}$. Here n is the total number of nodes of the uniform mesh with value $76 \times 76 = 5776$, T_i is the temperature at node i at time 1.05×10^4, and T_i^{true} is the "true" temperature at node i at time 1.05×10^4. The relation between error and refinement level of the adaptive mesh is given in Fig. 8. The computed dendrites with grid spacing 32 are significantly different (much 'fatter') than the converged solution with the finest grid spacing. In fact, since grid spacing 32 is even larger than the initial crystal radius, it is expected that dendrites computed with grid spacing 32 are much 'fatter' than the converged solution. However, the solution with grid spacing 16 can still give us an insight on the interaction between the dendrites. The fact that grid spacing can be 40 times larger than the grid spacing used in the literature [13,20] ($16 = 40 \times 0.4$) and still able to provide reasonable solution suggests that the present technique has significant potential for the study of the interaction between many dendrites. Notice that grid spacing 40 times larger means that the element size is 40^2 times larger and the number of elements just $\frac{1}{40^2}$ times the original required number of elements. This would lead to a significant reduction of computational requirement in terms of memory and time. In 3D, this advantage will be even more obvious, since the number of elements can be

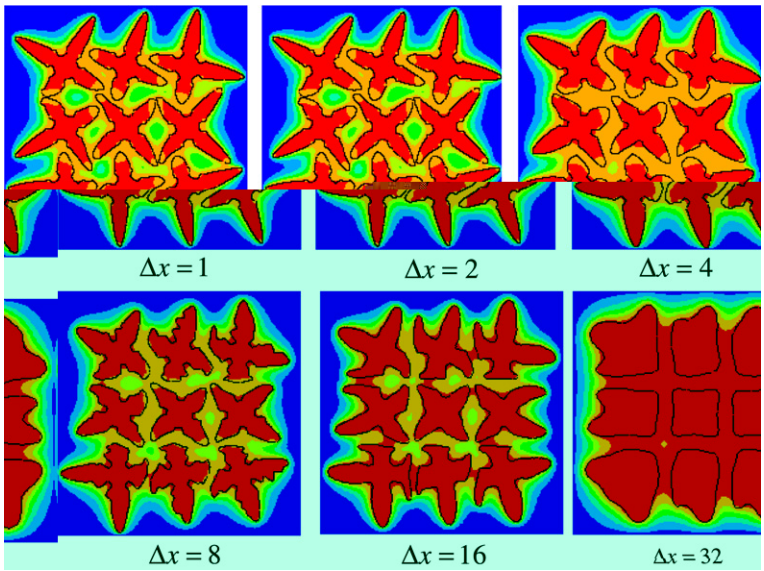


Fig. 7. Interaction between nine crystals with spacing 600 using various grid spacings 1, 2, 4, 8, 16 and 32.

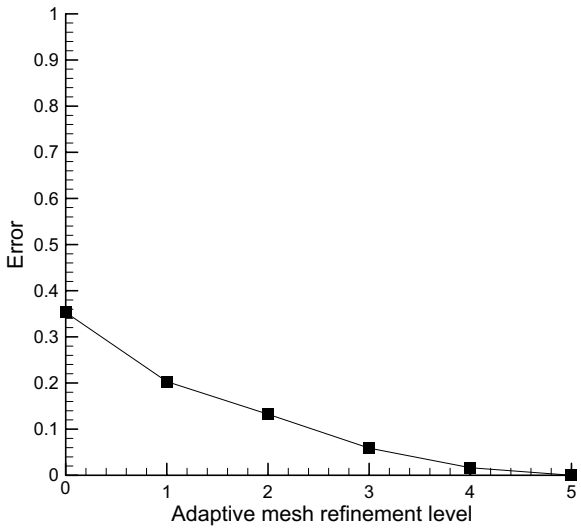


Fig. 8. Error at refinement levels 0–5 (corresponding to grid spacings 32, 16, 8, 4, 2 and 1).

reduced to $\frac{1}{40^3}$ of the required number of elements using grid spacing 0.4. We will demonstrate the interaction of multiple 3D dendrites in Section 4.2.3.

The main reasons that the present method converges nicely are (1) that the interface is tracked with a level set which could give accurate normal direction and interface curvature for accurate determination of the equilibrium temperature using the Gibbs–Thomson relation, and (2) that energy is numerically conserved since no essential boundary condition is applied on the interface, which could compensate for the numerical error in the interface velocity evaluation.

4.2. Columnar-to-equiaxed transition (CET)

Columnar to equiaxed transition is a phenomenon of great importance in casting. For decades, researchers have relied on the cellular automata method [1,2,25,34] to give numerical estimation of the microstructure type

(columnar or equiaxed) at different solidification speeds and thermal gradients. Arguing that cellular automata method has issues of mesh dependence and lack of demonstration for its convergence, Ref. [26] provides the first CET study using the phase field method. In this section, we will first study the same numerical example as in [26] as a validation of our numerical method. We will compare our numerical results of columnar-to-equiaxed transition (CET) with an analytical estimation and numerical results using a phase field method in [26]. Independence of the results on the numerical grid size is not demonstrated in the phase field study of [26].

Here we will first perform a convergence study for one typical case to determine the appropriate grid spacing for numerical simulation (Section 4.2.1). The investigation of CET for different process conditions is then reported in Section 4.2.2. Since crystal growth in real world is three-dimensional growth, we will consider the interaction between multiple crystals in 3D (Section 4.2.3). We will also incorporate many more crystals by using a much larger computational domain to study how the randomness of seed orientation affects the microstructure pattern (Section 4.2.4).

4.2.1. Convergence study

The important material properties considered are taken from [26] and are given as follows: initial concentration $C_0 = 3$ wt.%, partition coefficient $k_p = 0.14$, liquidus slope $m_l = -2.6$ K/wt.%, solute diffusion coefficient $D = 3000$ μm^2 s, melting temperature $T_m = 933.47$ K, required undercooling for nucleation $\Delta T^n = 8$ K, Gibbs–Thomson relation coefficient $\epsilon_c = 0.24$ K μm, surface tension anisotropy $\epsilon = 0.01$ and domain height $3\sqrt{2}d_s$ with $d_s = 9.7$ μm. The spacing between two potential nucleation sites is $\sqrt{2} \times d_s$, so that the density of potential nucleation sites is d_s^{-2} with a body centered structure as shown in Fig. 9. In [26], 23 cases with various solidification speeds and thermal gradients are considered using a mesh with 151 grid points in the y-direction by fully-exploiting symmetry. The uniform mesh used in [26] is approximately equivalent to the adaptive mesh used in this work with refinement level 10 and with an initial coarse mesh 20×1.

