

A sharp-interface model coupling VOSET and IBM for simulations on melting and solidification

Kong Ling^{a,b}, Wen-Quan Tao^{a,*}

^aKey Laboratory of Thermo-Fluid Science & Engineering of MOE, Xi'an Jiaotong University, Xi'an, China

^bDepartment of Mechanical, Material and Aerospace Engineering, Illinois Institute of Technology, Chicago, Illinois, USA



ARTICLE INFO

Article history:

Received 21 March 2018

Revised 28 August 2018

Accepted 30 August 2018

Available online 1 September 2018

Keywords:

VOSET

Immersed boundary method

Melting

Solidification

ABSTRACT

This article presents a sharp-interface model for solving solid-liquid phase change of pure materials. An interface capturing method combining VOF and level set, VOSET, is employed to track the moving interface during phase change, and the level-set function generated in VOSET is used to consider the effect of the phase boundary when solving the fields of temperature and velocity. In order to deal with the interaction between the solid and liquid phases, an immersed boundary method (IBM) is incorporated into SIMPLER algorithm. The fundamental numerical approaches presented in this article were individually assessed by test problems, all showing good agreements with benchmark solutions. The established sharp-interface model was then applied in simulating melting and solidification without and with fluid flow, where competitive performance in accuracy was demonstrated by the numerical results.

© 2018 Elsevier Ltd. All rights reserved.

1. Introduction

Melting and solidification are widespread in nature and industries such as lava evolution, casting and energy storage. It is a comprehensive process involving fluid flow, heat transfer and phase change, which makes it extremely difficult to solve analytically. Fortunately, the fast developing numerical methods provided us many powerful tools for studying those problems. An important consideration in direct simulations on those problems is how to describe the solid-liquid interface as well as its movement during phase change.

So far, varieties of numerical methods have been proposed for the simulation of melting and solidification. A major group among them uses interface tracking methods, which primarily include front tracking, volume-of-fluid (VOF), and level-set (LS) methods, to handle the interface movement during phase change. In addition, enthalpy method and phase-field method (PFM) are also widely used approaches for solid-liquid phase change simulation.

Front tracking method handles the interface movement by tracking the marker points placed on the phase boundary. It is usually incorporated with the use of a fixed underlying grid where heat conduction is solved. Juric and Tryggvason [1] incorporated a front tracking method into a finite difference method and conducted numerical simulation on some problems involving heat

conduction and solidification. The heat released during solidification was distributed in the computational cells within a certain distance to the interface, and a source term was added in the heat conduction equation. On this basis, Udaykumar et al. [2] established a sharp-interface model for solidification simulation, which considered phase change on a zero-thickness interface. Udaykumar and Mao [3] simulated a solidification process in a solution using a front tracking method, where several physical processes were considered including heat transfer, mass transfer and phase change. Al-Ravahi and Tryggvason [4] applied a front tracking method in simulating a solidification process of pure material considering the effects of anisotropy as well as the liquid flow. These numerical methods were later extended in three dimensions [5]. Zhao and Heinrich [6] incorporated front tracking method into finite element method to simulate the solidification of pure materials. This method was further developed for three dimensions [7] and multiple components [8]. Hu et al. [9] developed a computational model based on front tracking method for solid-liquid phase transition, and applied it in simulating dissolution, precipitation and melting. Huang et al. [10,11] developed a combination of front tracking method and lattice Boltzmann method, where an immersed boundary method was employed for solid-liquid interaction. Front tracking method has revealed competitive accuracy in various numerical studies on solidification and melting simulations. Another important advantage of this method is being easy to handle the interface sharply, since the interface moving velocities can be computed directly on the marker points. However, the implementation of front tracking method is more complicated than scalar-field

* Corresponding author at: 28 Xian Ning Road, Xi'an, Shaanxi, 710049, China.
E-mail address: wqtao@mail.xjtu.edu.cn (W.-Q. Tao).

methods [12]. It requires a well-designed data structure to describe the marker points and the connections between them. As the interface experiences large deformation, one needs to delete old or insert new marker points to maintain appropriate distances between them. Another significant challenge lies in the algorithms dealing with interface topological changes like merging and splitting. In three dimensions, those challenges becomes remarkably greater.

Level-set (LS) method [13] employs a continuous scalar field, namely level-set function, to describe the two phases with its zero contour representing the interface. The interface is tracked by solving a hyperbolic evolution equation with a given velocity field. Usually, the signed distance function was selected as the level-set function. Some researchers demonstrated the applicability of level-set method in the simulating solidification. Chen et al. [14] employed a level-set method in a finite difference method, and some phase change problems including Stefan problem and unstable solidification were numerically investigated. Tan and Zebars [15] adopted level-set method to simulate solidification into undercooled liquid considering the influence of the liquid flow in two- and three-dimensions. Rauschenberger et al. [16] implemented LS method in the open source code OpenFOAM. It was then used to simulate a crystal ice growth in undercooled water, obtaining tip growth velocities consistent with marginal stability theory [17,18]. Ghoneim [19] adopted level-set method in finite element method to establish a meshfree interface model for phase change simulation, which was then applied for dendritic solidification in structured and unstructured grids. Level-set method is relatively simple in computational procedure, and it has a sharp description on the interface. However, LS method is also popularly known for its poor performance in mass conservation.

