Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2007) 728–754
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

Multiscale modeling of alloy solidification using a database approach

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 20 March 2007; received in revised form 14 August 2007; accepted 16 August 2007
Available online 31 August 2007

Abstract

A two-scale model based on a database approach is presented to investigate alloy solidification. Appropriate assumptions are introduced to describe the behavior of macroscopic temperature, macroscopic concentration, liquid volume fraction and microstructure features. These assumptions lead to a macroscale model with two unknown functions: liquid volume fraction and microstructure features. These functions are computed using information from microscale solutions of selected problems. This work addresses the selection of sample problems relevant to the interested problem and the utilization of data from the microscale solution of the selected sample problems. A computationally efficient model, which is different from the microscale and macroscale models, is utilized to find relevant sample problems. In this work, the computationally efficient model is a sharp interface solidification model of a pure material. Similarities between the sample problems and the problem of interest are explored by assuming that the liquid volume fraction and microstructure features are functions of solution features extracted from the solution of the computationally efficient model. The solution features of the computationally efficient model are selected as the interface velocity and thermal gradient in the liquid at the time the sharp solid–liquid interface passes through. An analytical solution of the computationally efficient model is utilized to select sample problems relevant to solution features obtained at any location of the domain of the problem of interest. The microscale solution of selected sample problems is then utilized to evaluate the two unknown functions (liquid volume fraction and microstructure features) in the macroscale model. The temperature solution of the macroscale model is further used to improve the estimation of the liquid volume fraction and microstructure features. Interpolation is utilized in the feature space to greatly reduce the number of required sample problems. The efficiency of the proposed multiscale framework is demonstrated with numerical examples that consider a large number of crystals. A computationally intensive fully-resolved microscale analysis is also performed to evaluate the accuracy of the multiscale framework.
© 2007 Elsevier Inc. All rights reserved.

Keywords: Multiscale modeling; Solidification; Database approach; Level set method; Crystal growth

* 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.2007.08.016

1. Introduction

Alloy solidification is multiscale in nature. Interaction between thousands or even millions of crystals gives the overall behavior of the solidification process and defines the properties of the final product. Investigating the interaction and growth of crystals in the microscale is computationally very intensive, whereas macroscale models lack accuracy since they are based on a large number of simplifications. Multiscale modeling by coupling macroscopic and microscopic models allows us to take advantage of both the efficiency of the macroscopic models and the accuracy of the microscopic models. There are mainly three types of multiscale techniques for solidification used in the literature: (1) resolving microscale details in an efficient way, such as using an adaptive mesh refinement [1–4] or an adaptive moving grid [5], since the main complexity in the microscale arises from a moving solid–liquid interface, (2) performing analytical studies [6] or simple numerical computation [7] in the microscale and passing the required information to the macroscale computation, and (3) designing multiscale algorithms driven by microscopic numerical solution data, e.g. regression fit [8] and subgrid modeling [9].

By combining features of front tracking and fixed domain methods, we recently demonstrated the ability to simulate interaction between hundreds of randomly nucleated crystals by utilizing adaptive mesh refinement, the level set method and parallel computation [4]. Despite this improvement, adaptive techniques [1–4] are in nature microscale models that are not practical for investigating interaction between far more than a few hundreds of crystals.

An alternative multiscale technique performs analytical study in the microscale with certain simplifications to provide microstructural information to the macroscale model. Ref. [6] models the solidification system as solid phase, inter-dendritic and extra-dendritic liquid phases. The macroscopic transport equations for these three phases are derived using volume-averaging technique and closed by supplementary relations, which are obtained from analysis in the microscale. In [6], there is no numerical computation performed at the microscale. But in some other studies (e.g. [7]), the microscopic problems are not avoided completely. Assuming periodic distribution of crystals, numerical computation of a single crystal growth is carried out for every point of a macroscopic grid to provide information for the macroscale computation. Although the numerical computation is reduced, the assumption of periodicity is physically unrealistic. This limits its application to only equiaxed growth. Without the ability to capture important physics in the microscale (e.g. randomness in nucleation and crystal orientation), both of these methods [6,7] are phenomenological in nature regarding their prediction for example of the microstructure type and size.

Multiscale algorithms driven by data provide the potential for investigating the interaction between thousands or even millions of crystals. The microscale model tries to accurately capture important physics. Computation with the microscale model is then used to provide necessary data for macroscale computation. In [8], microscale computations are used to obtain data for regression fit of a predictive equation, which is further used for macroscale computation. A disadvantage of this method is that the pre-assumed predictive equation has a simple form. In [9], the idea of subgrid modeling is used for multiscale modeling of solidification. The computational domain is divided into a number of small sub-domains, so that the simulation time can be greatly reduced. However, this method is still computationally very intensive. Qualitatively, similar macroscopic conditions would lead to similar microstructure. In subgrid modeling, this similarity is not taken account to reduce the required computational effort. In this work, we will quantitatively explore the similarity between microscale computations.

The multiscale method used in this work falls into the heterogeneous multiscale method (HMM) framework [10], which aims at designing combined macroscopic–microscopic computational methods that are much more efficient than solving the full-microscopic model and at the same time lead to a desired level of accuracy. The idea of building a database based on results from microscopic simulation of selected problems and using this database for multiscale computation is very straightforward and has been widely used in many research areas. The database approach has not yet been fully explored for solidification processes. The complex nature of solidification process causes difficulty in finding relevant problems to build the database, and selecting results of relevant problems from the database to solve the problem of interest. Our emphasis in this work is on addressing some simple algorithms for handling these difficulties. The plan of this paper is as follows: Section 2 discusses the microscale model and the needed assumptions to reach

a macroscale model with two unknown functions. The strategy of selecting relevant sample problems for building the database and algorithms to use the sample problem data in the database for the solution of the problem of interest are presented in Section 3. Section 4 discusses the techniques for speeding up computation and quantifying microstructure features. Computational examples are provided in Section 5. At first, a solidification example is considered that allows comparison with the solution obtained from a time consuming fully-resolved microscale model. Finally, an example of the solidification of an Al–Cu alloy is provided to demonstrate the capability of the database approach to perform multiscale analysis for realistic alloy systems.

2. Mathematical model

The difficulty associated with modeling of solidification processes arises from the morphological complexity of the resulting microstructure and the variety of length scales in the system. Under typical solidification conditions, the system and interfacial structures are of the orders of 10^{-1} and 10^{-5} m, respectively.

Microscopic models (e.g. [1,3,11]) are developed to capture physical phenomena in the length scale of the interfacial structures. As pointed out in [6], there are three disparate microscopic length scales (decreasing in order): (1) the overall size of crystals (e.g. primary arm spacing λ_1 for columnar growth), (2) the secondary dendrite arm spacing and (3) the radius of a dendrite tip. The domain size in these microscale studies is usually chosen to contain one or more crystals/dendrites. The grid spacing in the discretization of the computational domain is chosen to be less than the tip radius (therefore also less than the overall size of the crystals and the secondary dendrite arm spacing).

For macroscale models (e.g. [12]), macroscopic variables including velocity, temperature and concentration are defined as the average values of microscopic variables within an averaging volume. The averaging volume is selected [13] such that the scale it represents is small enough to capture the global effects such as fluid flow, heat transfer and species distribution, but large enough to smooth out the details of the morphological complexities, inter-dendritic fluid flow, latent heat release and species redistribution. Therefore, for macroscale modeling of solidification, we typically need to select the averaging volume to be of the size of a few crystals.

In this work, we will investigate the interaction of multiple crystals using a microscale model with a fine mesh (with grid spacing small enough to capture interfacial structures such as secondary dendrite arm spacing and dendrite tip radius). With the microscale model, variation of temperature and concentration is observed in the range of crystal size. After averaging the microscale model results, information will be extracted to aid computation in a coarse-mesh using the macroscale model. The microscale model will also be utilized to validate the macroscale model results.

The following simplifications are taken for both the microscale and macroscale models:

(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.
(3) Solute diffusion in the solid phase is neglected.

2.1. Microscale model

The following governing equations for the temperature, θ, and solute concentration in the liquid, C^l, are used for modeling solidification in the microscale:

$$\rho c \frac{\partial \theta(\mathbf{x},t)}{\partial t} = k\nabla^2\theta(\mathbf{x},t), \quad \mathbf{x} \in \Omega, \tag{1}$$

$$\frac{\partial C^l(\mathbf{x},t)}{\partial t} = D\nabla^2 C^l(\mathbf{x},t), \quad \mathbf{x} \in \Omega^l. \tag{2}$$

Solving the above microscale model is numerically nontrivial due to the existence of the moving crystal/liquid interface. Due to the phase transformation, solute is rejected from the solid to the liquid leading to a solute rejection flux at the interface, $D\frac{\partial C^{\mathrm{l}}}{\partial \mathbf{n}} = -(1-k_{\mathrm{p}})C^{\mathrm{l}}V\mathbf{n}$. The temperature on the interface, θ_{I}, satisfies the Gibbs–Thomson relation: $\theta_{\mathrm{I}} = \theta_{\mathrm{m}} + m_{\mathrm{l}}C^{\mathrm{l}} + \varepsilon_{\mathrm{c}}(\mathbf{n})\kappa + \varepsilon_{\mathrm{V}}(\mathbf{n})V$, where $\mathbf{n}$ and κ are the interface normal direction and curvature, respectively. Finally, the velocity of the solid–liquid interface is governed by the heat flux jump (classical Stefan condition), $V = \frac{q^{\mathrm{s}}-q^{\mathrm{l}}}{\rho L}$.

A level set equation is used here to implicitly describe the evolution of the solid–liquid interface:

$$\frac{\partial \phi}{\partial t} + V\|\nabla\phi\| = 0. \tag{3}$$

The signed distance function $\phi(x,t)$ is simply the distance of location x to the interface at time t but with '+' sign, if x is the liquid phase and '−' sign if x is in the solid phase. According to this definition, $\phi = 0$ represents the solid–liquid interface.

Nucleation is a very complicated phenomenon related with the type and amount of impurities in the material. We denote the number of impurities per unit volume, i.e. the density of impurities, as ρ^n. Suppose that the domain of interest has a volume of V, then there will be $\rho^n V$ potential nucleation sites with each impurity serving as a potential nucleation site. For each potential nucleation site i, we model three random variables, the potential location for nucleation x_i, the required undercooling for nucleation $\Delta\theta_i^n$ and the orientation I_i of the nucleated crystal. These variables are defined below:

(1) x_i is the location of the potential nucleation site. We assume that x_i is equiprobable at each point of the whole domain.
(2) $\Delta\theta_i^n$ is the required undercooling for the potential nucleation site i to nucleate and become a crystal seed. If and only if the undercooling at the potential nucleation site i is greater than $\Delta\theta_i^n$ (i.e. $\theta_{\mathrm{m}} + m_{\mathrm{l}}C_i - \theta_i \geqslant \Delta\theta_i^n$, where θ_{m} is the melting temperature, m_{l} is the liquidus slope, C_i and θ_i are concentration and temperature at potential nucleation site i), site i becomes a crystal seed. We assume that $\Delta\theta_i^n$ 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. We only consider here the randomness of orientation angle in two dimensions, where 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 (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 our microscale model. In other words, the orientation of the crystal, I, is assumed to be fixed during growth of the crystal at the value it attains during nucleation (i.e. I_i if it is nucleated from potential nucleation site i).

Implementation of the above microscale model has been discussed in [4,11] with demonstration of convergence and accuracy for the computed solutions. In this work, we assume that the microscale model gives the true solution. Focus will be on multiscale modeling to obtain for our problem of interest the same results as obtained using the above microscale model but with significantly less computational effort.

2.2. Macroscale model

In volume-averaging models (e.g. [13]), a volume size is specified to perform averaging and obtain macroscopic variables. This averaging volume needs to be at appropriate size. If the averaging volume is too small, there may be not enough microscopic data to smooth out the local variation. On the other hand, if the averaging volume is too large, the global variation may not be accounted. Using the size of a few crystals as the averaging volume size might be appropriate. However, crystal sizes usually vary significantly even in the same solidification system. In this work, we will employ a different approach that avoids using an averaging volume for defining macroscopic variables.

Let us first define the solution of the microscale model (microscopic variables), i.e. temperature, $\theta(x,t)$, concentration, $C(x,t)$, orientation, $I(x,t)$ and signed distance $\phi(x,t)$ to the solid/liquid interface at location x and

time t. Computation of the microscopic variables is not deterministic because of the randomness in nucleation during solidification as discussed earlier in the microscale model. We define the expectation of the microscopic variables $\theta(x,t)$, $C(x,t)$, and $\phi(x,t)$ as the macroscopic temperature, macroscopic concentration and liquid volume fraction as follows:

$$T(x,t) \equiv \langle \theta(x,t) \rangle, \tag{4}$$

$$\bar{C}(x,t) \equiv \langle C(x,t) \rangle, \tag{5}$$

$$f(x,t) \equiv \langle H(\phi(x,t)) \rangle. \tag{6}$$

Here $H(\cdot)$ is the Heaviside function taking value 1 if the parameter is greater than 0, and value 0 if the parameter is less than 0. The expectation $\langle \cdot \rangle$ in the above definitions is over all possible distributions of potential nucleation sites. As a consequence of the above definitions, microscopic variables need to be computed for a few different distributions of potential nucleation sites before the macroscopic variables above are obtained.

In this work, we will only use the definitions in Eqs. (4)–(6) above to compare a fully-resolved microscale simulation over the domain of interest (when such simulation is possible) with macroscopic variables introduced as part of the multiscale framework.

Macroscopic temperature – The macroscopic temperature T is defined in this work as the solution of the heat diffusion equation with a latent heat term as follows:

$$\rho c \frac{\partial T}{\partial t} = k \nabla^2 T - \rho L \dot{f}. \tag{7}$$

This governing equation for the macroscopic temperature T is very similar to simple heat diffusion except with a latent heat term $-\rho L \dot{f}$. The first law of thermodynamics, energy conservation, is implied in Eq. (7). This assumption is also utilized in our previous macroscale model using volume-averaging techniques [13]. The function $f(x,t)$ (and thus its time-derivative $\dot{f}(x,t)$) defining the latent heat evolution is as of now an unknown function that needs to be specified as part of the multiscale framework.