In our computation, we use a coarser adaptive mesh with refinement level 9 (smallest grid spacing is 0.08 μm). In order to demonstrate that refinement level 9 is actually fine enough for this problem, we performed a convergence study using refinement levels 5–10 (discretization in space) for solidification speed 3000 μm/s and thermal gradient 1400 K/cm. For time discretization, the CFL coefficient used for adaptive time stepping is 1/3 for all computations in this example. To investigate the convergence quantitatively, we defined the error as the root mean square of the difference in the concentration and the "true" concentration (concentration obtained using refinement level 10) at time 3.47×10^{-2} s on nodes of a 640×32 uniform mesh, i.e. $\text{err} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\frac{C_i - C_i^{\text{true}}}{3.0}\right)^2}$. As shown in Fig. 10, the nucleation pattern for refinement level 5 is significantly

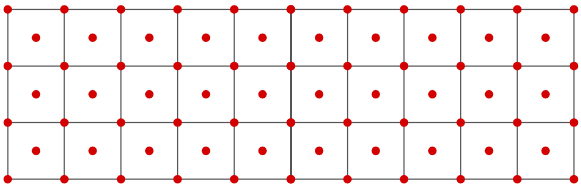


Fig. 9. Schematic of the computational domain and potential nucleation sites for the two-dimensional crystal growth.

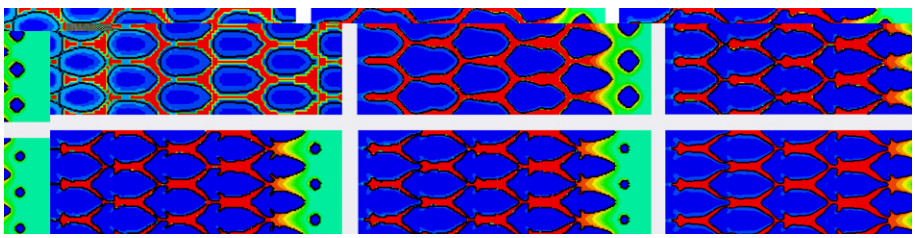


Fig. 10. Solute concentration for solidification speed 3000 μm/s and thermal gradient 1400 K/cm using adaptive meshing with refinement levels from 5 to 10 (first row: refinement level 5, 6, 7; second row: refinement level 8, 9, 10).

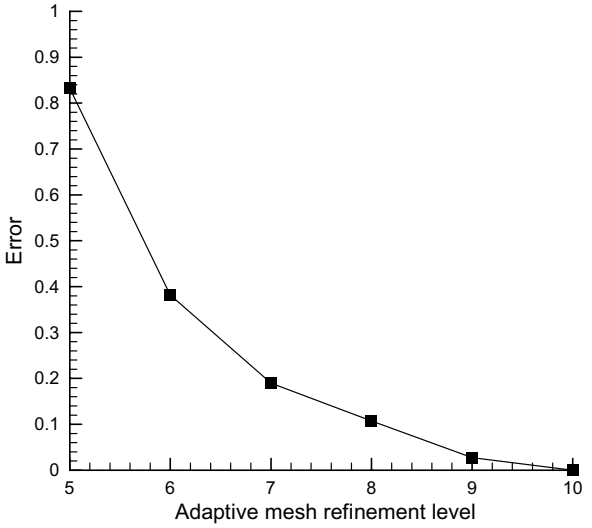


Fig. 11. Error at refinement level from 5 to 10 for solidification speed 3000 μm/s and thermal gradient 1400 K/cm.

different from the results obtained with refinement level 10, which leads to a very large error (around 0.84 as shown in Fig. 11). When the refinement level is increased to 7–9, the crystal shapes are visually very similar to the crystal shape obtained with refinement level 10. Since the error when using refinement level 9 is only about 0.04, we can assume that refinement level 9 is fine enough. For all two-dimensional computations in this problem, we use refinement level 9.

4.2.2. Investigation of CET

Here we perform the same 23 cases examined in [26] with solidification speeds and thermal gradients listed in Table 1. The computed results for the 23 cases are shown in Fig. 12. The growth types of the 23 cases (columnar, equiaxed and mixed) are plotted in Fig. 13 together with the analytical estimation of CET using Hunt's model [35] with $G = 0.666\sqrt{\rho^{n}}\Delta T^{t}\left(1-\left(\frac{\Delta T^{n}}{\Delta T^{t}}\right)^{3}\right)$ and $\Delta T^{t} = 216V^{1/2}$ [26], where G and V are the thermal gradient and pulling velocity for directional growth, respectively, ΔT^{n} is the required undercooling for nucleation and ΔT^{t} is the undercooling at the dendrite tip. In the model discussed in [26], only one phase field equation and one solute diffusion equation are solved. The temperature is not solved from the governing heat equation. Here, we take the same simplification. Initially the temperature is distributed linearly with a given thermal gradient G. The evolution of the temperature is then computed with a constant cooling rate, R, which is obtained from the given pulling velocity and thermal gradient through $R = V \times G$. Just as predicted by Hunt's model, the columnar growth is favored for low pulling velocity (solidification speed) and high temperature gradient, while the equiaxed growth is favored for high pulling velocity and low temperature gradient. Mixed growth (a transition microstructure from columnar to equiaxed) is observed for cases below but close to the analytical estimation. Notice that in these computational example, the temperature gradient considered are unrealistically large in order to observe and study the CET transition.

Table 1
Considered solidification speed (μm/s) and thermal gradient (K/cm)

#	1	2	3	4	5	6	7	8	9	10	11	12
V	3000	3000	3000	3000	3000	3000	3000	3000	6000	6000	6000	6000
G	140	1400	2800	5600	7000	8400	9800	21,000	7000	9800	21,000	39,200
#	13	14	15	16	17	18	19	20	21	22	23	
V	10,000	10,000	10,000	10,000	10,000	1000	1500	1500	2500	3500	4500	
G	7000	15,400	21,000	39,200	56,000	140	140	7000	7000	7000	7000	