VOF method [20] was originally proposed for liquid-gas multiphase flow, where the liquid-gas interface shifts with the fluid flow. López et al. [21] proposed a VOF method for the simulation of solidification, in which an equivalent velocity field is generated around the phase boundary according to the phase change rate. This method handles the solid-liquid interface diffusively by assuming phase change occurs in a zone around it with a given thickness. Improved accuracy was achieved by the use of a two-grid approach proposed by López et al. [22], in which the volume fraction is solved on the finer grid. Until now, we can find many sharp-interface VOF models for liquid-gas multiphase flows [23–25], but very few for solid-liquid phase transition. Rauschenberger and Weigand [26] tracked the interface based on reconstructed interfaces and the phase change rate. However, this method has a minor flaw that the volume fractions are updated only in cells containing interfaces, thus a single-phase cell cannot develop into an interfacial one even if the interface actually passes by. Therefore, an examination procedure was added in each time step to see whether the interface is getting approach to single-phase cells, and artificial modifications were made when needed.

Enthalpy method is another option for solid-liquid phase change simulation, in which the energy equation is solved in an enthalpy form containing both specific and latent heat. Pal et al. [27] proposed an enthalpy method and applied it in simulating crystal growth in undercooled liquid. Mencinger [28] used enthalpy method to simulate melting problems with natural convection with different dimensionless numbers, where an adaptive grid refinement was used to enhance the precision. Ulvrova et al. [29] used an enthalpy method to compute a melting problem and the result was compared with that obtained by a moving-mesh method. Karagadde et al. [30] developed a coupled numerical model to simulate dendrite growth of solid seeds along with their settlements in melt. In this model, enthalpy method and VOF are individually employed to deal with the phase change and the movement of solid phase; IBM was used to solve the flow field. Wu et al. [31] developed a lattice Boltzman model for solid-liquid

phase change in porous media, in which the energy equation was solved in an enthalpy-based approach.

Phase-field method (PFM), which has been applied in many numerical studies on dendritic solidification, uses an auxiliary function that smoothly varies from zero to unity across the interface. This method has important physical mechanisms, for the governing equations are derived from thermodynamic potentials of the system. Beckermann et al. [32] applied phase-field method for dendritic growth, where the influence of the fluid flow was investigated. Do-Quang and Amberg [33] applied a semi-sharp phase-field method in an adaptive mesh. The computational model was used to simulate the solidification of on a dendrite as it moves freely in the melt. Zhao et al. [34] developed a phase-field method for macro-scale melting problems.

A common advantage of phase field method and enthalpy method is that no explicit interface tracking algorithm is required. Using either method, the simulation on a phase change process can be accomplished only by solving a series of partial differential equations, which greatly simplifies the numerical procedure. However, both PFM and enthalpy method are inherently diffuse-interface models due to the lack of a geometrical description on the interface position. By a standard phase field method, accurate results can be obtained only when the interface thickness is strictly small. As the cost, however, it takes remarkably more computational resources, especially in macro-scale problems. Whereas, It deserves to be mentioned that phase field method has been greatly improved in this regard with the efforts of many researchers, such as Karma and Rappel [35] who introduced a thin-interface version of PFM, and Amberg [36] who developed a semi-sharp PFM.

The objective of the present study is to develop a simple yet accurate computational model for solid-liquid phase change with interface described sharply. We employed a recently developed interface tracking method, VOSET, [37,38] to handle the moving interface owing to that volume fraction and level-set function are both available in computations. As part of VOSET, VOF has a sharp description on the interface when an interface reconstruction process is included. The level-set function generated in VOSET can not only accurately calculate interface normal and curvature in a simple way [37], but also perform as convenient tool for interpolations regarding phase boundary. However, in the originally proposed VOSET, the VOF part is designed for liquid-gas free surface flow. Therefore, we will in this article present a new VOF-based interface tracking method to make it better adapted to the features of interface movement in solid-liquid phase change. To consider the influence of fluid flow, a simple method will be introduced to incorporate immersed boundary method (IBM) into SIMPLER algorithm. Finally, we would like to point out that, despite many differences between melting and solidification processes in phenomenon, the fundamental physical problems involved are the same. A numerical framework designed for the simulation of solidification should be applicable to melting, and vice versa. In the present study, therefore, the proposed computational model were applied in both melting and solidification problems.

The rest of this article is organized as follows. In Section 2, the physical model and assumptions concerning on solid-liquid phase change are introduced. Section 3 describes the numerical methods including VOF-based interface tracking, solving the temperature field and the incorporation of IBM into SIMPLER. Section 4 presents the applications for some specific problems including both melting and solidification. Finally, the conclusions are drawn in Section 5.