Macroscopic concentration – The macroscopic concentration $\overline{C}(x,t)$ is assumed to be constant, i.e. $\overline{C}(x,t) = C_0$, where C_0 is the initial concentration. Since convection is not considered in this work, solute rejected from the solid crystals/dendrites will only cause micro-segregation [14] with variation of concentration in the scale of the solute boundary layer. The size of the solute boundary layer (estimated as D/V, where D is the solute diffusion coefficient and V is the solidification speed) is usually less than the size of the crystals. So in the macroscale, which involves a significant number of crystals, the variation of the concentration is very small. The assumption of $\overline{C}(x,t) = C_0$ is thus reasonable. This is also true for our previous macroscale volume-averaging model in the case of negligible fluid flow [12].

Microstructure features – One is often interested in the features of the solidification microstructure at the end of the solidification process, such as the primary dendrite arm spacing, the secondary dendrite arm spacing or the Heyn's interception measure. Let us denote in an abstract format the finally obtained microstructure features at location x as $\Lambda(x)$. The precise definition of $\Lambda(x)$ used in this work will be discussed later in Section 3.4. In this work, we consider the macroscopic temperature $T(x,t)$, macroscopic concentration $\bar{C}(x,t)$, liquid volume fraction $f(x,t)$, and microstructure features $\Lambda(x)$, as the macroscopic variables in our multiscale framework. An averaging volume is needed to define the microstructure features Λ at a particular point x as demonstrated in Fig. 1. We need to orient the average volume (rectangle) in a way such that it takes account of the growth direction (the rectangle's top and bottom sides are along the growth direction). In this work, we will predict $\Lambda(x)$ utilizing the data generated from directional growth problems (on the microscale) that are defined from the features (growth velocity and temperature gradient) at point x. $\Lambda(x)$ will be used to provide a measure of the type and size of the obtained microstructure at location x.

The final microstructure features are assumed to depend only on the cooling history and temperature gradient history as follows:

$$\Lambda(x) = \Lambda(\mathbf{R}^x, \mathbf{G}^x), \tag{8}$$

where the cooling history $\mathbf{R}^x$ and temperature gradient history $\mathbf{G}^x$ are two functions of time t. They are defined as $\mathbf{R}^x \equiv \frac{\partial T(x,t)}{\partial t}$ and $\mathbf{G}^x \equiv \|\nabla T(x,t)\|$, respectively. In this assumption (Eq. (8)), we use superscript x

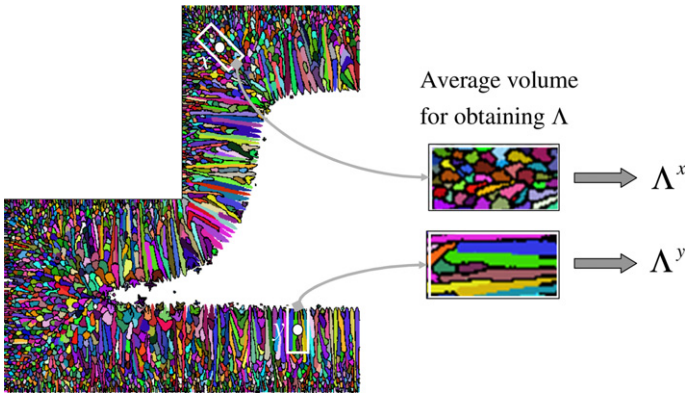


Fig. 1. Schematic of average volume to obtain microstructure features at two points. The features Λ are defined by statistical averaging of the results of appropriately defined microscale directional solidification problems.

to emphasize that the cooling rate and temperature gradient depend on the location x. The bold form ($\mathbf{R}$ and $\mathbf{G}$ instead of R and G) is used to denote that the cooling rate and temperature gradient vary with time. The final microstructure features after solidification, $\Lambda(x)$, are often of great interest as they are related with the mechanical properties of cast products. There are lots of analytical and numerical studies in the literature [4,14–16] investigating the effects of cooling rate and thermal gradient on the microstructure pattern for directional solidification. All these studies [4,14–16] demonstrate that the cooling rate and temperature gradient determine the microstructural features for a given material even though there is no universal agreement on the functional form of this dependence. However, they all point to Eq. (8) as a reasonable assumption.

One significant simplification made in these available studies is that of constant cooling rate and constant temperature gradient, so the commonly used approximations are of the form $\Lambda(x) = \Lambda(R^x, G^x)$. In this work, we are relaxing this assumption to $\Lambda(x) = \Lambda(\mathbf{R}^x, \mathbf{G}^x)$. Although our assumption states that the whole history of cooling rate and temperature gradient determines the microstructure, intuitively we know that only a part of this history is important. At time much earlier or much later than a particular period, cooling rate and temperature gradient have only minor effects on the microstructure. Therefore, it should be possible to replace $\mathbf{R}^x$, $\mathbf{G}^x$ in the above assumption with much simpler parameters. The proposed multiscale approach will provide a reasonable replacement.

Liquid volume fraction f – We assume that the macroscopic liquid volume fraction only depends on the temperature and microstructure features as follows:

$$f(x,t) = f(T(x,t), \Lambda(x)). \tag{9}$$

In our previous macroscale volume-averaging model [13], the volume fraction was taken only as a function of temperature for a given alloy, i.e. $f(x,t) = f(T(x,t))$. However, for a given alloy, we may obtain different microstructure patterns, e.g. planar, cellular, dendritic, or mixed. To more accurately model the liquid volume fraction variation, we consider its dependence on microstructure features, which leads to the above assumption.

As a summary, our macroscale model defining T, $\overline{C}$, $f(x,t)$ and $\Lambda(x,t)$ is given as follows:

$$\rho c \frac{\partial T}{\partial t} = k\nabla^2 T - \rho L \dot{f}, \tag{10}$$

$$f = f(T, \mathbf{R}, \mathbf{G}), \tag{11}$$

$$\Lambda = \Lambda(\mathbf{R}, \mathbf{G}), \tag{12}$$

$$\overline{C}(x,t) = C_0. \tag{13}$$

The purpose of the above simplified model is not only to capture the effects of microstructure evolution on the macroscopic variables, but also to provide means to extract the microstructural features for a given values of macroscopic variables.

2.3. Unknown functions

Notice that even though the macroscale model contains only macroscopic variables, one cannot solve it directly due to the existence of the two unknown functions, $f(T, \mathbf{R}, \mathbf{G})$ and $\Lambda(\mathbf{R}, \mathbf{G})$. Information is needed from the microscale in order to determine these functions.

One approach in computing these functions is using analytical expressions. For example, the primary spacing can be approximated using formulas for cellular or dendritic directional growth with constant cooling rate and thermal gradient [14]. Simplifications are also taken for deriving analytical expressions for $f(T, \mathbf{R}, \mathbf{G})$ (e.g. Lever or Scheil rules given in the form of $f(T)$ [17]). In these expressions, very little microscale information is incorporated in computing the volume fraction. Therefore, the estimated volume fraction may deviate significantly from the actual values for a given temperature and microstructural features.

Another approach is to use numerical analysis that is capable of removing several of the simplifications required in analytical calculations. In fact, numerical methods and of course experimentation are the only ways to quantitatively evaluate the unknown functions f and Λ without introducing too many simplifications. A number of efforts in the direction of numerical study have been presented. For example, in our previous work [4], we proposed a method based on the level set method with markers, which can be used to simulate interaction between multiple crystals. In this earlier work, demonstration of convergence of the results with adaptive mesh refinement was shown. In this work, we will use this previously developed microscale solver to generate information for evaluating the two unknown functions.

3. The database approach

It is straightforward to think of running the microscale model for some particular problems and post-process the obtained results to derive some estimation of the functions f and Λ. The particular problems, which usually have small domains (finer scale), must be of some relevance to our macroscopic problem of interest, which usually has a much bigger domain (coarser scale). We call the selected particular problems with smaller domains as the 'sample problems'. This idea is referred to here as the database approach. One can define different sample problems with different domains, different initial conditions or different boundary conditions. Some of them may be relevant to our problem of interest, some may not. Selecting good sample problems would be the key to success of the database approach for multiscale solidification modeling.

3.1. Domain of the sample problem

In addition to the domain of the problem of interest, we herein define the domain of the sample problem where dendritic solidification results will be generated for use in the database multiscale approach. The domain of the sample problem in general depends on the physics that we are interested to resolve on the microscale. In particular, it should be large enough to contain a sufficient number of crystals for averaging and small enough to minimize the computational cost of simulating in this sample domain dendritic solidification (microscale model). In our numerical examples, the domain of the sample problems is selected as a rectangle that at the end of solidification contains around 10–100 crystals. For the examples considered later in this paper, this is approximately $\frac{2}{100}$ to $\frac{8.64}{100{,}000}$ of the area of the problem of interest (i.e. of the problem where we are interested to perform multiscale analysis).

3.2. Model M and features F_M

In Section 2, we discussed two models (microscale and macroscale models). The microscale model (also referred to here as the fully-resolved model) can be applied both to the sample problem domain as well as to the domain of the problem of interest. However, the simulation corresponding to the last problem may be impractical to perform for most realistic systems due to the intensity of its computational requirements.

Here, we introduce a third model (that for simplicity of presentation we will refer to as model M) to solve the solidification problem.

This model 'treats the material as a pure material without modeling of nucleation'. Melting temperature of the 'pure' material is modeled to be $\theta_m + m_l C_0$. The governing equations for model M are as follows:

$$\rho c \frac{\partial \theta_M}{\partial t} = k \nabla^2 \theta_M, \tag{14}$$

$$V_M = \frac{q_s - q_l}{\rho L}, \tag{15}$$

$$\frac{\partial \phi_M}{\partial t} + V_M \|\nabla \phi_M\| = 0. \tag{16}$$

Here, θ_M, V_M and ϕ_M are the temperature, interface velocity and signed distance from the interface. The subscript M is used to emphasize that the solution is obtained from model M (not from the microscale model). q_s and q_l are the heat fluxes at the solid and liquid sides of the interface, respectively.

Model M can be applied both to the problem of interest as well as to the sample problem domain. Fig. 2 shows for example its application to the domain of the problem of interest. Because the solidification material (an alloy) is treated as a pure material without modeling of nucleation in model M, we expect that the solution of model M (θ_M and ϕ_M) will not capture the important physics during the solidification process as well as it will be captured by the previously discussed microscale model or macroscale model.

The solid–liquid interface solved with model M is stable since pure material is assumed without nucleation. We define the solution features of model M at x, $F_M(x)$, to be the solidification speed, $V_M(x)$, and thermal gradient in the liquid phase, $G_M^l(x)$, when the solid–liquid interface passes through location x, i.e. $F_M(x) \equiv \left(V_M(x), G_M^l(x)\right)$ as demonstrated in Fig. 2. For any given set of solution features $\left(V_M, G_M^l\right)$, we can find a unidirectional sample problem such that when it is solved with model M, the obtained solution features at any location of the problem domain are the same as the given solution features. This is discussed next.

3.3. Model M applied to the sample problem domain for modeling directional solidification with constant features F_M

Model M is introduced here as a simplified model capable of identifying sample problems relevant to our problem of interest. Let F_M represent the features at point x of the problem of interest obtained with the application of model M. We can now identify a sample problem that when solved with model M corresponds to directional solidification with constant features F_M. Indeed, it is not hard to verify that

$$\theta_M = \theta_m + m_l C_0 + \begin{cases} \frac{\alpha G_M^s}{V_M}\left\{1 - \exp\left[\frac{-V_M(x - V_M t)}{\alpha}\right]\right\}, & \text{when } x < V_M t \\ \frac{\alpha G_M^l}{V_M}\left\{1 - \exp\left[\frac{-V_M(x - V_M t)}{\alpha}\right]\right\}, & \text{when } x \geqslant V_M t \end{cases}, \tag{17}$$

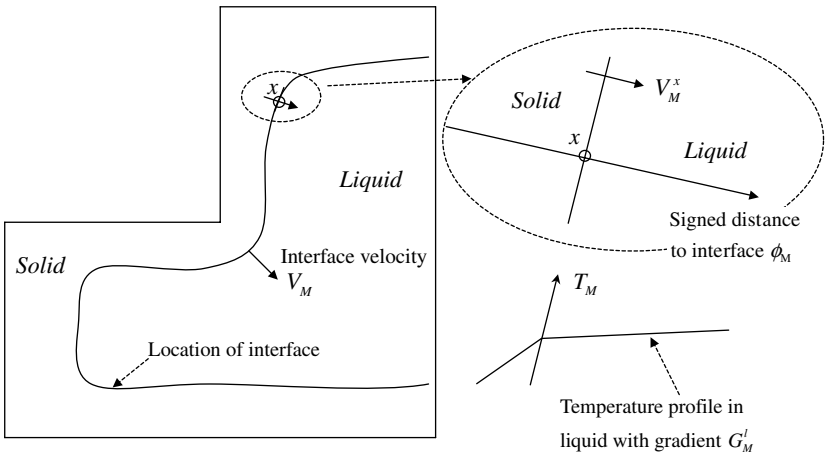


Fig. 2. Schematic of solution features of model M ($V_M(x)$ and $G_M^l(x)$) as applied to the domain of the problem of interest.

is an analytical solution that satisfies model M and gives the exact features (V_M, G_M^l) on the whole domain. Here $\alpha \equiv \frac{k}{\rho c}$ and $G_M^s \equiv G_M^l + \rho L V_M/k$. The above solution satisfies $\rho c \frac{\partial \theta_M}{\partial t} = k\nabla^2\theta_M$. The interface position at time t is at $x = V_M t$, since the temperature θ_M at location $x = V_M t$ equals to $\theta_m + m_l C_0$, which is the melting temperature of the 'pure' material. The temperature gradient at the left side of the interface is G_M^s, whereas the temperature gradient at the right side of the interface is G_M^l.