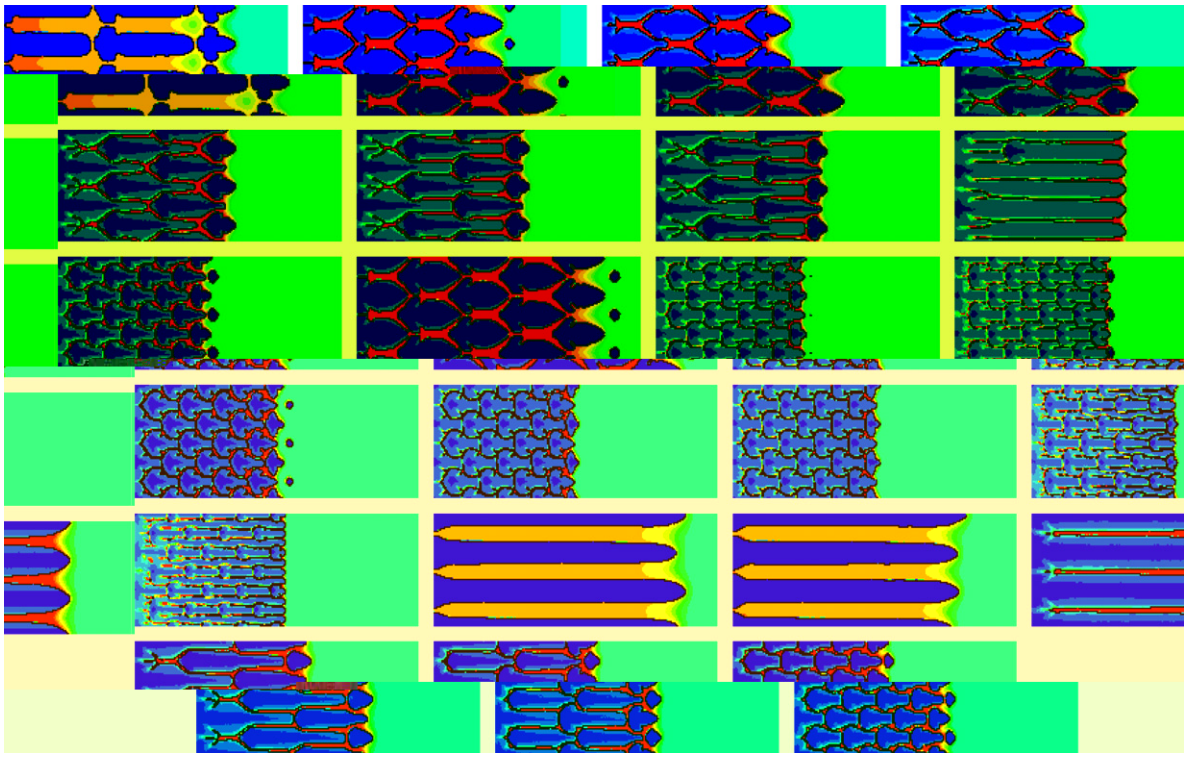


Fig. 12. Computation results showing solute concentration for the 23 cases listed in Table 1 (first row: case 1–4; second row: 5–8 and so on). Gibbs–Thomson coefficient $\epsilon_c = 0.24$ K μm.

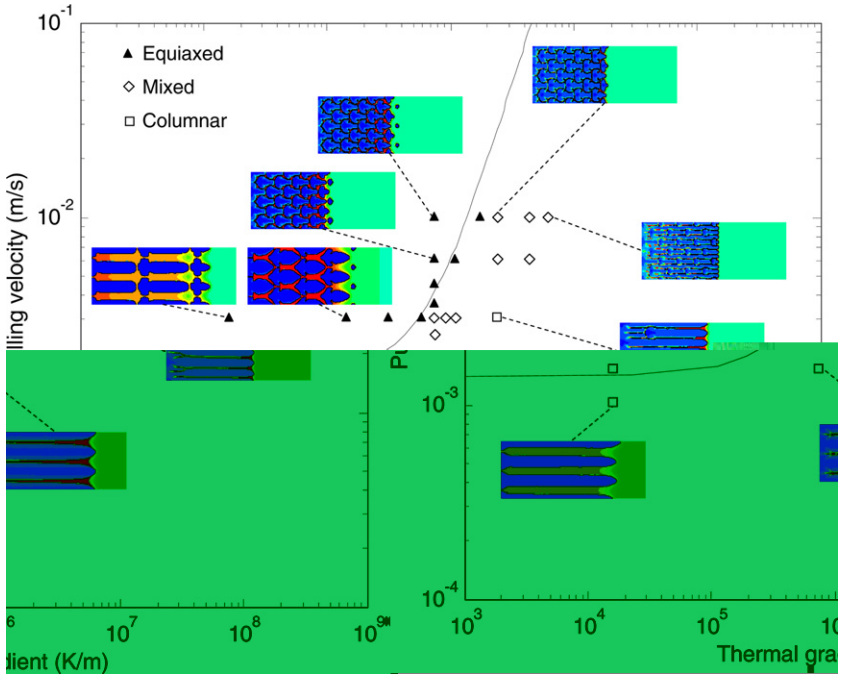


Fig. 13. CET map for directional solidification for the 23 cases listed in Table 1. Gibbs–Thomson coefficient $\epsilon_c = 0.24$ K μm.

Comparing with the results in [26], we obtain similar growth pattern. However, we did not observe secondary dendrite formation. Even with the finest mesh in the convergence study as shown in Fig. 10, secondary dendrites do not show up in the numerical solution. Since convergence of the numerical solution is observed in Fig. 10, we believe that the specified material properties will not lead to significant secondary dendrites using our mathematical model. In order to demonstrate the ability to capture secondary dendrites using the present level set method, we reduce the magnitude of Gibbs–Thomson relation coefficient ϵ_c from 0.24 K μm to 0.01 K μm with corresponding results shown in Fig. 14. The growth type of the 23 cases (columnar, equiaxed and mixed) are plotted in Fig. 15 together with the analytical estimation using Hunt's model. A comparison for case 2 is given in Fig. 16 to demonstrate the effects of ϵ_c on the formation of secondary dendrites. Comparing with the results shown in Figs. 12 and 13 with larger magnitude of the Gibbs–Thomson relation coefficient ϵ_c, we observe that the CET transition occurs at a smaller thermal gradient for a giving solidification speed.

Without utilizing symmetry to reduce the computation requirement as was the case in [26], we observe a continuous transition from equiaxed growth to columnar growth as shown in case 8 in Fig. 12, and cases 4 and 12 in Fig. 14. If symmetry is utilized, the computation requirement is much less (only 1/6) than that without using symmetry. However, if the solution symmetry is inherently different from the applied symmetry, we may obtain very different results as shown in Fig. 17. For example, the computation domain width may not be an integer times of the dendritic growth wavelength, which is related with processing and material parameters as analytically estimated using $\lambda = 2\pi\left(\frac{D\Gamma}{-Vm_1C_0}\right)^{1/2}$ [36]. As shown in the right of Fig. 17, the domain boundary restricts the dendrites from adjusting their wavelength to an optimal value. This leads to dendrite tips with slightly different shapes. However, if a larger domain is used, the dendrites can gradually adjust their wavelength leading to dendritic tips with the same shape. More importantly, the wavelength of dendritic growth is significantly smaller than the wavelength obtained without utilization of symmetry. Therefore, for more accurate solution, a larger computational domain is preferred as it can reduce the effects of the artificial computational domain boundary on the numerical solution. However, the computation time will also increase significantly with an enlarged domain. For example, the results obtained using symmetry shown in the right of Fig. 17 take around 45 min using one computation node with two 3.8 G CPUs, while the results obtained without using symmetry shown in the left of Fig. 17 take around 4.5 h with the same computation node. Selecting appropriate computational domain size is indeed an issue of balancing between the numerical solution accuracy and the required computational effort.