2. Physical description

Here we consider the solid-liquid phase change of a pure substance with fluid flow taken into consideration. Fig. 1 sketches a concerned domain containing solid and liquid phases separated by

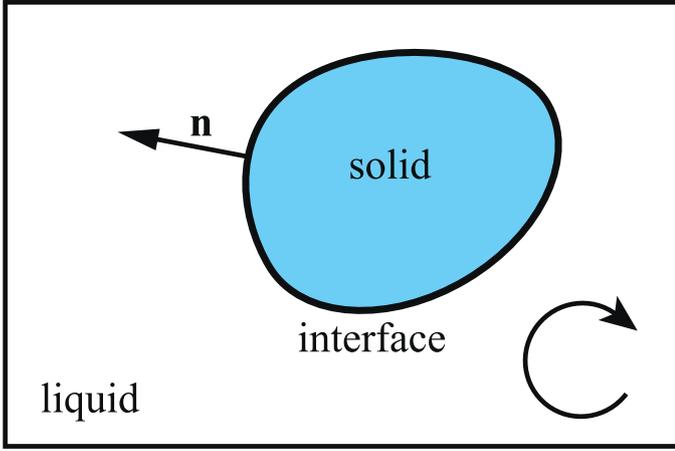


Fig. 1. Schematic of the solid-liquid phase change problem.

an interface. The unit normal \mathbf{n} , perpendicular with the interface, is defined pointing into the liquid phase. The solid-liquid interface moves owing to the phase change occurring on it. The solid phase keeps stationary, whereas, there may be fluid flow in the liquid region due to natural or forced convection.

Here we assume the solid and liquid phases both have constant properties such as density, thermal conductivity, specific heat etc., but those properties may differ in different phases.

Under the constant-property assumption, the temperature equation can be expressed as:

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \nabla \cdot (\alpha_S \nabla T) \quad \text{for solid phase,} \quad (1a)$$

and

$$\frac{\partial T}{\partial t} + \nabla \cdot (\mathbf{u}T) = \nabla \cdot (\alpha_L \nabla T) \quad \text{for liquid phase.} \quad (1b)$$

The temperature has a certain value on the solid-liquid interface depending on phase equilibrium condition:

$$T = T_i \quad (1c)$$

In Eq. (1), α represents the thermal diffusivity defined by $\alpha = k/\rho c_p$, \mathbf{u} is the fluid velocity. Subscripts S , L and I denote solid, liquid and interface, respectively.

Through a thermodynamic analysis [39], the interface temperature can be determined by Gibbs–Thomson condition expressed as:

$$(T_i - T_m) + \frac{T_m(c_{pL} - c_{pS})}{L} \left(T_i \ln \frac{T_i}{T_m} - T + T_m \right) + \frac{T_m \sigma}{L \rho} \kappa + \frac{V_n}{\nu} = 0 \quad (2)$$

In Eq. (2), the interfacial temperature and the melting point are denoted by T_i and T_m , respectively. σ is the surface tension, and L is the latent heat. κ is the interface curvature which takes a positive value at a convex point on the solid interface. The influence of kinetic mobility is taken into consideration by the last term, in which V_n is the growth velocity of the solid phase and ν is the coefficient for kinetic mobility.

When phase change occurs under a small temperature difference, Eq. (2) can be simplified as:

$$(T_i - T_m) \left[1 + \frac{(c_{pL} - c_{pS})(T_i - T_m)}{L} \right] + \frac{T_m \sigma}{L \rho} \kappa + \frac{V_n}{\nu} = 0, \quad (3)$$

since $\ln(T_i/T_m) \approx (T_i/T_m) - 1$.

In the present study we consider the case when:

$$\frac{(c_{pL} - c_{pS})(T_i - T_m)}{L} \ll 1. \quad (4)$$

Thus Eq. (3) can be finally reduced to the classic Gibbs–Thomson condition:

$$T_i = T_m + \epsilon_c \kappa + \epsilon_v V_n, \quad (5)$$

where $\epsilon_c = -T_m \sigma / L \rho$ accounts for the effect of surface energy, and $\epsilon_v = -1/\nu$, which is named as Gibbs–Thomson coefficient, corresponds to the effect of kinetic mobility.

The interface moves only when phase change occurs, thus the interface moving velocity can be determined by Stefan condition which relates the phase change rate with the temperature gradients on the phase boundary.

$$\rho [L + (T_i - T_m)(c_{pL} - c_{pS})] V_n = k_S \left(\frac{\partial T}{\partial n} \right)_S - k_L \left(\frac{\partial T}{\partial n} \right)_L \quad (6)$$

Note that the specific heats are included in the left-hand side. It keeps a constant enthalpy change during phase change with the interface temperature not equaling to the melting point.

The velocity and pressure field in the liquid phase can be described by incompressible continuity equation and Navier–Stokes equation:

$$\nabla \cdot \mathbf{u} = 0 \quad (7)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \mathbf{u}) = -\nabla p + \nabla \cdot (\eta \nabla \mathbf{u}) + \mathbf{f} \quad (8)$$

In Eq. (8) the pressure is denoted by p , and η is the viscosity of the liquid phase. \mathbf{f} denotes the volume force, such as gravity, imposed on the fluid phase.