To define a sample problem giving solution features (V_M, G_M^l) everywhere on its domain, we only need to select a domain and apply boundary/initial conditions according to the above analytical solution. In our numerical examples, the following is used to define a sample problem: (1) Its domain is as discussed in Section 3.1. (2) The initial temperature is given by Eq. (17) with $t = 0$. (3) Adiabatic conditions are considered at the top and bottom sides, whereas the heat fluxes at the left and right side are taken by differentiation of the analytical solution in Eq. (17):

$$q_{\text{left}} = kG_M^s \left\{ \exp\left[\frac{-V_M(0 - V_M t)}{\alpha}\right] \right\}, \tag{18}$$

$$q_{\text{right}} = -kG_M^l \left\{ \exp\left[\frac{-V_M(L - V_M t)}{\alpha}\right] \right\}, \tag{19}$$

where L is the length of the sample domain in the growth direction.

3.4. Microscopic (fully-resolved) model applied to the sample problem domain for modeling directional solidification with constant features F_M

The solution of the fully-resolved microscale model in the sample domain is not unidirectional since symmetry in the vertical direction is not preserved due to the unstable solid–liquid interface (as a result of solute rejection) and the random nature of nucleation. However, application of the simple model M with the same initial and boundary conditions results in a unidirectional solution which further gives constant features F_M.

Our sample problems with the microscale model represent directional solidification from the left to the right. To avoid the effects of initial and final transient stages on the evaluation of f and Λ, we perform averaging of features only on a part of the computational domain as demonstrated in Fig. 3.

At the end of solidification, we count the number of intersections of the horizontal lines with crystal boundaries, N_x, within the domain for averaging as demonstrated in the right plot of Fig. 3. Similarly, we count the number of intersections of the vertical lines with crystal boundaries, N_y. In general, N_x is much less than N_y if the solidification microstructure pattern is columnar, while for equiaxed type of microstructure, N_x is only slightly less than N_y. If the microstructure is fine (coarse), the number of intersections N_y is large (small). Since the microstructure type (columnar or equiaxed) and microstructure size are of great interest, we define the microstructure features as $\Lambda \equiv \left(\frac{N_x}{N_y}, \frac{1}{N_y}\right)$. If $\frac{N_x}{N_y} \leqslant 0.7$, we assume that the microstructure type is columnar. If $\frac{N_x}{N_y} > 0.7$, then we assume the microstructure type to be equiaxed. Here, $\frac{1}{N_y}$ is a measure of the microstructure size, thus a larger $\frac{1}{N_y}$ corresponds to a larger microstructure size.

Let us now concentrate in the computation of the liquid volume fraction $f(T, F_M)$ at a given temperature T for solidification with features F_M. Recall that each sample problem is designed such that it leads to constant features F_M when solved with model M. $f(T, F_M)$ can be interpreted here as the probability of being liquid 'in the average' at a given temperature T within the sample problem defined by features F_M. Using the sample problem results, we estimate $f(T, F_M)$ as the ratio of the number of grid points in our sample problem that remain liquid at T to the total number of grid points. For each grid point x in the sample problem grid, 'x

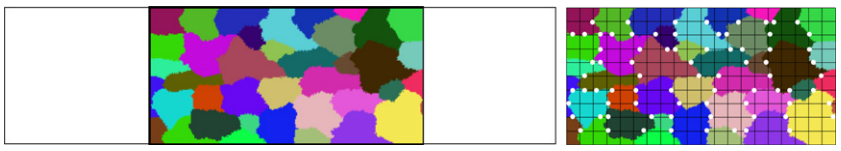


Fig. 3. (Left) Domain for computation (outer rectangle) and domain for performing averaging (inner rectangle). (Right) Schematic of the process for obtaining microstructure features.

remains liquid at temperature T' is equivalent to 'x solidifies at a temperature lower than T', that is '$\theta_s(x) < T$', with θ_s the temperature at x at the time t_s that it solidifies. Therefore, $f(T, F_M)$ can be evaluated using the field of θ_s as follows:

$$f(T, F_M) \approx \frac{\#\text{ of nodes with } \theta_s < T}{\text{total } \#\text{ of nodes}}, \quad \text{sample problem defined with } F_M. \tag{20}$$

Notice that the evaluation of $f(T, F_M)$ in Eq. (20) is performed for each microscale sample problem only after the solidification is complete in the whole domain. The above formulation (Eq. 20) can be used only when θ_s (temperature at solidification) is obtained for all nodes. Before all nodes are solidified, $\theta_s(x)$ (temperature at solidification) remains unknown for the liquid nodes ($\phi(x) > 0$). If the sample problem domain is increased, then more number of points are used for averaging, which means better accuracy of estimation $f(T, F_M)$. However, the computation cost of solving the microscale model for the sample problem will also increase.

3.5. Sample problems relevant to the problem of interest

Given a sample problem, we need a methodology to infer its relevance to our problem of interest. One idea is to use the microscale model for making this decision. Suppose that at location x for the problem of interest, we obtain solution features (i.e. liquid volume fraction $f^x$and microstructure features Λ^x), F^x_{full} using the fully-resolved model (this is of course a difficult or computationally impossible problem). Let us also assume that for the sample problem, we obtain (f, Λ) using the microscale model as discussed in Section 3.4. If $F^x_{\text{full}} \approx F_{\text{full}}$, i.e. $(f^x, \Lambda^x) \approx (f, \Lambda)$, then we could say that the selected sample problem is relevant to our problem of interest at location x. Instead of using the fully-resolved model, we can use the much simpler model M to figure out the relevance of the sample problem to the problem of interest. We could run the simple model M for the problem of interest and the sample problem to obtain solution features F^x_M and F_M. Comparing these solution features, we could have a sense whether the selected sample problem is relevant or not to our problem of interest at location x.

Before exploring further the above idea, let us first introduce a number of definitions:

(1) Given predefined solution features F, if a sample problem $\mathbf{P}$ (with appropriate domain $\Omega^{\mathbf{P}}$, initial and boundary conditions) gives constant solution features over $\Omega^{\mathbf{P}}$ (denote the constants as $F^{\mathbf{P}}_M$) using model M (i.e. $\forall y \in \Omega^{\mathbf{P}}$, $F^y_M = F^{\mathbf{P}}_M$), and if the constant solution features equal the predefined solution features (i.e. $F^{\mathbf{P}}_M = F$), then we say that the sample problem $\mathbf{P}$ is *relevant* to solution features F with model M.
(2) Suppose that for the problem of interest at location x, the solution features of model M are F^x_M. If a sample problem $\mathbf{P}$ is *relevant* to solution features F^x_M with model M, then we say that the sample problem $\mathbf{P}$ is *relevant* to location x for the problem of interest with model M.
(3) If the sample problem $\mathbf{P}$ is *relevant* to location x for the problem of interest with both model M (i.e. $F^{\mathbf{P}}_M = F^x_M$) and the fully-resolved model (i.e. $F^{\mathbf{P}}_{\text{full}} = F^x_{\text{full}}$), then we say that the sample problem $\mathbf{P}$ is *consistent* with model M at location x for the problem of interest.

Remark 1. Note that in the definition of 'relevance' of the sample problem $\mathbf{P}$ to location x of the problem of interest, we require constant solution features of model M for the sample problem. Constant solution features over the sample problem domain gives us convenience in averaging the fully-resolved model solution, since all grid points within the sample problem domain undergo the same condition (having the same solution features) in the sense of model M. It is reasonable to use all grid points for performing the average to smooth out variations in the microscopic solution (obtained using the fully-resolved model) as was detailed in Section 3.4.

3.6. Multiscale framework

Even though it is difficult to find a sample problem that is *relevant* to location x with the fully-resolved model, we can find a sample problem, $\mathbf{P}$, *relevant* to location x with the simpler model M (i.e. $F^{\mathbf{P}}_M \approx F^x_M$) as discussed earlier. By introducing the assumption that $\mathbf{P}$ is *consistent* with model M (i.e. $\mathbf{P}$ also *relevant* to location x with the fully-resolved model), we can run the fully-resolved model to obtain $(f^{\mathbf{P}}, \Lambda^{\mathbf{P}})$, which

is also (f^x, Λ^x) according to the *consistency* assumption discussed earlier. In other words, once solution features F_M are given, we can find a *relevant* sample problem and use the *consistent* assumption to obtain f and Λ. Therefore, f and Λ can be understood as functions of F_M, i.e. $f = f(T, F_M)$ and $\Lambda = \Lambda(F_M)$. With this idea, the macroscale model is changed to a multiscale model as follows (see Fig. 4):

$$\rho c \frac{\partial T}{\partial t} = k\nabla^2 T - \rho L \dot{f}, \tag{21}$$

$$f = f(T, F_M), \tag{22}$$

$$\Lambda = \Lambda(F_M). \tag{23}$$

Comparing the above model with the original macroscale model, we have now used F_M to replace $(\mathbf{R}^x, \mathbf{G}^x)$. The procedure of using the database approach would then involve three important steps:

(1) Solve model M for the problem of interest to obtain the solution (θ_M and ϕ_M) and extract the solution features $F_M = (V_M, G_M^l)$.
(2) Define and solve *relevant* sample problems using the microscale model to evaluate $f(T, F_M)$ and $\Lambda(F_M)$.
(3) Solve $\rho c \frac{\partial T}{\partial t} = k\nabla^2 T - \rho L \dot{f}$ with the obtained information of $f = f(T, F_M)$.

These three steps are demonstrated in Fig. 4. As shown in the box with dashed line (lower right of the figure), model M and the fully-resolved model are used for the same sample problem. The sample problem strategy discussed in Section 3.3 guarantees that the solution features of model M are equal to the given solution features F_M^x. We have an analytical solution with model M for the sample problem, so model M is not numerically solved for the sample problem (as is the case for the problem of interest in step 1 of the algorithm above). Only the fully-resolved model is numerically solved for the sample problem to generate data for estimation of $f(T, F_M)$ and $\Lambda(F_M)$ so that the macroscale model incorporates information from the microscale.

Once we have obtained the values of f at the various temperature levels for each node (according to F_M at the corresponding node) after step 2, we need to keep track of these values at each node in the memory. Notice that the liquid volume fraction f appears in the temperature governing equation of the macroscale model, whereas the microstructure features Λ do not appear in the temperature governing equation. Obtaining microstructure features Λ can be understood as a post-processing process. Since their values do not affect other

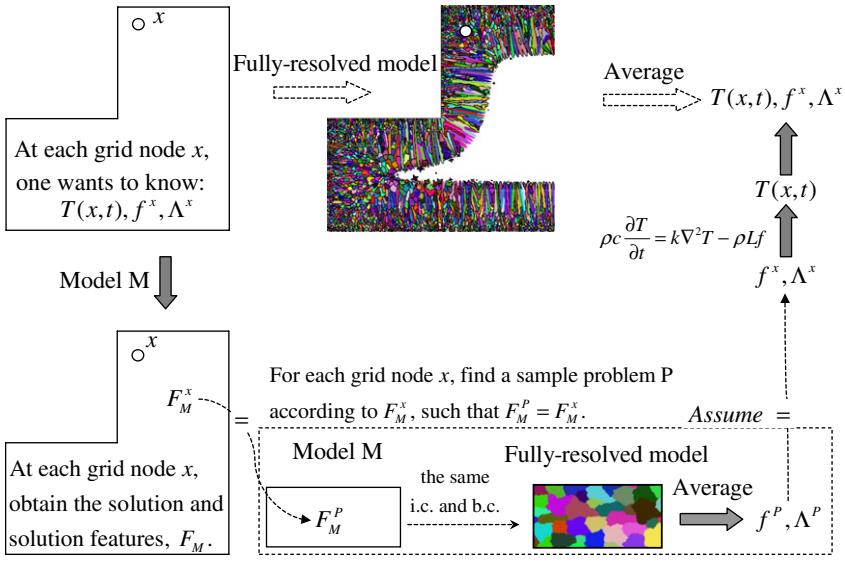


Fig. 4. Schematic of the multiscale framework (steps indicated with the dark arrows).

computations, the microstructure features Λ do not have to be tracked in the memory during the computation. All the effect of the microscale on the macroscale is in the liquid volume fraction f.

The two horizontal arrows on the upper part of Fig. 4 demonstrate the procedure of using the fully-resolved model to obtain macroscopic variables. This procedure is usually very time consuming and may be impossible to perform due to limitations in computational resources. For this reason, we only solve the fully-resolved model for the problem of interest in the first numerical example in Section 5 for validating the database approach. The term "average" above the second arrow on the upper part of Fig. 4 is used to represent the process of obtaining macroscopic variables including temperature, liquid volume fraction and microstructure features as discussed in Section 2.2. However, due to the high computational cost, in the numerical example with validation, only the macroscopic temperature T is computed following its definition (i.e. by averaging microscopic temperature obtained from different sampling of potential nucleation sites). For validation of the volume fraction f at a given time t, we will compare the solid–liquid interface position at time t (obtained by the fully-resolved model) with the volume fraction contour lines (obtained by the database approach). Also for validation of the microstructure features Λ at a given location x, we will compare the microstructure at location x obtained by the fully-resolved model with the microstructure obtained by the database approach (i.e. by searching in the database to obtain the microstructure with solution features nearest to $F_{\mathrm{M}}(x)$).

The first step in the above algorithm can be understood as a 'prediction step', since the temperature is predicted as θ_{M} using model M. The second step gathers information on volume fraction and microstructure features from the microscale computations in the sample problem. The third step can be understood as a 'correction step', since the temperature is corrected with updated information on volume fraction.

The boundary and initial thermal conditions for the algorithm discussed up to now are the same for the problem of interest when it is solved with model M, the fully-resolved model (e.g. in our first example in Section 5.1.4) or the multiscale model. However, to increase the accuracy of the methodology while maintaining the simplicity of model M, in our multiscale implementation, model M is applied only for a part of the domain around the interface. The details of the overall algorithm are discussed next.