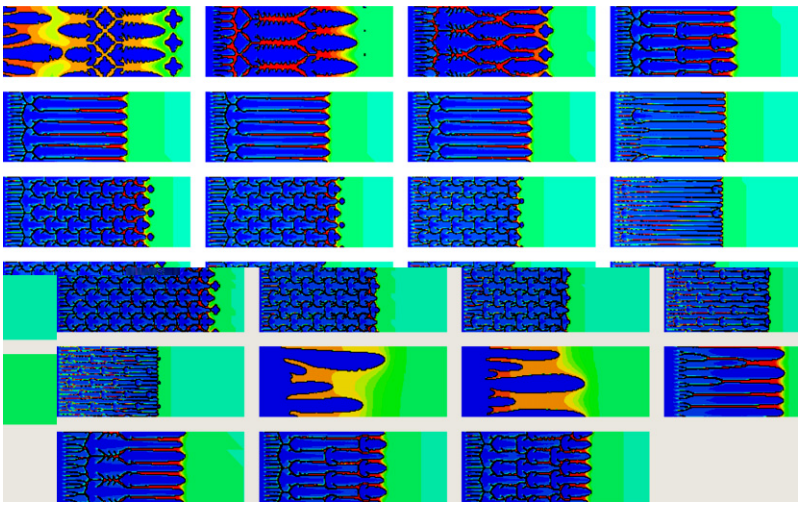


Fig. 14. Computation results showing solute concentration for the 23 cases listed in Table 1 (first row: case 1–4; second row: 5–8 and so on). Gibbs–Thomson coefficient $\epsilon_c = 0.01$ K μm.

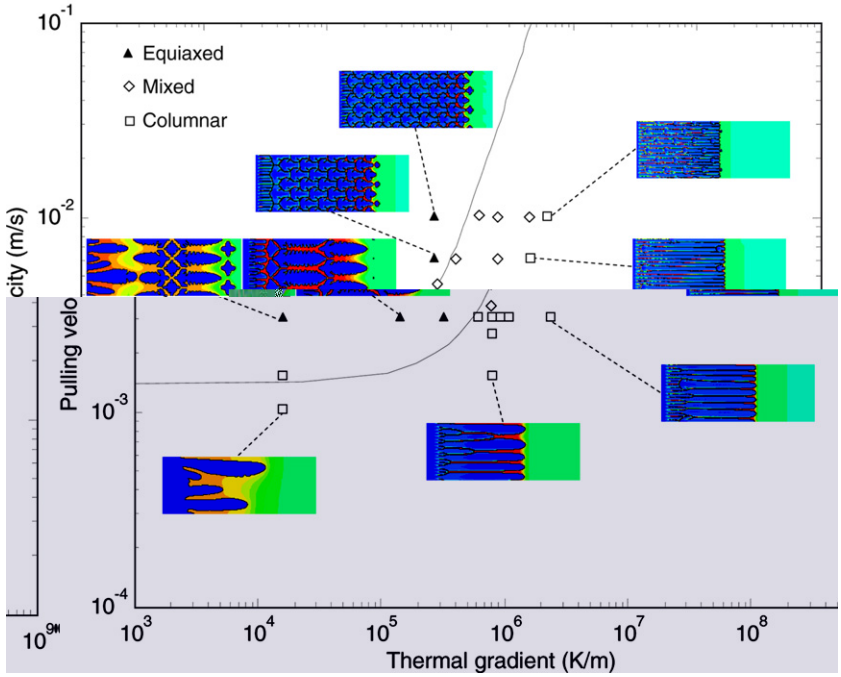


Fig. 15. CET map for directional solidification for the 23 cases listed in Table 1. Gibbs–Thomson coefficient $\epsilon_c = 0.01$ K μm.

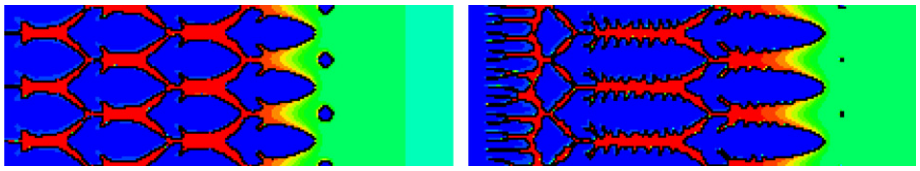


Fig. 16. Computation results showing solute concentration for case 2 with $\epsilon_c = 0.24$ K μm (left) and $\epsilon_c = 0.01$ K μm (right).

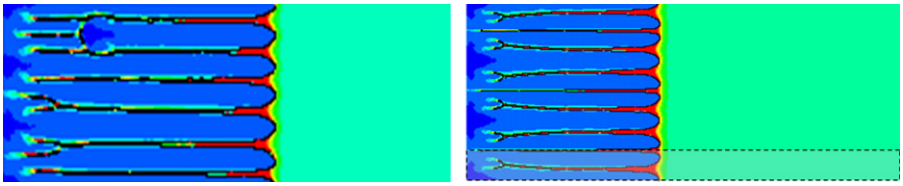
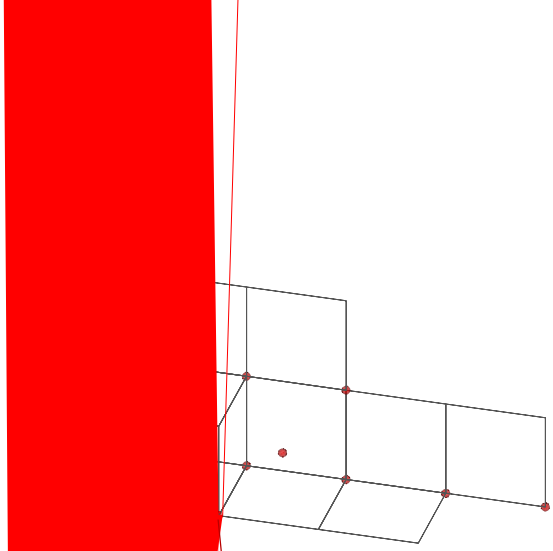


Fig. 17. Computed growth pattern showing solute concentration without using symmetry (left) and with symmetry (right) for the case 8 listed in Table 1. The box with dashed line shown in the right figure is the computation domain when using symmetry.