According to mass conservation, the liquid flow velocity at the phase boundary is related to the phase growth rate (V_n), density ratio (ρ_S/ρ_L) and the interface normal (\mathbf{n}):

$$\mathbf{u}_I = \left(1 - \frac{\rho_S}{\rho_L} \right) V_n \mathbf{n} \quad (9)$$

If the solid and the liquid phases have the same density, i.e., $\rho_S/\rho_L = 1$, obviously, the liquid velocity at interface equals to zero.

3. Numerical methods

3.1. Level-set function by VOSET

In VOSET method, a scalar field respecting the volume fraction of the reference phase in a computational cell is used to describe the distribution of the two phases. Here, the scalar function, denoted by C , is defined as the volume proportion of solid phase. Therefore, the *volume-of-fluid* in the present actually refers to volume of solid. It takes the value of 1 in a computational cell filled with solid phase and 0 in one filled with liquid phase, and $0 < C < 1$ denotes the cell contains both phases. The evolution of the interface can be described by the standard advection equation of the volume fraction:

$$\frac{\partial C}{\partial t} + \mathbf{u}_I \cdot \nabla C = 0 \quad (10)$$

in which \mathbf{u}_I represents the interface moving velocity.

VOSET successfully combines the advantages of VOF and LS approaches by solving both volume fraction and level-set function. It can not only keep mass conservation well, but also calculate the interface normal and curvature accurately. An important feature of VOSET is that the level-set function is calculated in a geometrical way without the need of discretization of its governing equation.

Concretely, with a given field of volume fraction, VOSET calculates a corresponding level-set function by carrying out the following steps iteratively:

- (1) Calculate the interface normal and reconstruct the interface;
- (2) Mark the computational cells around interfaces within a certain distance;

(3) Calculate the nearest distances to the reconstructed interfaces in the marked cells.

Note, in Step (1) the interface is reconstructed as a set of line segments using Piecewise Linear Interface Calculation (PLIC) approach [40]. In the first iteration, since the level-set function is not available, the volume fraction is used to calculate the interface normal, and in the following iterations, the interface normal is calculated by the level-set function calculated in Step (3) at the last iteration. The purpose of Step (2) is to save computational resources since in most cases level-set function is needed only around the phase boundary. The generated level-set function is actually a signed distance function with its absolute value equalling to the nearest distance to the interface. One can refer to Refs. [37], [38] and [41] for more details of this geometrical method.

3.2. Interface normal and curvature

Using the level-set function calculated in VOSET, the interface normal \mathbf{n} and curvature κ are computed by:

$$\mathbf{n} = \frac{\nabla\phi}{|\nabla\phi|} \tag{11}$$

and

$$\kappa = \nabla \cdot \frac{\nabla\phi}{|\nabla\phi|} \tag{12}$$

respectively.

Apart from accuracies in the calculations of normal and curvature, the level-set function provides a convenient way in the linear interpolations of temperature and velocity around the phase boundary.

3.3. Tracking the moving interface

The interface tracking method presented here, which performs as a part of VOSET, is based on an unsplit VOF scheme proposed by López et al. [42], but not equivalent to it. In the method by López et al. [42], “flux polygons” are created based on the velocities at the cell vertices which are interpolated in prior from those at cell centers nearby. This technique can avoid overlaps between the flux polygons.

Since the solid phase is assumed to be stationary, its volume does not shift between computational cells. Instead, it grows/shrinks on the interfaces as solidification/melting occurs. However, traditional VOF method can also be applied in solid-liquid phase change problems by setting an equivalent velocity field and supposing the solid volume “flows” through the cell faces. The artificially generated velocity field, therefore, should be consistent with the velocity of the interface movement produced by the phase change.

Our method also tracks the interface by the velocities at cell vertices, but they are interpolated from the interface moving velocities on nearby reconstructed interfaces. The method interpolating velocities from phase boundary to cell vertices is illustrated in Fig. 2. For a given cell vertex like point O in Fig. 2, the 4×4 cells around it are checked whether they contain interface segments. If no interface is found, i.e., all these cell are occupied by pure liquid or solid phase, a zero velocity is set on the cell vertex. Otherwise, the velocity on point O is computed by a weighted interpolation considering the distance to the interfaces:

$$\mathbf{v}_O = \frac{\sum \mathbf{v}_i/d_i}{\sum 1/d_i} \tag{13}$$

In this equation, \mathbf{v}_i denotes the moving velocity of reconstructed interface i , and d_i refers to the distance from its center to point O.

Provided the interface has been reconstructed in the current time step (C^n) and that the velocities at the grid vertices have been