3.7. Overall multiscale algorithm

The procedure discussed in the previous section assumes that the selected sample problem is *consistent* at any location x of the domain of the problem of interest with model M. This is a strong assumption especially for a problem with large domain. The model M is not capable to very accurately model the evolution of temperature. At the beginning of solidification, the error in temperature (i.e. the difference to temperature computed by the fully-resolved model) is small. So the above procedure can predict the solidification microstructure quite well for the early stages of solidification. However, as the error in temperature obtained using model M (i.e. θ_{M}) becomes larger with time, the above procedure becomes less accurate for predicting the obtained solidification microstructure.

In order to deal with this difficulty, we need to find a way to constrain the error accumulation in model M by utilizing the temperature data obtained (from an earlier iteration) from solving the macroscale model (by solving the heat diffusion equation with a latent heat term $-\rho L\dot{f}$). The iterative method we used is demonstrated in Fig. 5. For the first iteration (iteration 0 in Fig. 5), the procedure is as before. But at later iterations, we apply model M only on the region near the solid–liquid interface (darker region in Fig. 5). At places far away from the solid–liquid interface (lighter region in Fig. 5), we use the temperature from the macroscale model obtained in the previous iteration. In this way, the temperature from the macroscale model, is always applied as boundary condition for the region near the solid–liquid interface (i.e. darker region in Fig. 5) on which model M is solved. Using temperature from the macroscale model as boundary condition constrains the error of model M from accumulating.

In this approach, we need to define the region for applying model M (i.e. darker region in Fig. 5). Since model M gives a signed distance field ϕ_{M}, we have the information of distance to the solid–liquid interface at any location. We can introduce a parameter L_{M} to specify the size of the region for applying model M. For the region within distance L_{M} away from the solid–liquid interface (i.e. $|\phi_{\mathrm{M}}| \leqslant L_{\mathrm{M}}$), we apply model M. Otherwise (i.e. $|\phi_{\mathrm{M}}| > L_{\mathrm{M}}$), we use the temperature data from the macroscale model obtained in the previous iteration.

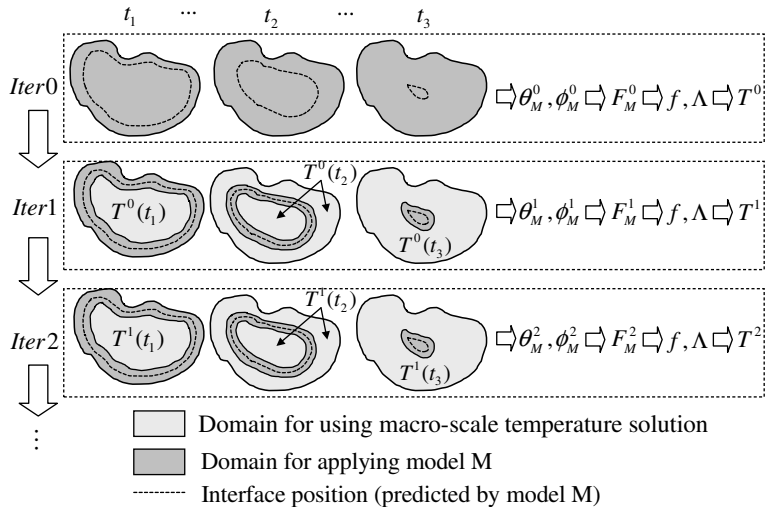


Fig. 5. Schematic of the overall algorithm.

The following is the procedure for implementation of this algorithm:

(1) Set iteration number $i = 0$.
(2) Solve model M to obtain θ_{M}^i, ϕ_{M}^i. If $i > 0$, essential boundary condition $\theta_{\mathrm{M}} = T^{i-1}$ is applied at nodes with $|\phi_{\mathrm{M}}| > L_{\mathrm{M}}$ when solving for the temperature field with model M. Obtain solution features F_{M}^i for every node point.
(3) For solution features F_{M}^i at each node of the grid for the whole domain, find $f(T, F_{\mathrm{M}})$ and $\Lambda(F_{\mathrm{M}})$ by solving the appropriate sample problems.
(4) Obtain solution of temperature field T^i by solving $\rho c \frac{\partial T}{\partial t} = k\nabla^2 T - \rho L \dot{f}$ with the updated liquid volume fraction f.
(5) Set $i = i + 1$ and return to step 2 (three iterations were typically sufficient in the numerical examples considered).

This algorithm uses solution of the macroscale model to improve our prediction of f, Λ and correspondingly of T. Comparing with the algorithm without iterations, the additional cost we pay is the computation time for the required iterations and the storage for T^{i-1}. To perform iteration i, we need to track the solution of T^{i-1} on the whole domain (for each node) and at all times (for each time step). Since we are using coarse grid spacing and large time steps for T (macroscale computation), this additional storage is quite affordable.

4. Numerical implementation

4.1. Reducing the number of the needed sample problems using interpolation in the feature space

Solving the sample problem using the fully-resolved model is computational intensive due to the required fine mesh. So in order to speed up the computation, we want to minimize the number of times for solving sample problems. The idea of interpolation is shown in Fig. 6. We only need to run a sample problem on each node of the mesh (triangular mesh is shown in the schematic, but quadrilateral mesh is also used in the examples) generated in the feature space ($G_{\mathrm{M}}^{\mathrm{l}}$ vs. V_{M} instead of x vs. y). For evaluation of f and Λ at an arbitrary feature, we will do interpolation.

As shown in Fig. 6, we first obtain F_{M}^x on each node (macroscale grid) using model M. The relation between temperature and liquid volume fraction at features F_{M}^x (i.e. $f(T, F_{\mathrm{M}}^x)$) in general is not available in the database. However, we can find a few such relations at features close to F_{M}^x (as demonstrated in the top left

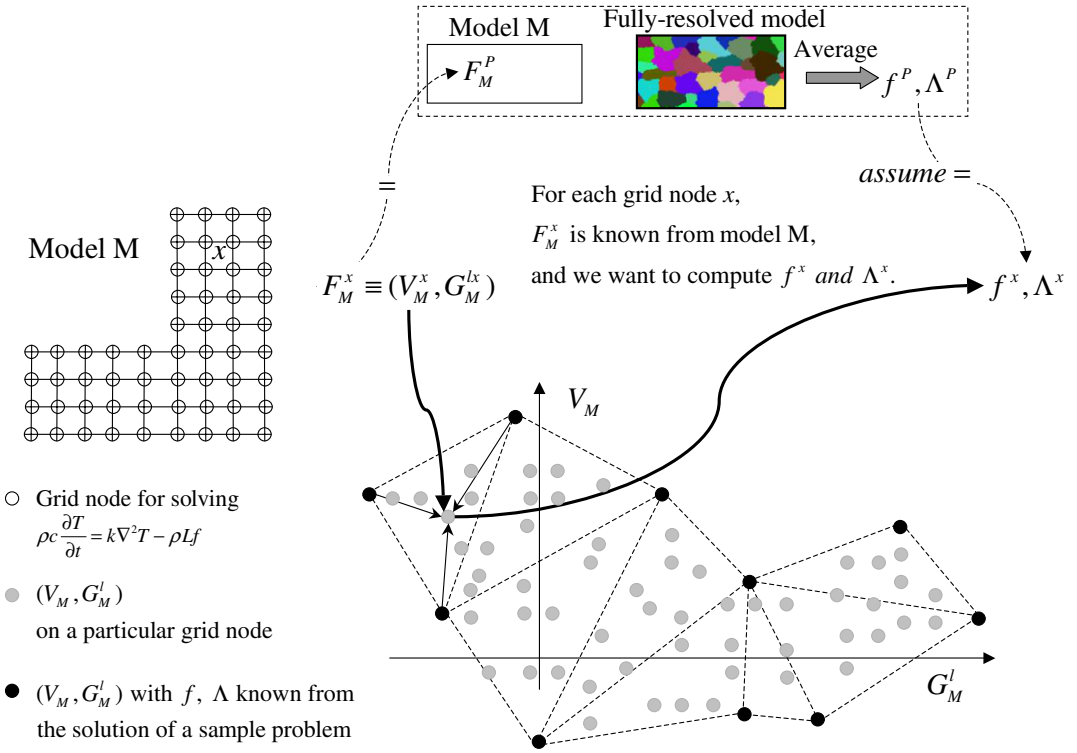


Fig. 6. Schematic of using interpolation to reduce the number of the required sample problem solutions. The top shows how we can use the time consuming fully-resolved model to evaluate f and Λ corresponding to an arbitrary feature. The bottom shows all obtained solution features (gray dots), a mesh generated in the feature space and how interpolation is used to obtain f and Λ based on results of sample problems corresponding to nodes of the mesh (black dots).

triangle) in the database. They will be used to interpolate $f\left(T, F_{\mathrm{M}}^x\right)$. We need to do this interpolation and store the interpolated liquid volume fraction f for all nodes. After this, solving the macroscale model requires the solution of the diffusion equation with a latent heat term determined by the interpolated f.

4.2. Storing sample problem results

In the database approach, there is a huge amount of data to be processed. For example, in our first numerical example in Section 5, if a uniform mesh is used, the required mesh size is 1024×256 with around 0.25 million nodes. If a CFL coefficient 0.1 is used, then we will need about $1024/0.1 \approx 10{,}000$ time steps. On each node and at each time step, we have variables such as C, θ, I, and ϕ. So we will have around $0.25M \times 10{,}000 \times 4 = 10G$ float numbers as results of just one sample problem. For efficiency in storing data and extracting useful information from data, we want to store the minimum amount of data. Another difficulty in storing sample problem results is related with the adaptive meshing technique we used for solving sample problems on the microscale. Computation is performed on an adaptive mesh, while statistics must be extracted using a uniform mesh.

To deal with these difficulties, during computation of the sample problem, at each time step, we store the location x, current time (solidification time) t_{s}, solidification temperature (temperature at x at the time t_{s}) θ_{s} and orientation angle I for each node only at the time it is solidified (i.e. $\phi^{n-1} > 0$ and $\phi^n \leqslant 0$). Since the smallest grid spacing Δx is used for the region within some distance ($3\Delta x$ in our computations) away from the solid/liquid interface in both solid and liquid sides of the interface, each node i in the equivalent uniform mesh (i.e. structured mesh with grid spacing Δx) will have data (including I, θ, C) directly obtained from finite element computation when $-3\Delta x \leqslant \phi_i \leqslant 3\Delta x$. If $\phi_i > 3\Delta x$ or $\phi_i < -3\Delta x$, then data (including I, θ, C) may have to be obtained from interpolation, since the node may not exist in the adaptive mesh. During our computation, the

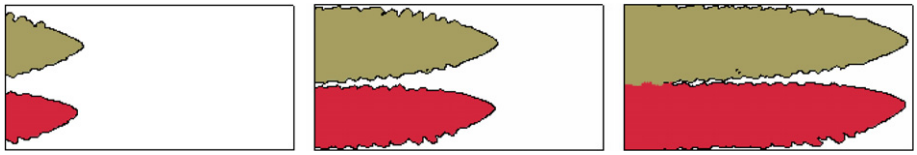


Fig. 7. Contour line of field t_s at value $t_s = 250$, 325 and 400 for one of the sample problems with $V_M = 0.02281344$ and $G_M^l = 0.6708713$ discussed in the numerical examples section. Regions of $t_s \leqslant 250$, $\leqslant 325$ and $\leqslant 400$ are colored with orientation angle to identify two different crystals.

selected CFL coefficient is always less than 1. Therefore, each node in the equivalent uniform mesh exists in the adaptive mesh for the time interval within which it is solidified. In this way, each node in the equivalent uniform mesh will have data (i.e. x, t_s, θ_s and I) recorded exactly once, even if adaptive meshing is used for computation. We keep appending data (x, t_s, θ_s and I) to a file, until all nodes in the adaptive mesh are solidified (i.e. with $\phi < 0$). These recorded data basically contain information of t_s, θ_s and I for all nodes in the equivalent uniform mesh. However, the data is sorted by solidification time (not by location), since each row of data is recorded at the time step during which the corresponding node is solidified (i.e. $\phi^{n-1} > 0$,and $\phi^n \leqslant 0$). So after computation, we read the recorded file and obtain the t_s, θ_s and I fields on the equivalent uniform mesh according to the location information x. Although thousands of time steps are usually involved for a sample problem, the three fields ($t_s(x)$, $\theta_s(x)$ and $I(x)$) provide all necessary information for the multiscale model. The utilization of the fields $\theta_s(x)$ and $I(x)$ to provide an estimation of the liquid volume fraction f and microstructure features F_M was already discussed in Section 3.4.

Finally, note that the field t_s contains information of the interface position at any time level. The contour line of the field t_s gives the position of the solid–liquid interface at the corresponding time level (contour line value) as shown in Fig. 7.

4.3. Other implementation details

Note that the potential nucleation sites generally are not at the finite element nodes. In the implementation of the nucleation algorithm and for storing the location of potential nucleation sites, we assign a list to each element which contains the locations of all potential nucleation sites sampled inside this element. This list may be empty, if there is no potential nucleation site inside this element or very long if there are lots of potential nucleation sites inside it. For determination of whether a potential nucleation site nucleates, we do interpolation (using data on the nodes of the element which contains the potential nucleation site) to find the actual undercooling at the potential nucleation site, and compare it with the required undercooling. One also needs to pay attention to the potential nucleation sites while remeshing. If an element is divided into a few sub-elements, each potential nucleation site inside the parent element is assigned to the sub-element which contains it.