4.2.3. Multiple three-dimensional crystal interaction

As we know, crystal growth is inherently three-dimensional. Here, we perform a 3D study of the interaction between multiple crystals. Material properties are selected to be the same as in the 2D study with Gibbs–Thomson coefficient $\epsilon_c = 0.24$ K μm. However, the spacing between two potential nucleation sites is taken as $\sqrt[3]{2} \times d_s$, so that the density of potential nucleation sites is d_s^{-3} with a body centered structure as shown in Fig. 18. As computation in 3D is more intensive than in 2D, we fully-utilize symmetry. Grid spacing is selected as 0.05 μm (grid spacing 0.08 μm was used for the 2D example). For solidification speed 3000 μm/s and thermal gradient 1400 K/cm, we obtain the 3D crystal shape as shown in Fig. 19. The solute concentration at cross sections $x = 0$ and $x = z$ is shown in Fig. 20. In comparison to the two-dimensional computation



(Fig. 7), the requi ndercooling for nucleation 8 K is never obtained. Therefore, nucleation is not observed. In three-dimensio he solute rejected from the dendrite tips can diffuse away in various directions around the dendrite tip. S e dendrite tip velocity may be able to 'catch up' with the pulling velocity for a given thermal gradient, ce the rejected solute is easier to diffuse away comparing with the two-dimensional situation and increas e equilibrium temperature on the interface. Therefore, the maximum undercooling in front of the dend tips is not increasing as the dendrites grow, while in the two-dimensional growth, the diffusion of solute ejected from dendrite tips is confined to either left of right direction of the dendrite tip. Therefore, the de rite tip velocity may not be as large as the pulling velocity if the thermal gradient is not

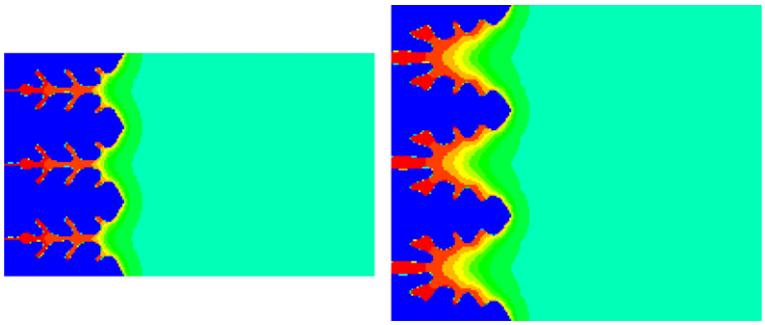


Fig. 20. Solute concentration at cross sections $x = 0$ (left) and $x = z$ (right) for 3D crystal growth with solidification speed 3000 μm/s and thermal gradient 1400 K/cm.

sufficiently large. This further leads to increasing the maximum undercooling in front of the dendrite tips until nucleation occurs as the maximum undercooling is more than ΔT^{n}.

In order to observe nucleation phenomena in three-dimensions, we reduced the required undercooling for nucleation from 8 K to 7 K with results shown in Figs. 21 and 22. As shown in the cross section $x = z$ (Fig. 22), the pattern of solute concentration has a similarity with the results obtained in two-dimensions. However, the pattern of solute concentration in the cross section $x = 0$ is quite different from the two-dimensional results. Similarly to the 2D computation, if the thermal gradient is increased from 1400 K/cm to 21,000 K/cm, the growth pattern will convert to columnar growth as shown in Figs. 23 and 24. The three-

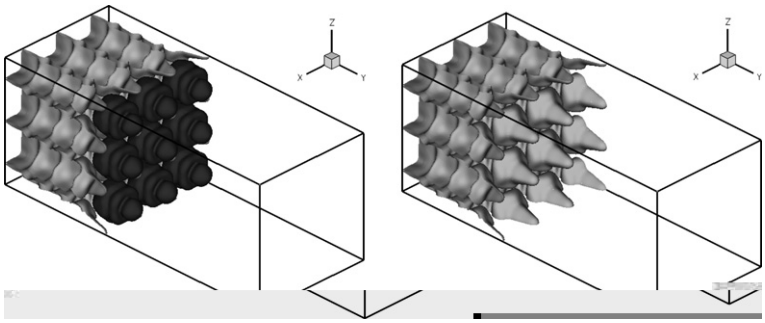


Fig. 21. Three-dimensional crystal growth with required undercooling for nucleation 7 K (solidification speed 3000 μm/s and thermal gradient 1400 K/cm). The right figure is the same plot as the left figure without plotting the nucleated crystal in the dendrite front.

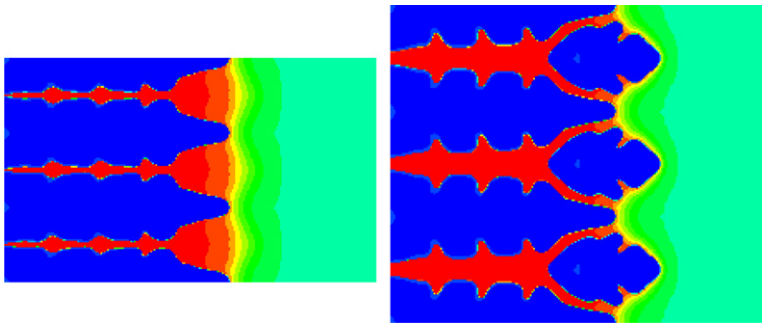


Fig. 22. Solute concentration at cross sections $x = 0$ (left) and $x = z$ (right) for 3D crystal growth with required undercooling for nucleation 7 K (solidification speed 3000 μm/s and thermal gradient 1400 K/cm).

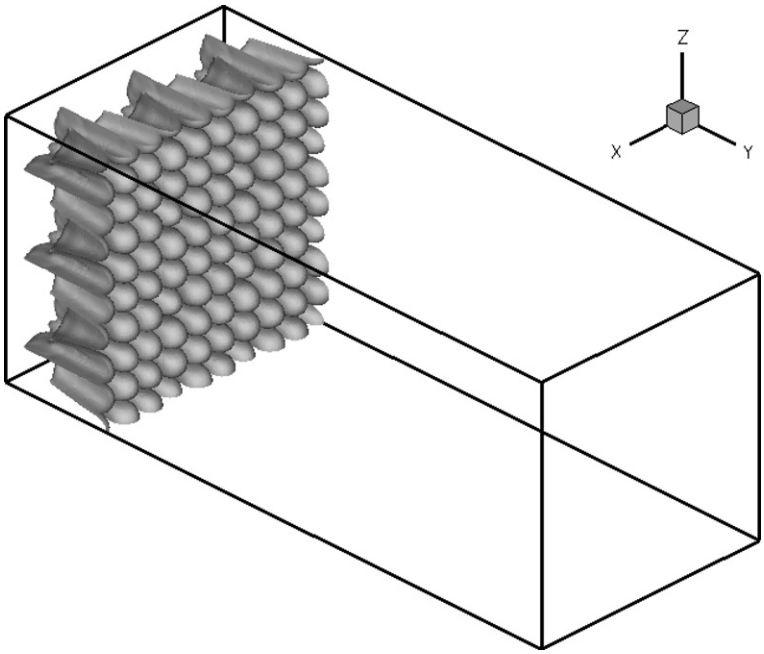


Fig. 23. Three-dimensional crystal growth with solidification speed 3000 μm/s and thermal gradient 21,000 K/cm.