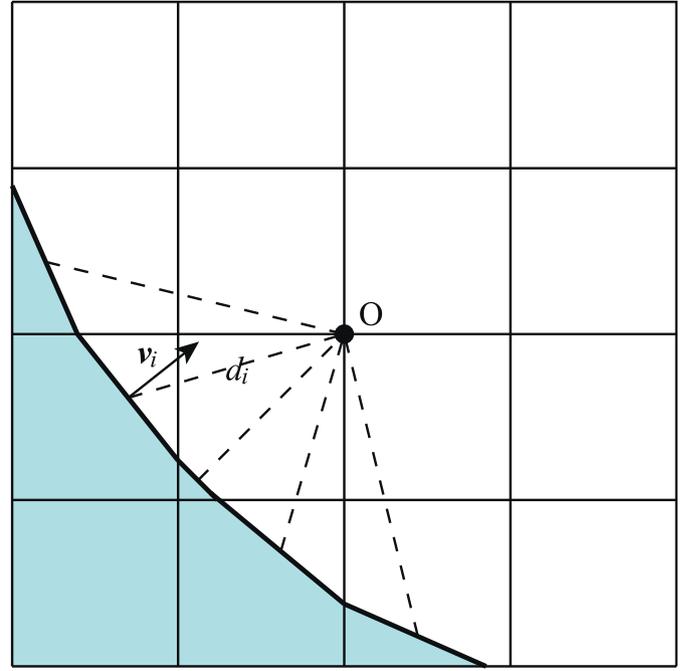


Fig. 2. Interpolating the velocity at a grid vertex from nearby interfaces.

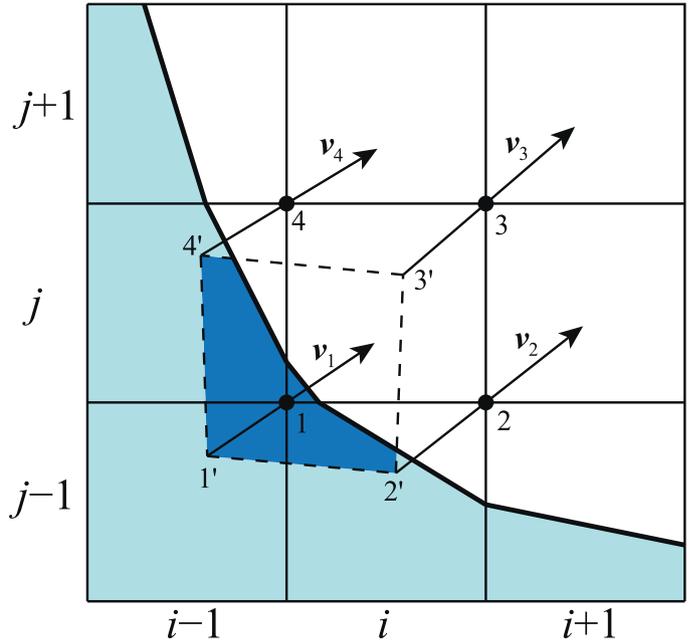


Fig. 3. Updating volume fraction in a computational cell.

calculated, the volume fraction in a computational cell is updated in the way illustrated in Fig. 3. We consider the four vertices (see points 1, 2, 3 and 4 in Fig. 3) around cell (i, j) in which the volume fraction needs to be updated. We first find four corresponding points 1',2',3' and 4' along the opposite directions of the velocities at those grid vertices, whose coordinates are calculated by:

$$\mathbf{x}(k') = \mathbf{x}(k) - \mathbf{v}_k \Delta t \quad \text{for } k = 1, 2, 3, 4 \tag{14}$$

From Eq. (14) we can see that, under the velocities at those grid vertices, points 1',2',3' and 4' will respectively move to points 1, 2, 3, and 4 in Δt , and polygon 1'2'3'4' will occupy the computational cell (i, j) then. Therefore, the volume fraction of cell (i, j) at the next time step can be approximated as the solid volume portion in