5. Numerical examples

5.1. Verification of the database approach

The following dimensionless material properties are considered in this section: potential nucleation site density $\rho^n = 100$, the required undercooling for nucleation $\Delta\theta_n$ satisfies a normal 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 $\theta_m = 0$, initial concentration $C_0 = 0.1$, liquidus slope $m_l = -10$, partition coefficient $k_p = 0.1$, Gibbs–Thomson relation with anisotropy in kinetic undercooling $\theta_* = \theta_m + m_l C^l - \frac{2}{3}\{1 - \cos[4(\alpha - I)]\}V$, where α is the angle between the positive x axis and the interface normal direction. Notice that the large latent heat $L = 100$ makes the problem nontrivial on the macroscale, whereas the large Lewis number $Le = 100$ makes the problem nontrivial on the microscale.

The computational domain is of size 40×40. However, the left-top quarter is not included making the computational domain irregular. Initially, the whole domain is at temperature 10. The right side is assumed to be

adiabatic. The temperature applied on all other sides is taken as $T_b = 50\exp(-t/10) - 40$ or $T_b = 100\exp(-t/20) - 90$ in two different runs in order to study the effects of the boundary conditions on the obtained solidification microstructure.

5.1.1. Computational results using model M

Using an adaptive mesh with smallest grid spacing 0.1563, we obtain solution features of V_M and G_M^l for temperature boundary condition $T_b = 50\exp(-t/10) - 40$ and $T_b = 100\exp(-t/20) - 90$ as shown in Fig. 8. Using four computation nodes (each with $2 \times 3.8G$ CPUs), the required computational time was around 1 h.

By comparing the results for the two cases, we find that the growth velocity V_M for the case with $T_b = 50\exp(-t/10) - 40$ is much lower than for the case with $T_b = 100\exp(-t/20) - 90$, whereas the temperature gradient G_M^l is only slightly smaller.

5.1.2. Results of sample problems

Fig. 9 plots the computed V_M and G_M^l for all locations in both runs (with temperature boundary condition $T_b = 50\exp(-t/10) - 40$ and $T_b = 100\exp(-t/20) - 90$) in the feature space. As discussed in Section 4.1, we can use interpolation to obtain the liquid volume fraction and the microstructure features for each of the features in the left plot of Fig. 9. Sixty four sample problems are selected to give data for interpolation. The features (V_M and G_M^l) of these 64 sample problems are selected to be uniformly distributed in the log scale as shown on right plot of Fig. 9. Recall that both analytical and experimental results show that microstructure features are in the power form of growth velocity and thermal gradient (i.e. $\sim V^\alpha G^\beta$ with α, β constants).

For each sample problem, we use a domain size of 10×2.5 with smallest grid spacing 0.0098 for adaptive meshing. The selected grid spacing 0.0098 is much smaller than the grid spacing 0.1563 used for running model M. Using a CFL coefficient 0.125, it takes around 8000 time steps to perform the simulation for each sample problem. We use a cluster with 64 nodes (each node with $2 \times 3.8G$ CPUs) to compute these sample problems. Each node is used to perform computation of one sample problem. The computational time is around 5 h. Results of these sample problems in the domain for statistical analysis (as demonstrated in Fig. 3) are shown in Fig. 10. From results of these sample problems, we observe a continuous transition between equiaxed growth and columnar growth with various grain sizes. This study is similar to the previous numerical studies [16,4], in which computations are performed at various thermal gradients and growth velocities. In [16,4], the thermal gradient G and growth velocity V are assumed to be constant, which are only applicable to directional growth. There is no relation established between the performed numerical study at a particular growth speed V and thermal gradient G with other problems. However, in this work, the thermal gradient G_M^l and growth velocity V_M are obtained from the solution of model M for the problem of interest. The sample problems are *relevant* to our problem of interest with model M. We will utilize the information obtained from the micro-scale solution of sample problems to solve the problem of interest by introducing the assumption that the sample problem is also *consistent* to our problem of interest with model M.

Notice that the microstructure pattern (obtained from the fully-resolved model) over the whole sample problem domain is rather uniform for each sample problem. This is a consequence of constant solution features (from model M) over the sample problem domain.

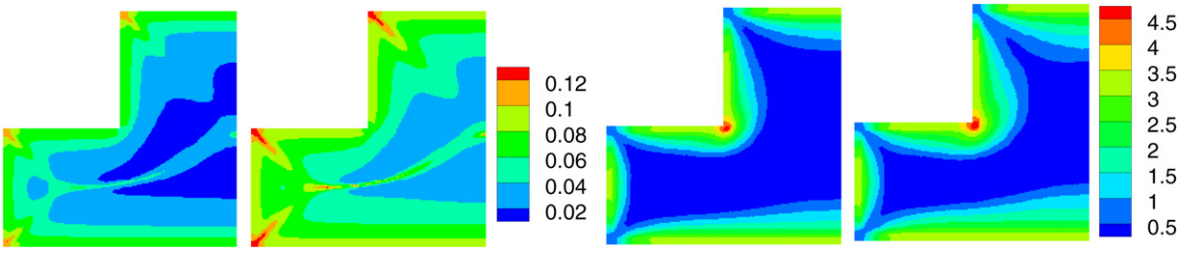


Fig. 8. Contour of V_M (left two plots) and G_M^l (right two plots) for temperature boundary condition $T_b = 50\exp(-t/10) - 40$ (the first and the third plots) and $T_b = 100\exp(-t/20) - 90$ (the second and fourth plots).

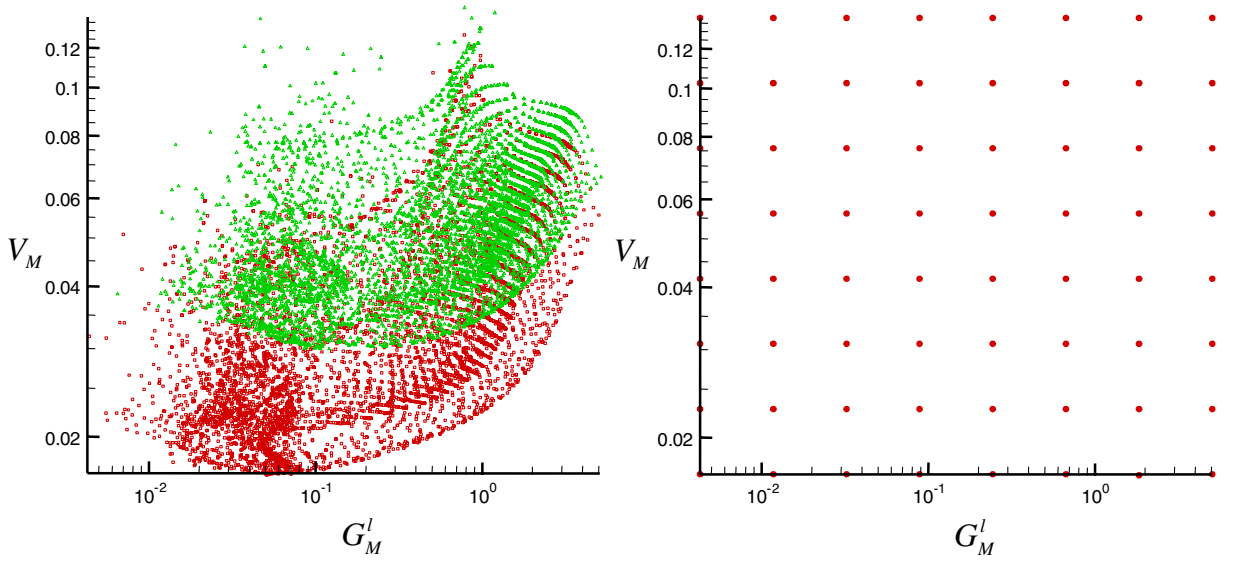


Fig. 9. (Left) Obtained features of model M, (V_{M} and $G_{\mathrm{M}}^{\mathrm{l}}$) with temperature boundary condition $T_{\mathrm{b}} = 50\exp(-t/10) - 40$ (red square symbols) and $T_{\mathrm{b}} = 100\exp(-t/20) - 90$ (green triangle symbols). (Right) (V_{M}, $G_{\mathrm{M}}^{\mathrm{l}}$) of 64 sample problems selected for applying the fully-resolved model. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

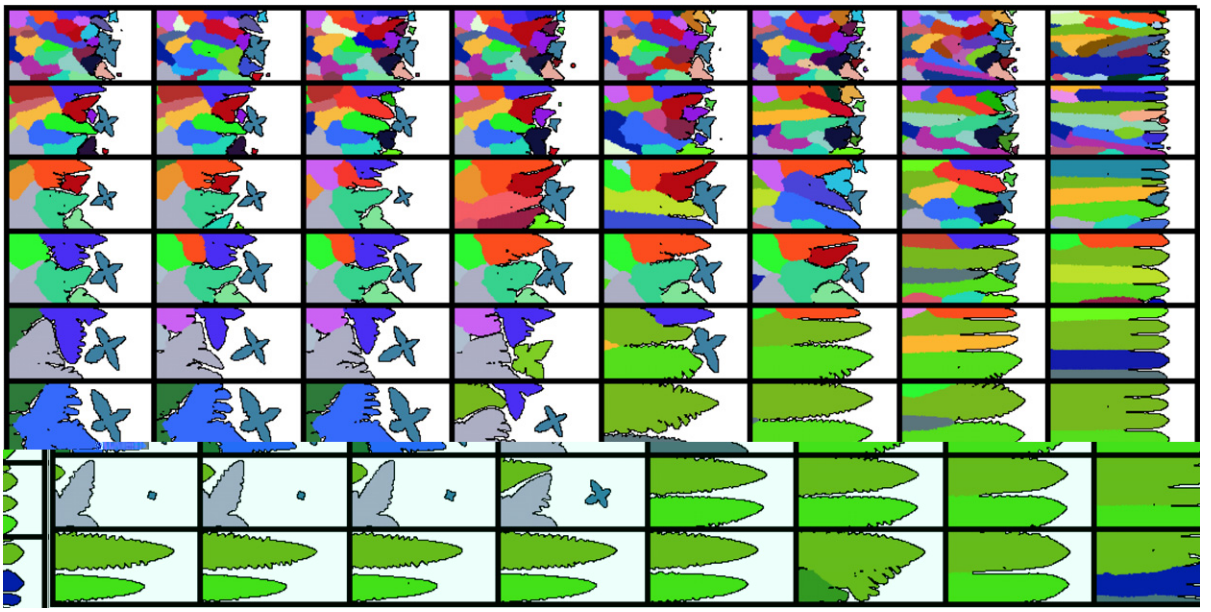


Fig. 10. Obtained microstructure of 64 sample runs. Each rectangle corresponds to a feature (V_{M}, $G_{\mathrm{M}}^{\mathrm{l}}$) shown in the right plot of Fig. 9. The temperature gradient increases from left to right on a given row (constant growth velocity) and the growth velocity increases from bottom to top on a given column (constant temperature gradient).

5.1.3. Results of the database approach

From the results of model M, we obtain $F_{\mathrm{M}}(x)$ on the whole domain. The computed relation between the liquid volume fraction f and temperature T for the 64 sample problems considered is shown in Fig. 11. Using interpolation based on the obtained relation of the liquid volume fraction and temperature for the sample problems with different F_{M} (i.e. $f(T, F_{\mathrm{M}})$), we are able to find the liquid volume fraction $f(x,t)$ for

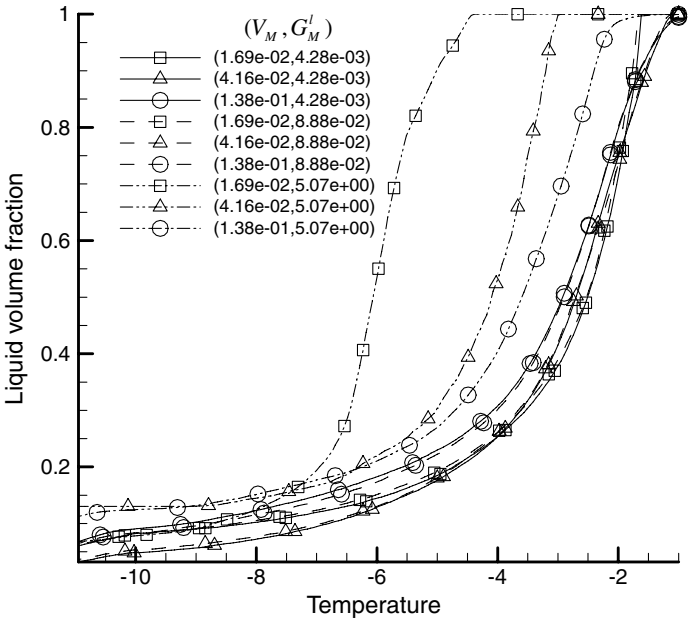


Fig. 11. Obtained relation between the liquid volume fraction and temperature from sample problems with various features F_{M} (64 sample problems are computed but only 9 are plotted here).

solving the macroscale model. The temperature obtained from the macroscale model can then be used to improve the computation of $F_{\mathrm{M}}(x)$. Fig. 12 demonstrates how the predicted microstructure feature $\frac{N_x}{N_y}$ changes with iterations. The field of $\frac{N_x}{N_y}$ at the second iteration is very similar to the field of $\frac{N_x}{N_y}$ at the third iteration demonstrating that the solution has converged after three iterations. In our computation, L_{M} is selected to be 5.

With the results of the 64 sample problems (database) in place and the obtained V_{M}, $G_{\mathrm{M}}^{\mathrm{l}}$ for the problem of interest, we can now perform interpolation to obtain the microstructure size $\frac{1}{N_y}$ and microstructure type $\frac{N_x}{N_y}$ as demonstrated in Figs. 13 and 14. Using a uniform mesh with grid spacing 0.3125 and a fixed time step of 0.1, the computation time with one computation node ($2 \times 3.8G$ CPUs) is around 20 min for the case with boundary condition $T_{\mathrm{b}} = 50\exp(-t/10) - 40$ and around 10 min for the case with boundary condition $T_{\mathrm{b}} = 100\exp(-t/20) - 90$.