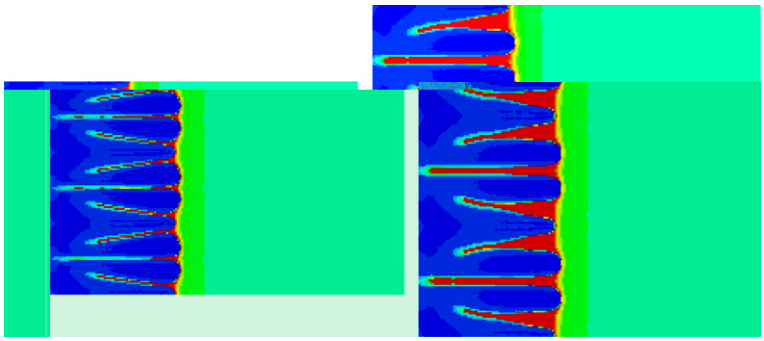


Fig. 24. Solute concentration at cross sections $x = 0$ (left) and $x = z$ (right) for 3D crystal growth with solidification speed 3000 μm/s and thermal gradient 21,000 K/cm.

dimensional computation in these two cases shows that the two-dimensional computational results can serve as a tool to qualitatively but not quantitatively understand realistic 3D crystal growth.

4.2.4. Effects of randomness on crystal orientation

In the above computation, we have assumed that all crystals have the same orientation to allow us to compare with the results in [26]. However, from the first numerical example studying interaction between nine crystals, we have found that crystal orientation plays an important role during multiple crystal growth. Here we incorporate the effects of randomness in crystal orientation (uniformly distributed from 0 to 2π) for cases 2 and 6 of Table 1. The Gibbs–Thomson relation coefficient is selected to be 0.01 K μm as in the computation of the 23 cases with reduced Gibbs–Thomson relation coefficient. All other parameters are selected to be the same. Due to the randomness in crystal orientation, symmetry is broken and more dendrites have to be included in the numerical solution to obtain the growth pattern. Therefore, we increased the domain height from $3\sqrt{2}d_s$ to $9\sqrt{2}d_s$.

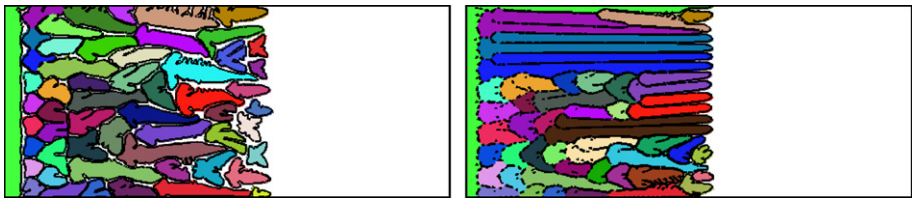


Fig. 25. Computed growth pattern for cases 2 (left) and 6 (right) with consideration of randomness in crystal orientation. The colors used show crystal orientations.

Computation results of cases 2 and 6 are shown in Fig. 25. In comparing with the results shown in Fig. 14 we conclude that randomness in crystal orientation has a great effect on the interaction between crystals. If the nucleated crystal orientation is 'lucky' to be almost the same as the growth direction, then the crystal may grow very long before its growth is blocked by other crystals. In general, crystals with orientation close to the growth direction are favored in growth. The slight difference in orientation angle between nearby dendrites may lead to increasing gap, which gives room for nucleation of new dendrites, or decreasing gaps, which will eventually block the growth of one dendrite. This phenomenon is observed in Fig. 26 at two different time levels.

4.3. Interaction between crystals with consideration of randomness in required undercooling for nucleation

In the previous example, we considered the effects of randomness in crystal orientation, while the required undercooling for nucleation was assumed to be constant (8 K). With this assumption, a potential nucleation site will never nucleate if the actual undercooling is less than 8 K. A consequence of this is that there is little variety of microstructure size even though the processing parameters, such as velocity and thermal gradient are significantly different. For example, at very low thermal gradient, we obtain slim equiaxed microstructure as shown in the results of case 1 in the previous example. However in practice, such slim equiaxed microstructure is not common. Instead, coarse equiaxed microstructure is often observed, which is known as the columnar-to-equiaxed transition and usually appears in the center of castings.

In the previous example, the temperature field was not solved. A cooling rate $R = V \times G$ was directly applied. This assumption also limits the application to directional growth where a temperature gradient can be controlled. In many other processes e.g. sand casting, the temperature (or heat flux) is only controlled at the domain boundary instead of the whole domain. To give results with more practical importance, here we perform a numerical study without these assumptions by using random undercooling for nucleation and by solving the heat equation. The following dimensionless material properties are considered: potential nucleation site density $\rho^{\mathrm{n}} = 100$, required undercooling for nucleation ΔT^{n} is random with distribution $N(1.5,0.2^2)$, density $\rho = 1$, heat capacity $c = 1$, heat diffusion coefficient $k = 1$, latent heat $L = 100$, Lewis number $Le = 100$, melting temperature $T_{\mathrm{m}} = 0$, initial concentration $C_0 = 0.1$, liquidus slope $m_{\mathrm{l}} = -10$, partition coefficient $k_{\mathrm{p}} = 0.1$, Gibbs–Thomson relation with kinetic undercooling $T_* = T_{\mathrm{m}} + m_{\mathrm{l}} C^{\mathrm{l}} - \frac{2}{3}(1 - \cos(4(\theta - I)))V$. A computation domain with size 10×2.5 is used. Initially, the whole domain is liquid.

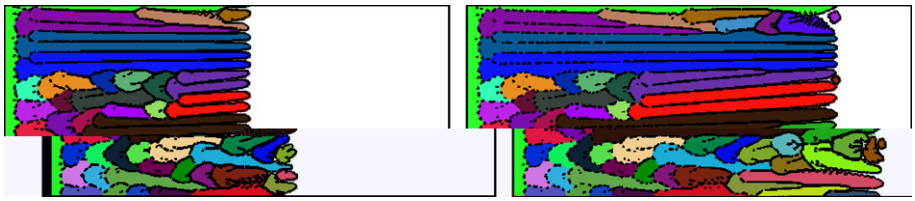


Fig. 26. Computed growth pattern for case 6 with consideration of randomness in crystal orientation at time step 6000 and 8000. The colors used show crystal orientations.