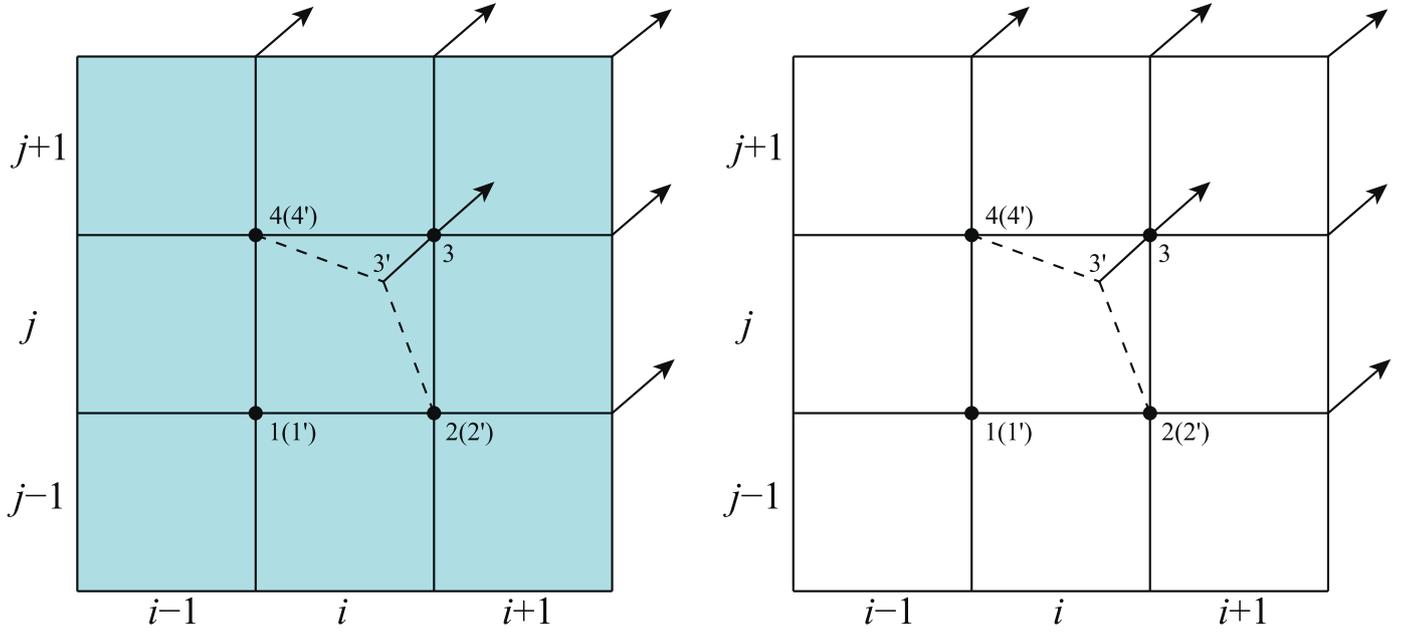


Fig. 4. Computing the volume fraction in a single-phase cell where the discrete velocity is not divergence free. The solid cell keeps $C = 1$ and the liquid cell keeps $C = 0$.

polygon $1'2'3'4'$ at the present time step:

$$C_{i,j}^{n+1} = \frac{F_{1'2'3'4'}}{S_{1'2'3'4'}} \quad (15)$$

In this equation, $S_{1'2'3'4'}$ refers to the entire area of the shifting polygon $1'2'3'4'$, and $F_{1'2'3'4'}$ denotes the area occupied by solid phase in this polygon. Here we need two geometrical techniques to calculate those areas appearing in Eq. (15). First, the area of a given polygon needs to be calculated from a list of its vertices' coordinates, and Eq. (5) in Ref. [43] is used for the calculation. Second, the intersection between the polygon $1'2'3'4'$ and a solid-phase polygon inside a computational cell needs to be calculated. Note that, the solid-phase polygon is in some cases (such as $(i-1, j-1)$ in Fig. 3) equivalent to a rectangular cell, and in other cases (such as $(i, j-1)$ in Fig. 3) takes only a part of the cell as it is partially bounded by an interface segment. Since a phase polygon in our model is invariably convex, the intersection between two polygons is resolved by Sutherland–Hodgman algorithm [44] composed of several polygon-clipping procedures.

As can be seen that, a single formula written as Eq. (15) is adopted to update the volume fractions. Apart from the simplicity, this formula brings out more advantages in handling some troubles which usually arise in VOF. According to the geometrical meaning in the right hand of Eq. (15), the numerator varies from zero to the value of the denominator, which automatically guarantees the boundedness of the volume fraction, i.e., $0 \leq C_{i,j}^{n+1} \leq 1$. More importantly, this approach can automatically avoid the wisps resulting from non-divergence-free velocity field when using “phase fluxes” in traditional VOF methods. Those wisps can be avoided by a divergence correction procedure when calculating volume fractions [42,43]. However, such correction is not needed in our method since the calculation of volume fractions is not based on “phase flux”. This is particularly important in the present study where the interpolated “velocity field” is none-zero only around the phase boundary and indeed not divergence free. Here we use two examples shown in Fig. 4 to illustrate how the wisp is avoided in a cell with non-divergence-free discrete velocities. The considered cell (i, j) is occupied by pure solid in the left figure and pure liquid in the right one. When applying Eq. (15) to update the volume fraction, the numerator (the solid phase area in the shifting

polygon) equals to the denominator (the entire area of the shifting polygon) in Fig. 4(a) and equals to zero in Fig. 4(b). Therefore, it remains $C_{i,j}^{n+1} = 1$ in the solid cell and $C_{i,j}^{n+1} = 0$ in the liquid cell. Another advantage we would like to point out is our model can automatically transform cell status between single-phase and interfacial, and therefore the artificial modification procedure adopted by Rauschenberger and Weigand [26] is not needed.

As a short summary, the proposed interface tracking method based on VOF has advantages in robustness and simplicity. The validation on some test problems suggests the proposed method has a nearly 2nd order accuracy (see Appendix A for details).

3.4. Solving the temperature equation

The temperature field is calculated in a two-phase approach in which computational cells are handled differently depending on whether they contain interfaces. Owing to many discontinuous features across the solid-liquid interface, we would like to present more details in solving the temperature equations.

Before calculating of the temperature field, the computational cells are categorized into two types, which is shown in Fig. 5. The first type, namely single-phase cell, is occupied by a single phase, either liquid or solid. A cell in the second type is called interfacial cell here for it contains both phases and therefore has a piece of interface inside.