In order to give a picture of how the microstructure may appear at a particular location with features $\left(V_{\mathrm{M}}(x), G_{\mathrm{M}}^{\mathrm{l}}(x)\right)$, we can search in the database to find the microstructure with the features closest to $\left(V_{\mathrm{M}}(x), G_{\mathrm{M}}^{\mathrm{l}}(x)\right)$ as demonstrated in the 14 microstructures in Figs. 13 and 14. From the contour of field $\frac{1}{N_y}$

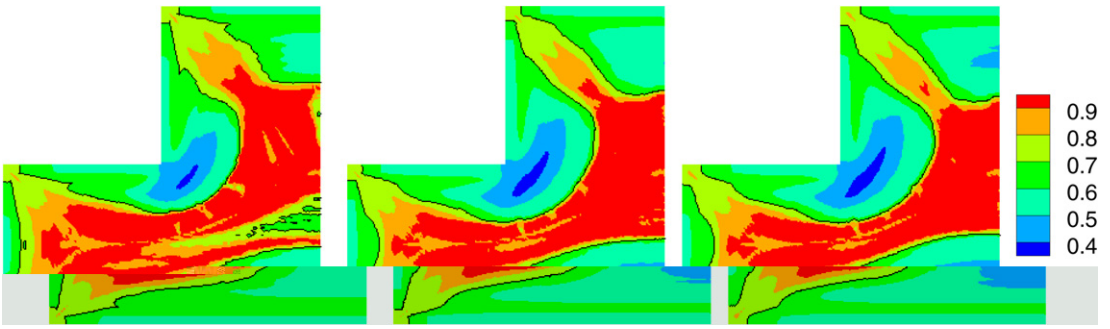


Fig. 12. Predicted field of $\frac{N_x}{N_y}$ at the first, second, and third iteration for the case with $T_{\mathrm{b}} = 100\exp(-t/20) - 90$. Contour line with value 0.7 demonstrates the predicted location of columnar to equiaxed transition (CET).

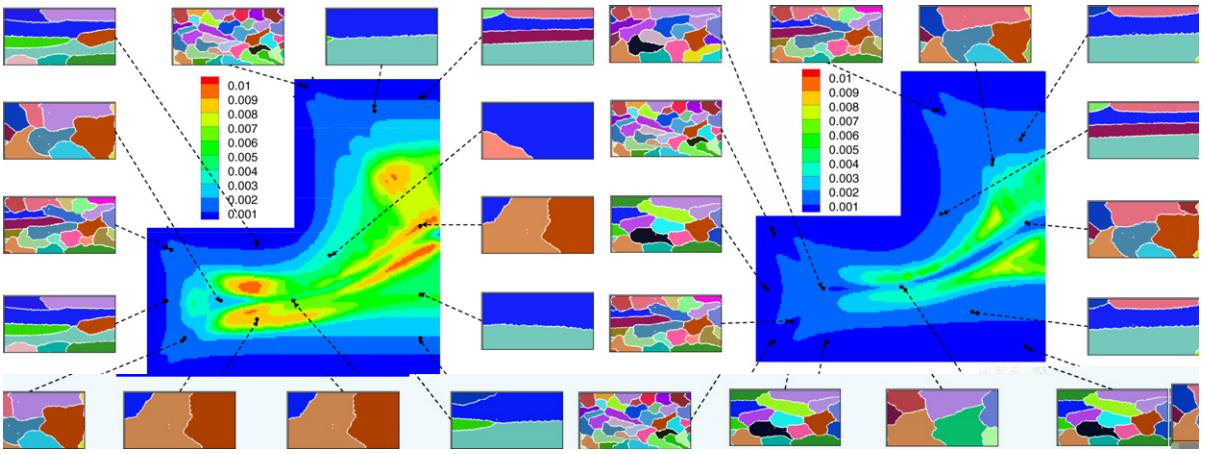


Fig. 13. Field of $\frac{1}{N_y}$ (the 14 microstructures shown correspond to the closest microstructure in the database). (Left) $T_b = 50\exp(-t/10) - 40$. (Right) $T_b = 100\exp(-t/20) - 90$.

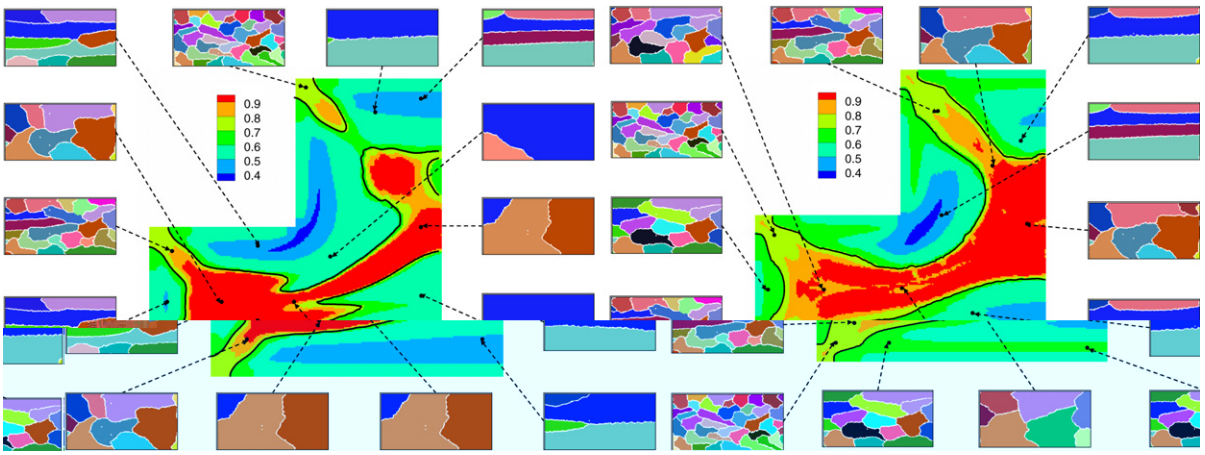


Fig. 14. Field of $\frac{N_x}{N_y}$ (the 14 microstructures correspond to the closest microstructure in the database). (Left) $T_b = 50\exp(-t/10) - 40$. (Right) $T_b = 100\exp(-t/20) - 90$.

or the closest microstructures in Fig. 13, we can predict that large grains will be formed in the center of the casting. The case of the boundary condition $T_b = 50\exp(-t/10) - 40$ leads to larger grains than the case with boundary condition $T_b = 100\exp(-t/20) - 90$.

The contour line in Fig. 14 corresponds to $\frac{N_x}{N_y} = 0.7$. It is the predicted location of columnar to equiaxed transition. In both cases, it is predicted that we will obtain equiaxed microstructures at the corners and also at the center of the domain.

5.1.4. Comparison of microstructure features and liquid volume fraction obtained from the microscale model and the database approach

As a validation of the database approach, we also performed the microscale model on the whole domain with smallest grid spacing 0.0098 for adaptive meshing. The CFL coefficient is selected to be 0.125. Using eight computation nodes (each node with $2 \times 3.8G$ CPUs), the total computation time for each case is about 2 days.

For validation of the database approach, the microstructure features $\Lambda(x)$ are not directly compared with the microscale results. We perform the comparison in the following way: (1) solve the microscale model to obtain microstructure details all over the domain, (2) obtain Λ using the database approach, pick up a few locations in the domain, search the closest microstructure in the database for these locations, and (3) for each

selected location, compare the searched microstructure (from the database) and the microstructure obtained using the microscale model at the corresponding location in the first step. Such a comparison for both cases (different boundary conditions) is demonstrated in Fig. 15. Although this type of comparison is only a qualitative one, it demonstrates that the database approach is capable of predicting microstructure patterns quite well with significantly less computational cost.

At time 130 for the case with $T_b = 50\exp(-t/10) - 40$, the solid–liquid front (using the microscale model) falls almost exactly inside the region with volume fraction $0.05 \leqslant f \leqslant 0.95$ (using the database approach) as shown in Fig. 16. This suggests that the database approach is also accurate for predicting volume fractions.

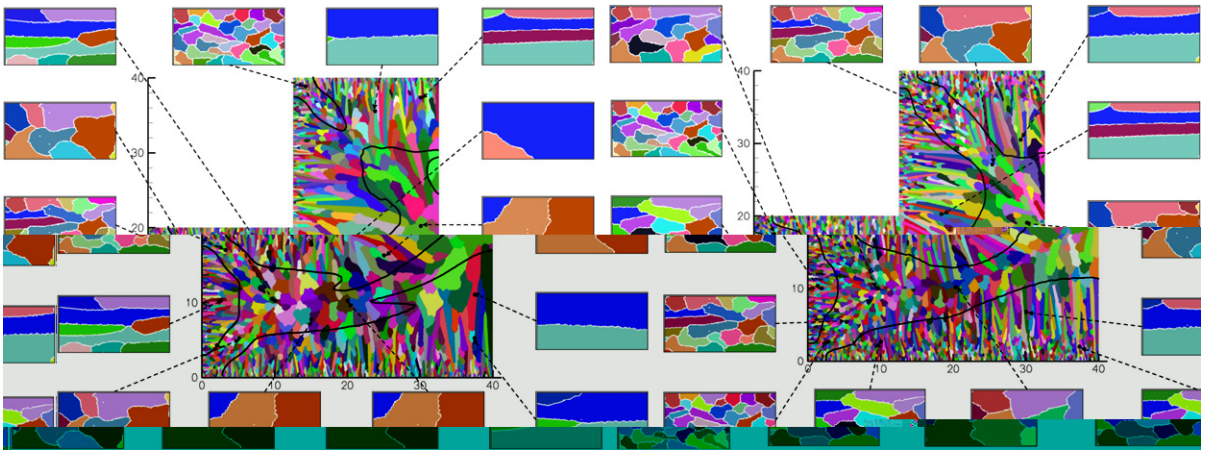


Fig. 15. Comparison of the predicted microstructures using the database approach with the microstructures obtained from solving the problem in the whole domain using the microscale model. (Left) $T_b = 50\exp(-t/10) - 40$. (Right) $T_b = 100\exp(-t/20) - 90$. For each plot (left or right): the picture in the middle is the fully-resolved result; the dark line in the middle picture is the predicted location of CET transition using the database approach $\left(\frac{N_x}{N_y} = 0.7\right)$; the 14 pictures (around the middle picture) are the closest microstructure in the database based on features F_M at selected locations.

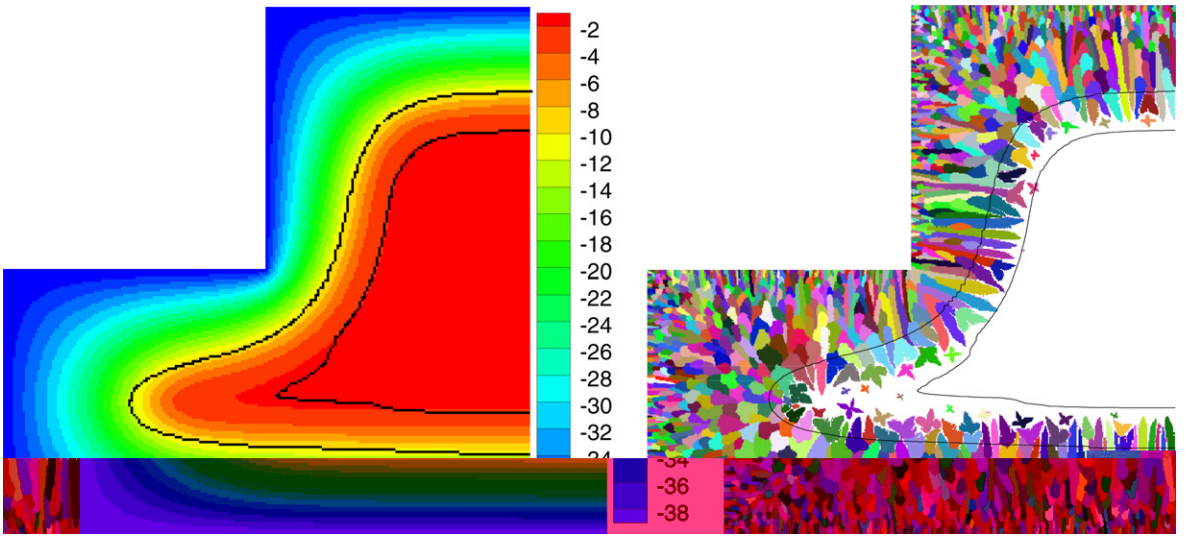


Fig. 16. (Left) Predicted temperature field and liquid volume fraction contours with values 0.95 and 0.05 at time 130 for the case with $T_b = 50\exp(-t/10) - 40$. (Right) Obtained microstructure (using microscale model) and liquid volume fraction contours with value 0.95 and 0.05 (using the database approach) at time 130 for the case with $T_b = 50\exp(-t/10) - 40$.

5.1.5. Comparison of the temperature field obtained from the microscale model and the database approach

As discussed earlier, the macroscopic temperature obtained from the database approach is the expectation of microscopic temperature (i.e. $T(x,t) = \langle\theta(x,t)\rangle$). In order to obtain the expectation of microscopic temperature for comparison with macroscopic temperature, we performed three microscale computations for the case with $T_b = 100\exp(-t/20) - 90$ using different sampling of potential nucleation sites. The microstructure at time 81.6 for these three computations is shown in Fig. 17. Because the potential nucleation sites in these three microscale computations are different, the obtained microstructural details are also different. Correspondingly, the microscopic temperature is also different as shown in Fig. 18.

In the database approach, we have approximated that $f(x,t) = f(T(x,t),\ F_M(x))$ instead of using $f(x,t) = f(T(x,t))$. Two questions may be raised regarding this approximation. First, will $f = f(T,F_M)$ lead to significantly different results (volume fraction and temperature) from $f = f(T)$? Secondly, is the approximation $f = f(T,F_M)$ good enough for obtaining similar results to those from the fully-resolved model. It is very obvious that F_M has a great effect on volume fraction as shown in Fig. 11. For the case with $T_b = 100\exp(-t/20) - 90$, we computed the temperature field at time 81.6 using the fully-resolved model (after averaging among the three microscale computations), using the database approach ($f = f(T,F_M)$) and using the level rule ($f = f(T)$) with results shown in Fig. 19. The temperature field obtained from the database approach is very similar to the temperature field obtained from the microscale model except that small scale variations of temperature exist in the results of the microscale model, which is due to nucleation and growth of equiaxed crystals and the variation in dendrite front position. The temperature field obtained from the Lever rule with approximation $f = f(T)$, however, is quite different from the microscale model result. The results in Figs. 16 and 19 demonstrate that the approximation of $f = f(T,F_M)$ can successfully incorporate the effects of microstructure morphology on volume fraction and temperature.