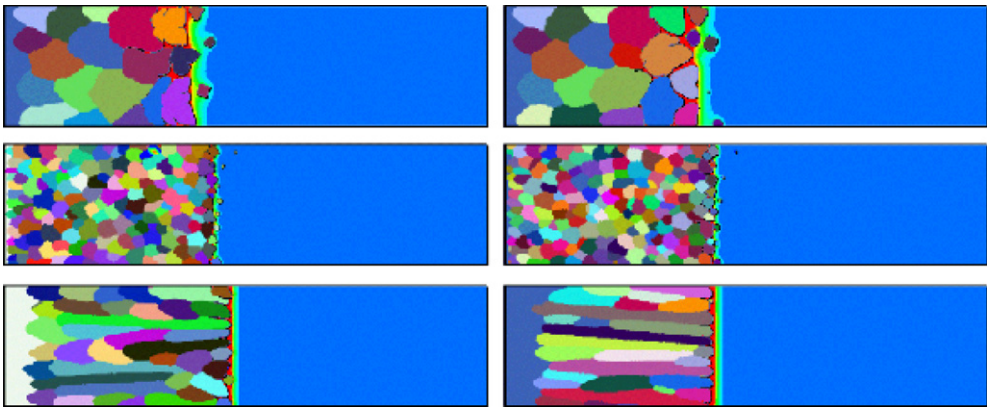


Fig. 27. Computed growth pattern on a domain of 10×2.5 at various conditions. Left: grid spacing 0.0098; Right grid spacing 0.0049. First row: $V = 0.1$, $G^{\mathrm{l}} = 0.01$; Second row: $V = 0.35$, $G^{\mathrm{l}} = 0.01$; Third row: $V = 0.137$, $G^{\mathrm{l}} = 30.0$. The colors used show crystal orientations.

The temperature in the whole domain is distributed with thermal gradient $G = G^{\mathrm{l}} \exp\left[\frac{-Vx}{\alpha}\right]$ with left side temperature at $T_{\mathrm{m}} + m_{\mathrm{l}} C^0$. At the left boundary, a heat flux is withdrawn from the domain with magnitude $q_{\mathrm{s}} = kG^{\mathrm{s}} \exp\left[\frac{-V(0-Vt)}{\alpha}\right]$, while at the right boundary, a heat flux enters the domain with magnitude $q_{\mathrm{l}} = kG^{\mathrm{l}} \exp\left[\frac{-V(10-Vt)}{\alpha}\right]$. Here, $\alpha \equiv \frac{k}{\rho c} = 1$ and $V = \frac{k(G^{\mathrm{s}} - G^{\mathrm{l}})}{\rho L}$. The reason to select such initial and boundary conditions is that it leads to uniform microstructure throughout the computational domain. Adaptive mesh with smallest grid spacing 0.0098 is used to obtain crystal pattern as shown in the left of Fig. 27. A computation domain of 10×2.5 is used here. Three combinations of V and G^{l} $((V, G^{\mathrm{l}}) = (0.1, 0.01), (0.35, 0.01), (0.137, 30.0))$ are studied with results shown in Fig. 27.

At low solidification speed and low thermal gradient ($V = 0.1$, $G^{\mathrm{l}} = 0.01$), we observe coarse equiaxed crystal growth, which is not captured in the previous example. By reducing the grid spacing by a half to 0.0049, we obtain different microstructures as shown in the left and right of Fig. 27. Due to randomness in crystal orientation angle and in required undercooling for nucleation, numerical noise may lead to very different microstructure details. However, the obtained microstructures with both grids are very similar in their patterns. This suggests that our model can be used to simulate microstructure evolution and predict microstructure patterns, even though there is always error in the numerical solution. The capability of investigating the interaction between tens of crystals with demonstration of convergence is an improvement in numerical study of crystal growth. However, hundreds or thousands of crystals are often involved in reality even for a small piece of casting. Based on the previous convergence study for interaction between a small number of crystals, we here demonstrate the ability of the current approach to capture interaction between hundreds of crystals by using the same material and a larger computational domain of size 80×80 (the left-top quarter is not included). The relation of the computational domain and actual domain is demonstrated in Fig. 28. Initially, the whole domain is at temperature 30. A cooling rate of 5 is applied at all computational domain boundaries

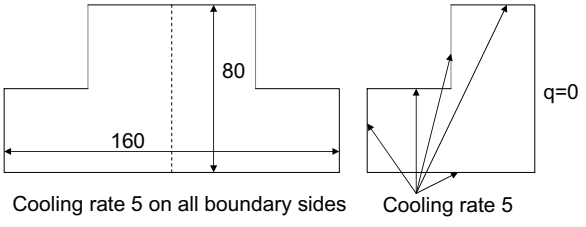


Fig. 28. Schematic of the actual domain (left) and computational domain (right).

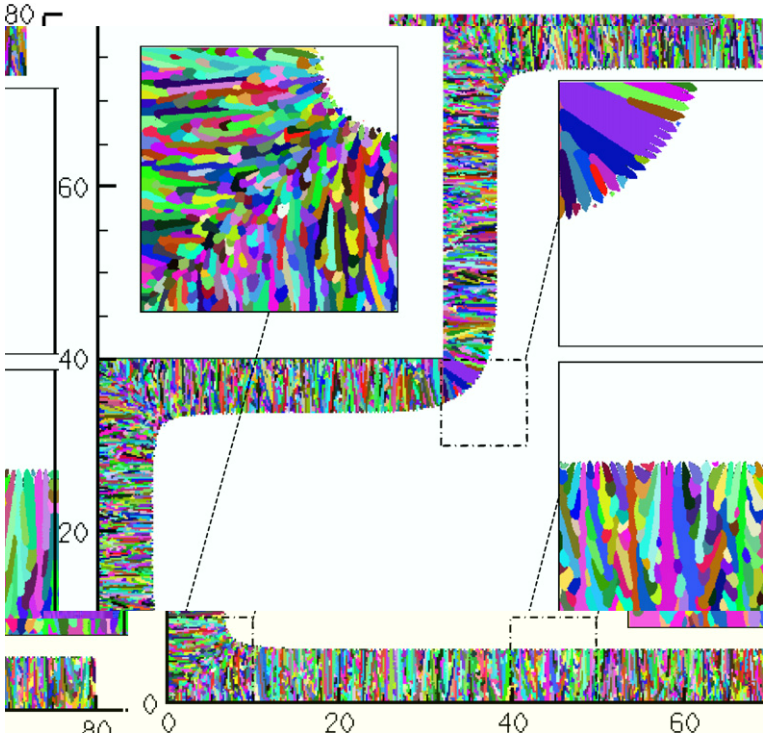


Fig. 29. Microstructure pattern for growth of hundreds of crystals in a two-dimensional domain. The colors used show crystal orientations.

except on the right side, where the heat flux is 0. Using the same grid spacing 0.0098 and a fixed time step of 7×10^{-3}, we obtain the microstructure at time 45 shown in Fig. 29. At the corner, equiaxed microstructure is obtained, while at places far away from the corner, a mixture of columnar growth and equiaxed growth is obtained. At the center of the domain, columnar growth dominates. Since hundreds of crystals are involved as shown in Fig. 29 the computation of this problem is very intensive. It takes around 24 h using 16 nodes each with two 3.8 G CPUs to obtain the results in Fig. 29.