3.4.1. Single-phase cells

In a single-phase cell, the standard finite volume method (FVM) [45,46] is used to discretize the temperature equation (Eq. (1)). Utilizing a C-N (Crack-Nickson) scheme for the time advancement, a semi-discretized temperature equation can be expressed as:

$$\begin{aligned} \frac{T^{n+1} - T^n}{\Delta t} + \frac{1}{2} [\nabla \cdot (\mathbf{u}^{n+1/2} T^n) + \nabla \cdot (\mathbf{u}^{n+1/2} T^{n+1})] \\ = \frac{1}{2} \alpha [\nabla \cdot (\nabla T^n) + \nabla \cdot (\nabla T^{n+1})], \end{aligned} \quad (16)$$

where the superscript n refers to the number of time step. The velocity at the middle time is estimated by:

$$\mathbf{u}^{n+1/2} = \frac{\mathbf{u}^n + \mathbf{u}^{n+1}}{2} \quad (17)$$

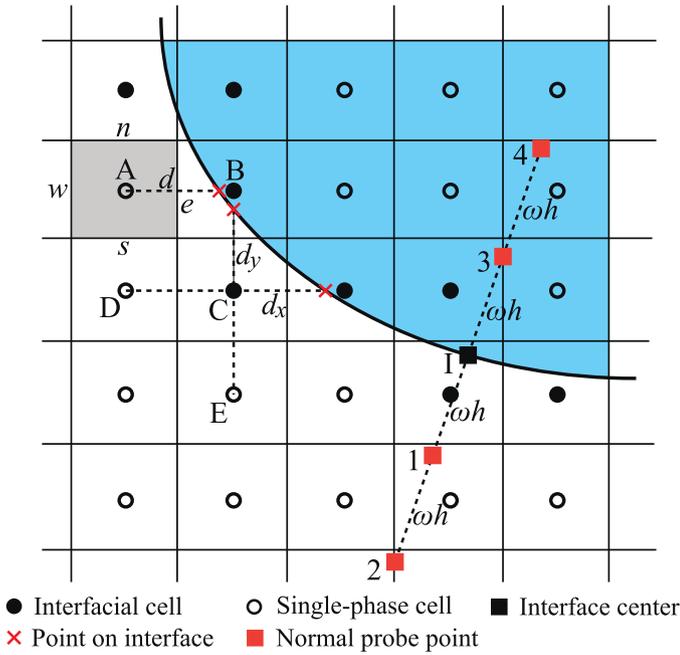


Fig. 5. Methods for solving temperatures in different types of cells and normal probe approach for estimating the temperature gradients on the phase boundary.

Integrating over control volume, the convection and diffusion terms can be individually discretized as:

$$\int_{CV} \nabla \cdot (\mathbf{u}T) dV \cong \sum_{f \sim CV} T_f \mathbf{u}_f \cdot \mathbf{A}_f \quad (18)$$

and

$$\int_{CV} \nabla \cdot (\nabla T) dV \cong \sum_{f \sim CV} (\nabla T)_f \cdot \mathbf{A}_f, \quad (19)$$

in which subscripts CV and f refer to the control volume of a computational cell and its cell faces. The temperatures on cell faces appearing in Eq. (18) are interpolated from nearby cell-center values using MUSCL scheme (monotonic upwind scheme for conservation law) [47], and central-difference scheme is taken in calculating the temperature gradients in Eq. (19).

Attention needs be paid to a special case in which a single-phase node has a neighbor located in the other phase. Cell A in Fig. 5 is an example of this case. In this situation, one needs to consider the effect of the interface between the two nodes when approximating the temperature and its gradient on cell face. Specifically, the temperature and its gradient on face e (the face separating cells A and B) are estimated by:

$$T_e \cong \frac{0.5hT_I + (d - 0.5h)T_A}{d} \quad (20)$$

and

$$(\nabla T)_e \cdot \mathbf{A}_e = \left(\frac{\partial T}{\partial x} \right)_e \cong \frac{T_I - T_A}{d}, \quad (21)$$

respectively.

In Eqs. (20) and 21, h is the size of the uniform Cartesian computational cell, and d is the distance between node A to the phase boundary in x direction, which is estimated by the level-set functions.

$$d \cong \frac{|\phi_A|}{|\phi_A| + |\phi_B|} h \quad (22)$$

3.4.2. Interfacial cells

Now attention is turned to the second type of the computational cells, namely interfacial cells. In these cells, the discrete temperature equations are established by a linear interpolation from the phase boundary rather than the discretization of the governing equation. For example, the temperature of node C in Fig. 5 satisfies:

$$T_C - T_I = \frac{d_x}{d_x + h} (T_D - T_I) \quad (23)$$

In some cases, one may have more than one choice to interpolate the temperature of an interfacial node. For node C in Fig. 5, the node at its top, i.e., node B, is also located in the other phase. In this situation, we choose the direction in which the node is the closer to the phase boundary. Thus, the temperature equation for node C should be replaced by:

$$T_C - T_I = \frac{d_y}{d_y + h} (T_E - T_I) \quad (24)$$

in case $d_y < d_x$. Here d_y is also calculated by level-set functions.

3.5. Temperature gradients on the phase boundary

Temperature gradients need to be estimated to compute the interface moving velocity induced by the phase change (Eq. (6)). In this regard, the normal probe technique proposed by Udaykumar et al. [2] is applied. As illustrated in Fig. 5, two points are found along the interface normal at ωh and $2\omega h$ for an interface segment from its center point in each phase. Following the recommendation by Al-Ravahi and Tryggvason [5], we used $\omega = 1.2$ in the present study. The temperatures at those points are then estimated from nearby cells using a bilinear interpolation, after which the temperature gradients on the two sides of the interface can be computed by linear or quadratic approximations [6] displayed as follows.