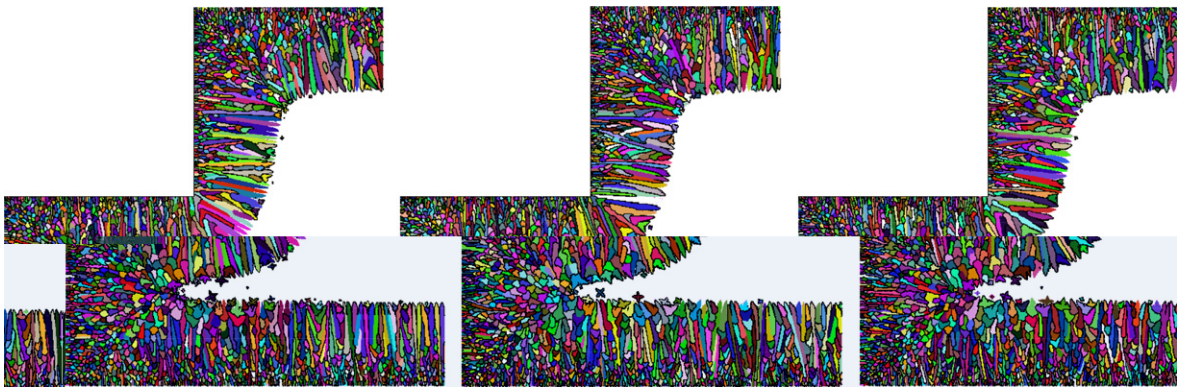


Fig. 17. Microstructure at time 81.6 for the case with $T_b = 100\exp(-t/20) - 90$ using different sampling of potential nucleation sites.

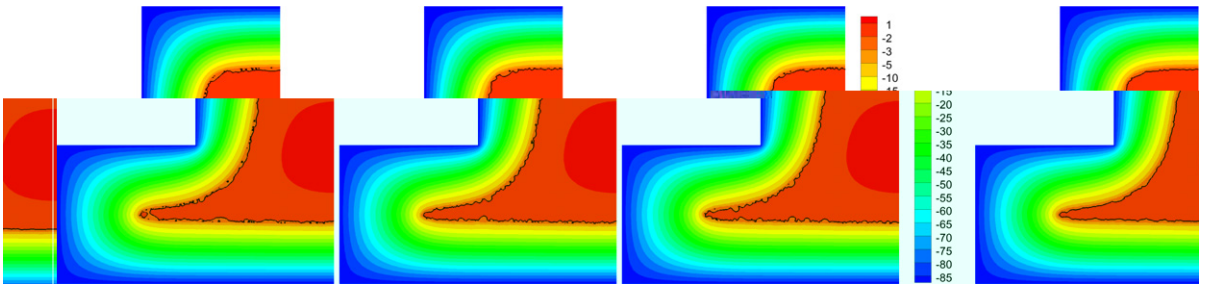


Fig. 18. (Left 3 plots) Microscopic temperature at time 81.6 for the case with $T_b = 100\exp(-t/20) - 90$ using different sampling of potential nucleation sites. (Right plot) Averaged microscopic temperature.

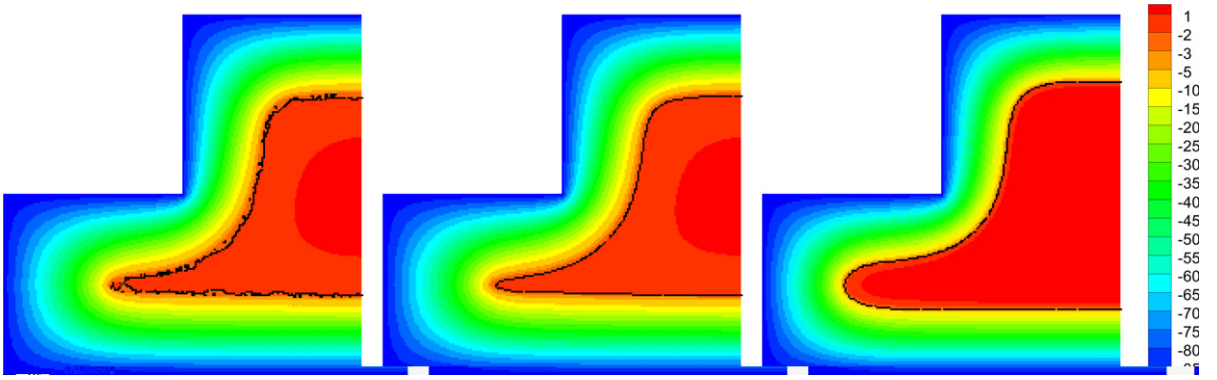


Fig. 19. Temperature field at time 81.6 for the case with $T_b = 100\exp(-t/20) - 90$. Contour line shows the position where the temperature is -2. (Left) Temperature field obtained from the microscale model by averaging among three computation results. (Middle) Predicted temperature field from the database approach. (Right) Predicted temperature using the Lever rule.

5.2. Application to the solidification of an Al–Cu alloy

In the previous numerical example, the material parameters were selected to demonstrate and validate the multiscale modeling approach. The domain size of the interested problem and domain size of the sample problem were selected to be around 40×40 and 10×2.5. The material parameters and domain size were selected in the above example so that it will be possible to perform fully-resolved computations in the microscale to validate the multiscale model results.

In this section, we will use an Al–Cu alloy with material parameters provided in [16]: partition coefficient $k_p = 0.14$, liquidus slope $m_l = -2.6$ K/wt.%, solute diffusion coefficient $D_l = 3000$ μm^2/s, melting temperature $T_m = 933.47$ K, Gibbs–Thomson relation coefficient $\epsilon_c = -0.24$ K μm, surface tension anisotropy $\varepsilon = 0.01$, density of potential nucleation sites d_s^{-2} with $d_s = 9.7$ μm. Here, d_s is the potential nucleation site distance. We model the required undercooling for nucleation as a Gaussian random variable with mean 7.5 K and standard deviation 1.25 K.

The domain size of the sample problem should be in the order of $10d_s$ to $100d_s$ to ensure that enough crystals are included to capture the overall behavior. The grid spacing for the fully-resolved model should be in the order of $0.1d_s$ to ensure that microstructure details can be computed. To satisfy these two constraints, we select our sample problem size as $120\sqrt{2}d_s \times 36\sqrt{2}d_s$ and grid spacing for the fully-resolved model as 0.8574 μm.

Other parameters related with heat transfer (which is not considered in [16]) are: density $\rho = 2400$ kg/m^3, heat capacity $c = 1.06$ kJ/(kg K), heat diffusivity $k = 82.6$ W/(m K), and latent heat $L = 397.5$ kJ/kg. We consider a square region with side length 100 mm (which is the typical size of a casting) filled with the Al–1%Cu alloy at an initial temperature of 970 K. The top side is adiabatic (i.e. $q = 0$), while other three sides are kept at temperature $T_b = 650\exp(-t) + 320$. A schematic of this example is provided in Fig. 20.

Our aim is to obtain the microstructure pattern after solidification and also the temperature field during solidification. If a fully-resolved model is used, the estimated number of grid nodes is $\left(\frac{100\text{ mm}}{0.8574\text{ μm}}\right)^2$ with value about $14G$. If a CFL coefficient 0.1 is used, the estimated number of time steps is about $10 \times \left(\frac{100\text{ mm}}{0.8574\text{ μm}}\right)$ with value $1.2M$. $1.2M$ time steps and $14G$ degrees of freedom is almost impossible to solve with current computational resources. Therefore, we will only provide results of the database approach for this example.

As in the first example, we first use model M to obtain solution features i.e. (V_M, G_M). Using symmetry of the problem, an uniform mesh with size 192×384 is used. Around 30 min are required to perform the computation using model M using one computation node (3.8 G × 2 CPU). The obtained fields of V_M and G_M are shown in Fig. 21. By plotting the obtained data in the V_M and G_M coordinates, we obtain the left plot in Fig. 22. A mesh with 11 nodes is generated to occupy roughly the same region as the obtained (V_M, G_M) data points. For each node of the mesh, we are going to solve a sample problem using the

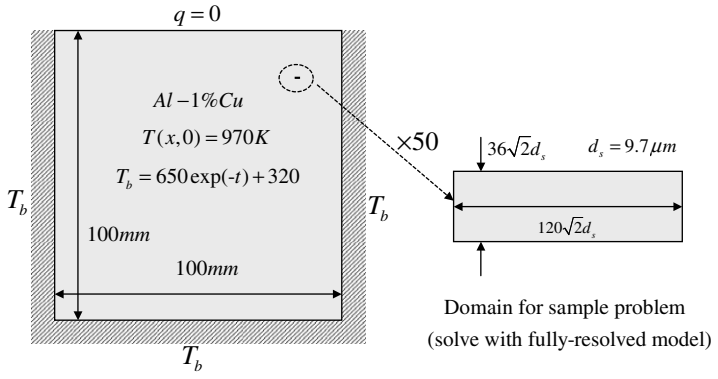


Fig. 20. Schematic of the solidification of an Al–Cu alloy. The very small rectangle inside the elliptic shape is used to demonstrate the relevant size of the domain of the sample problem versus the size of the domain of the problem of interest. It is magnified by 50 times in the right plot.

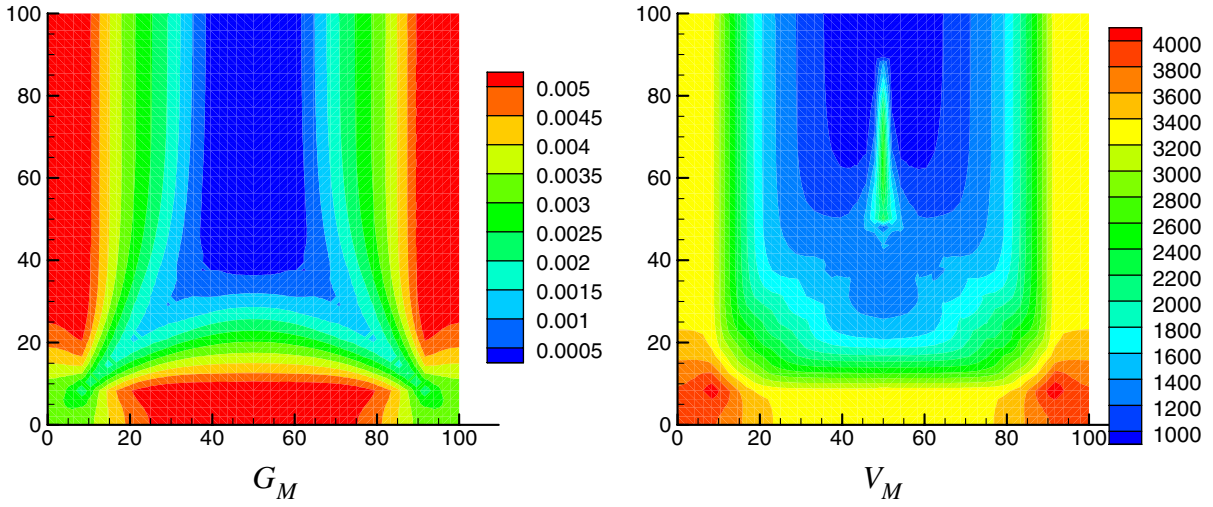
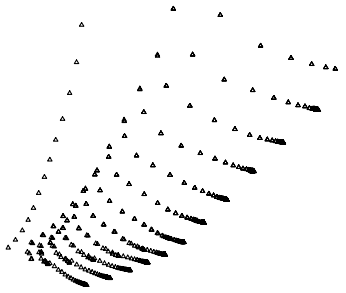


Fig. 21. Obtained G_M (left) and V_M (right) fields using model M. Units of axes, G_M and V_M are mm, K/μm and μm/s, respectively.



fully-resolved model. An adaptive mesh (equivalent to a uniform mesh with size 1920×576) is used for solving the sample problem. Note that in this example, periodic boundary conditions are applied for heat transfer, solute transport, level set evolution when solving the sample problem with the fully-resolved model as demonstrated in Fig. 23. Each sample problem takes about 36 h using one computation node (3.8 G × 2 CPU). Eleven computation nodes are utilized to solve simultaneously all 11 sample problems. Results of a few typical sample problems are shown in Fig. 24. As shown in Fig. 25, the liquid volume fractions for these sample runs are different from the one predicted by the Lever rule, which is widely used in macroscale solidification models. With liquid volume fraction f obtained for each of the sample problems, we are able to use interpolation to obtain f for any given (V_M, G_M) and solve the macroscale model

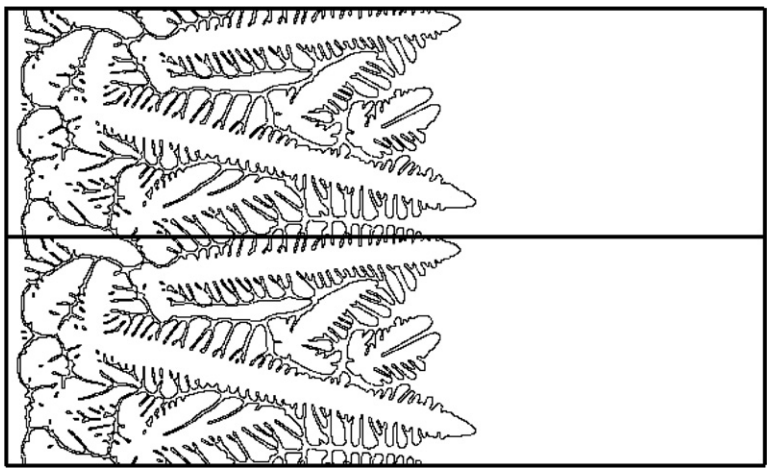


Fig. 23. Demonstration of sample problem domain with periodic boundary conditions applied at the top and bottom sides. The bottom half is the computational domain, the top half is just a copy of the solution from the bottom.