5. Conclusions

Our previous numerical work based on the level set method [22], which combines features of both front-tracking methods and fixed domain methods, was extended in this work to study interaction between multiple dendrites during solidification. By using markers to identify different crystals, we are able to study the interaction between multiple dendrites with a single signed distance function. Accuracy of our approach is demonstrated with convergence studies and comparison with the multi-level set method. Simulations of columnar to equiaxed transition are performed and compared with recent phase field results available in the literature [26]. New abilities provided by the current numerical approach include extension to three-dimensions and arbitrary crystallographic orientations. We find that the three-dimensional growth is significantly different from the two-dimensional growth for one of the two cases considered. However, at other considered cases with high thermal gradient, the microstructure patterns obtained with two- and three-dimensional modeling are both columnar growth. Randomness in crystallographic orientations was found to have a significant effect on the results. Efficiency of the multi-level set algorithm is demonstrated in an example that includes hundreds of crystals with consideration of randomness effects in both crystallographic orientation and required undercooling for nucleation. The algorithms developed here are currently being used as localized subgrid models for the development of efficient multiscale models of solidification processes.

References

[1] C.A. Gandin, M. Rappaz, A 3D cellular automation algorithm for the prediction of dendritic grain growth, Acta Mater. 45 (1997) 2187–2195.
[2] K. Kremeyer, Cellular automata investigations of binary solidification, J. Comput. Phys. 142 (1998) 243–262.
[3] H. Rafii-Tabar, A. Chirazi, Multi-scale computational modelling of solidification phenomena, Phys. Rep. 365 (2002) 145–249.
[4] W. Wang, P.D. Lee, M. McLean, A model of solidification microstructures in nickel-based superalloys: predicting primary dendrite spacing selection, Acta Mater. 51 (2003) 2971–2987.
[5] D. Juric, G. Tryggvason, A front-tracking method for dendritic solidification, J. Comput. Phys. 123 (1996) 127–148.
[6] P. Zhao, J.C. Heinrich, Front-tracking finite element method for dendritic solidification, J. Comput. Phys. 173 (2001) 765–796.
[7] P. Zhao, J.C. Heinrich, Numerical approximation of a thermally driven interface using finite elements, Int. J. Numer. Methods Eng. 56 (2003) 1533–1547.
[8] N. Al-Rawahi, G. Tryggvason, Numerical simulation of dendritic solidification with convection: three-dimension flow, J. Comput. Phys. 194 (2004) 677–696.
[9] A. Wheeler, W. Boettinger, G. McFadden, Phase-field model for isothermal phase transitions in binary alloys, Phys. Rev. A 45 (1992) 7424–7440.
[10] 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.
[11] A. Karma, W.J. Rappel, Phase field model for computationally efficient modelling of solidification with arbitrary interface kinetics, Phys. Rev. E 53 (1996) 3017–3020.
[12] A. Karma, W.J. Rappel, Quantitative phase-field modelling of dendritic growth in two and three dimensions, Phys. Rev. E 57 (1998) 4323–4349.
[13] N. Provatas, N. Goldenfeld, J. Dantzig, Efficient computation of dendritic microstructures using adaptive mesh refinement, Phys. Rev. Lett. 80 (1998) 3308–3311.
[14] 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.
[15] M. Plapp, A. Karma, Multiscale random-walk algorithm for simulating interfacial pattern formation, Phys. Rev. Lett. 84 (2000) 17401–17443.
[16] J. Jeong, N. Goldenfeld, J. Dantzig, Phase field model for three-dimensional dendritic growth with fluid flow, Physical Review E 64 (2001) 041602.
[17] J.A. Warren, W.L. George, A parallel 3D dendritic growth simulator using the phase-field method, J. Comput. Phys. 177 (2002) 264–283.
[18] W.J. Boettinger, J.A. Warren, C. Beckermann, A. Karma, Phase-field simulation of solidification, Annu. Rev. Mater. Res. 32 (2002) 163–194.
[19] S. Chen, B. Merriman, S. Osher, P. Smereka, A simple level set method for solving Stefan problems, J. Comput. Phys. 135 (1997) 8–29.
[20] Y.T. Kim, N. Goldenfeld, J. Dantzig, Computation of dendritic microstructures using a level set method, Phys. Rev. E 62 (2000) 2471–2474.
[21] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Sci. Comput. 19 (2003) 183–199.
[22] 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.
[23] N. Zabaras, B. Ganapathysubramanian, L. Tan, Modeling dendritic solidification with melt convection using the extended finite element method (XFEM) and level set methods, J. Comput. Phys. 218 (2006) 200–227.
[24] L. Tan, N. Zabaras, A level set simulation of dendritic solidification of multi-component alloys, J. Comput. Phys. 221 (2007) 9–40.
[25] M. Rappaz, C.A. Gandin, Probabilistic modelling of microstructure formation in solidification processes, Acta Metall. 41 (1993) 345–360.
[26] A. Badillo, C. Beckermann, Phase-field simulation of the columnar-to-equiaxed transition in alloy solidification, Acta Mater. 54 (2006) 2015–2026.
[27] N. Al-Rawahi, G. Tryggvason, Numerical simulation of dendritic solidification with convection: two-dimension geometry, J. Comput. Phys. 180 (2002) 471–496.
[28] Y. Jaluria, K.E. Torrance, Computational Heat Transfer, second ed., Taylor & Francis, 2003, pp. 49–54.
[29] H. Zhao, T. Chan, B. Merriman, S. Osher, A variational level set approach to multiphase motion, J. Comput. Phys. 127 (1996) 179–195.
[30] B. Merriman, J. Bence, S. Osher, Motion of multiple junctions: a level set approach, J. Comput. Phys. 112 (1994) 334–363.
[31] Tecplot: Plotting and Data Visualization Software – Tecplot 10. http://www.tecplot.com/.
[32] K. Schloegel, G. Karypis, V. Kumar, ParMetis: parallel static and dynamic multi-constraint graph partitioning, Concurrency and Computation: Practice and Experience 14 (3) (2002) 219–240.
[33] PETSc: http://www-unix.mcs.anl.gov/petsc/petsc-as/index.html.
[34] A. Spittle, S. Brown, Computer simulation of the effects of alloy variables on the grain structures of castings, Acta Metall. 37 (1989) 1803–1810.
[35] J.D. Hunt, Steady state columar and equiaxed growth of dendrites and eutectic, Mater. Sci. Eng. 65 (1984) 75–83.
[36] W. Kurz, D.J. Fisher, Fundamentals of solidification, third ed., Trans Tech Publications Ltd, Switzerland, 1989.