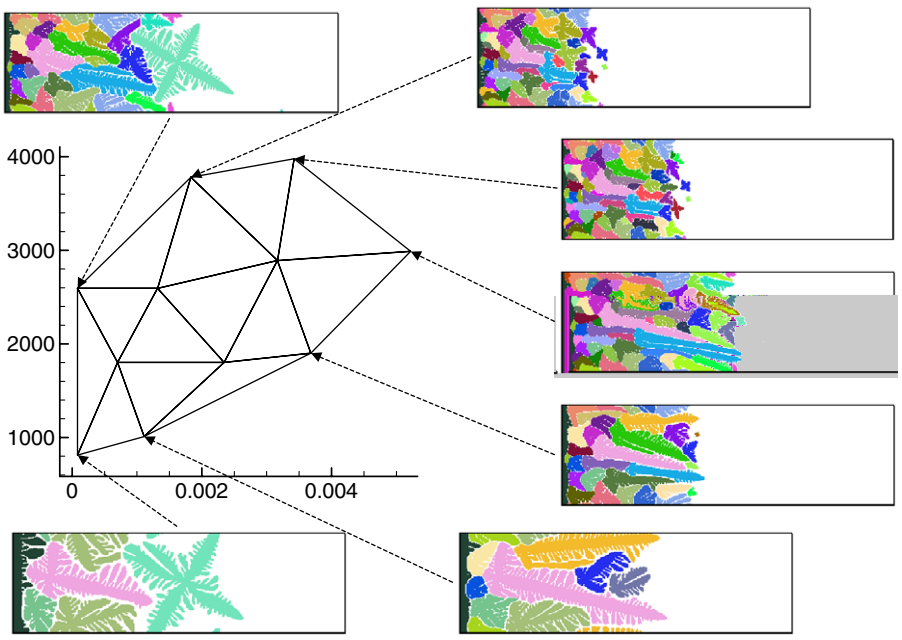


Fig. 24. Sample problem results using the fully-resolved model.

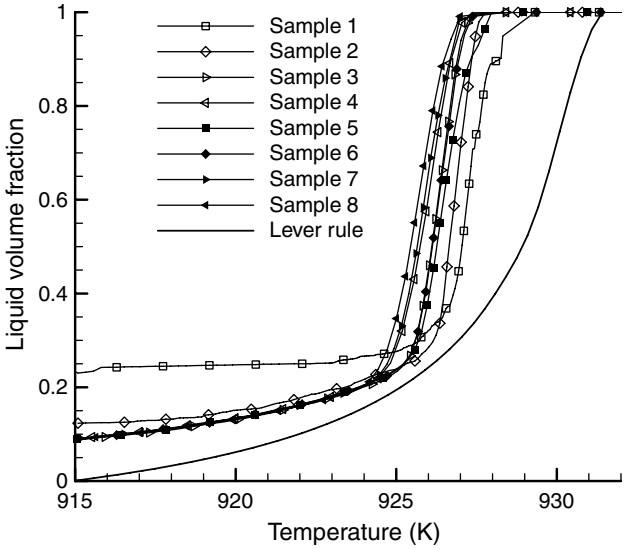


Fig. 25. Relation of volume fraction and temperature.

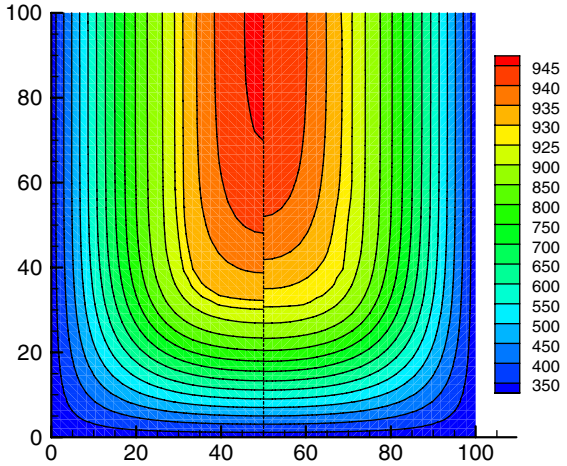


Fig. 26. Temperature field at time 40 s. (Left) Predicted by Lever rule. (Right) Predicted by the database approach.

$\left(\text{i.e. } \rho c\frac{\partial T}{\partial t} = k\nabla^2 T - \rho L\dot{f}\right)$ and perform iterations to improve accuracy of the solution. It takes about 1 hour to finish two more iterations using one computation node (3.8 G × 2 CPU). The predicted temperature at time 12.7 s is shown in the right half of Fig. 26. Comparing with the predicted temperature using Lever rule (shown in the left half of Fig. 26), the temperature in the center of the domain is about 5 K lower. This difference is expected since the volume fraction using the database approach is significantly different from the predicted volume fraction using the Lever rule. Finally, as a post-processing procedure, we can search in the database according to the fields of V_M and G_M to obtain the microstructure. Microstructure in the database with nearest solution features are given in Fig. 27 for eight locations as the location approaches from the side to the center (A–D in Fig. 27). The microstructure pattern changes from fine columnar to coarse columnar and then to coarse equiaxed. As the location approaches from the corner to the center (E–H in Fig. 27), the microstructure pattern changes from fine equiaxed to coarse equiaxed.

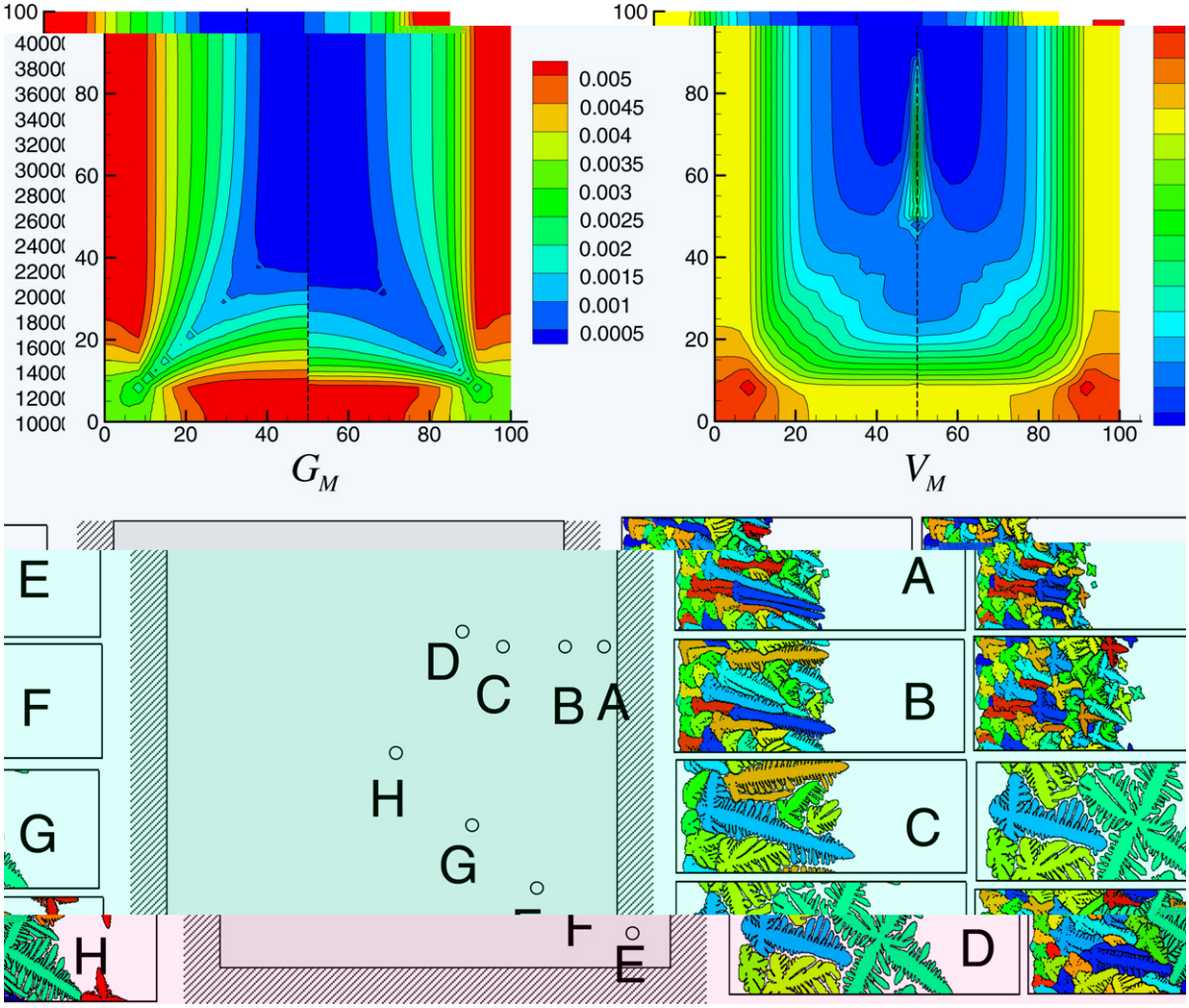


Fig. 27. G_M and V_M fields (left half is result of the first iteration, right half is result after three iterations) and nearest microstructure in database at location A (95 mm, 75 mm), B (90 mm, 75 mm), C (75 mm, 75 mm), D (60 mm, 80 mm), E (90 mm, 10 mm), F (80 mm, 20 mm), G (65 mm, 35 mm), H (50 mm, 50 mm).

6. Conclusions

In this work, we utilize three models for an efficient study of solidification. A computationally efficient model is used to select relevant sample problems. A microscale model is used to generate the database by solving the selected sample problems. A macroscale model is used to efficiently compute solidification with inputs from the database. The results of the macroscale model are further utilized to improve the evaluation of volume fraction and microstructure features. In other multiscale solidification studies in literature, only two models are used: a microscale model to capture microstructure details and a macroscale model to capture global effects. The computationally efficient model plays a crucial role in our multiscale framework. An analytical solution of the computationally efficient model is utilized to define *relevant* sample problems based on solution features obtained from the computationally efficient model. Interpolation is used to greatly reduce the number of sample problems needed. Numerical results of the sample problems using the microscale model are stored using only three fields, solidification time, temperature at solidification time, and orientation angle at solidification time. The proposed multiscale framework for solidification is demonstrated in two cases with the same irregular domain but different boundary conditions. The time consuming microscale model is also solved to

allow comparison with the computational results from the multiscale method. The predicted microstructure type (CET location), microstucture size, and volume fraction using the multiscale method compares well with the microscale model results.

This is the first study of using a database approach for solidification by quantitatively exploring similarities between sample problems and interested problems. The proposed framework with a computationally efficient model to select relevant sample problems is novel and potentially applicable to problems other than solidification. Current efforts are on applying this framework to three-dimensional solidification and extending it to solidification systems with convection.

References

[1] 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.
[2] T. Takaki, T. Fukuoka, Y. Tomita, Phase field simulation during directional solidification of a binary alloy using adaptive finite element method, J. Cryst. Growth 283 (2005) 263–278.
[3] B. Nestler, D. Danilov, P. Galenko, Crystal growth of pure substances: phase-field simulations in comparison with analytical and experimental results, J. Comput. Phys. 207 (2005) 221–239.
[4] L. Tan, N. Zabaras, Modeling the interaction of multiple dendrites in solidification using a level set method, J. Comput. Phys. 226 (2007) 131–155.
[5] W.M. Feng, P. Yu, S.Y. Hu, Z.K. Liu, Q. Du, L.Q. Chen, Spectral implementation of an adaptive moving mesh method for phase-field equations, J. Comput. Phys. 220 (2006) 498–510.
[6] C.Y. Wang, C. Beckermann, Equiaxed dendritic solidification with convection: Part I. Multiscale/multiphase modeling, Metall. Mater. Trans. A 27 (1995) 2754–2764.
[7] C. Eck, P. Knabner, S. Korotov, A two-scale method for the computation of solid–liquid phase transitions with dendritic microstructure, J. Comput. Phys. 178 (2002) 58–80.
[8] P.D. Lee, A. Chirazi, R.C. Atwood, W. Wang, Multiscale modelling of solidification microstructures, including microsegregation and microporosity, in an Al–Si–Cu alloy, Mater. Sci. Eng. A 365 (2004) 57–65.
[9] H. Rafii-Tabar, A. Chirazi, Multiscale computational modelling of solidification phenomena, Phys. Rep. 365 (2002) 145–249.
[10] W. E, B. Engquist, The heterogeneous multi-scale methods, Commun. Math. Sci. 1 (2003) 87–133.
[11] 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.
[12] D. Samanta, N. Zabaras, Macrosegregation in the solidification of aluminum alloys on uneven surfaces, Int. J. Heat Mass Transfer 48 (2005) 4541–4556.
[13] D. Samanta, N. Zabaras, Modeling melt convection in solidification processes with stabilized finite element techniques, Int. J. Numer. Method Eng. 64 (2005) 1769–1799.
[14] W. Kurz, D.J. Fisher, Fundamentals of Solidification, third ed., Trans Tech Publications Ltd., Switzerland, 1989.
[15] J.D. Hunt, Steady state columar and equiaxed growth of dendrites and eutectic, Mater. Sci. Eng. 65 (1984) 75–83.
[16] A. Badillo, C. Beckermann, Phase-field simulation of the columnar-to-equiaxed transition in alloy solidification, Acta Mater. 54 (2006) 2015–2026.
[17] D.L. Ceynar, C. Beckermann, Measurement of the density of succinonitrile–acetone alloys, J. Cryst. Growth 222 (2001) 380–391.