- Linear approximation

$$\left(\frac{\partial T}{\partial n} \right)_S \cong \frac{T_1 - T_I}{\omega h} \quad (25a)$$

$$\left(\frac{\partial T}{\partial n} \right)_L \cong -\frac{T_3 - T_I}{\omega h} \quad (25b)$$

- Quadratic approximation

$$\left(\frac{\partial T}{\partial n} \right)_S \cong \frac{-T_2 + 4T_1 - 3T_I}{2\omega h} \quad (26a)$$

$$\left(\frac{\partial T}{\partial n} \right)_L \cong -\frac{-T_4 + 4T_3 - 3T_I}{2\omega h} \quad (26b)$$

Substituting Eq. (25) or Eq. (26) into Eq. (6), one can obtain the velocity of the interface movement. The curvature of the interface at its center point obtained by bilinear interpolation from nearby cell-center values is then used to evaluate the surface energy effect Eq. (5). If the kinetic effect cannot be omitted, i.e., ϵ_V in Eq. (5) is not zero, Eqs. 5 and 25 (or 26) should be combined to solve the interface temperature and the moving velocity.

3.6. Solving the flow field

The flow field is solved by SIMPLER algorithm [45] in staggered grid. In the discretization of the momentum equation, MUSCL [47] and CD (central-different) schemes are employed for convection and diffusion terms, respectively. We used SIMPLER method rather than projection method [12] because it allows greater time intervals in unsteady flow simulation, which saves much computational time.

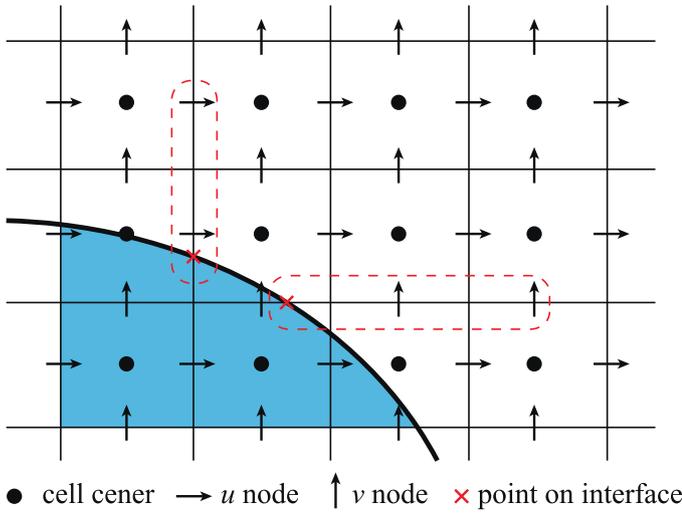


Fig. 6. Method calculating the desired velocity in staggered grid.

The interaction between the solid and liquid phases needs to be taken into consideration when solving the flow field. The velocity in the solid phase should be maintained zero for it is assumed stationary. In this regard, Al-Ravahi and Tryggvason [5] added a simple procedure when implementing projection method: the unprojected velocity obtained in the first step without considering the pressure gradient was set zero in the solid phase. Another approach one can choose is to set a relatively large viscosity in the solid region [27,29]. For higher accuracy, we used an immersed boundary method (IBM) proposed by Fadlun et al. [48]. Linear interpolation is applied for the velocity around the phase boundary, which has been demonstrated to have a second-order accuracy [48,49]. Here we shall present a method to incorporate IBM into SIMPLER algorithm we used to solve the flow field.

A commonly used way to apply the immersed boundary method is to add artificial forces to the momentum equation in the nodes located inside the solid phase as well as around the phase boundary such that the velocities there can be adjusted to the desired values. These nodes, either in the solid region or in vicinities of the phase boundary, should be marked in prior. Here, the level-set function is used to determine which phase a given node is located in. The velocity nodes, as Fig. 6 shows, are defined on cell faces rather than cell centers in a staggered grid. The level-set functions on the velocity nodes, therefore, should be approximated in prior:

$$\phi_{i+1/2,j} \cong \frac{\phi_{i,j} + \phi_{i+1,j}}{2} \quad (27a)$$

$$\phi_{i,j+1/2} \cong \frac{\phi_{i,j} + \phi_{i,j+1}}{2} \quad (27b)$$

The desired velocities, denoted by \bar{u} here, can be computed on different velocity nodes according to the level-set functions. The positive and negative values of a level-set function at a velocity node respectively denote the locations in solid and liquid phases. For a given velocity node, all possible situations are considered:

- (1) If the node is located in the solid region, the desired velocity is set as $\bar{u} = 0$.
- (2) If the node is located in the liquid phase while it has at least one neighbor nodes located in the solid phase, \bar{u} is interpolated from a neighboring liquid node by no-slip wall condition on the phase boundary (see Fig. 6). The interpolation method is the same with the one adopted in temperature interpolation, where level-set functions are used to estimate the distances needed in the calculation.

- (3) If the node and all its neighbors are located in the liquid phase, the additional force is not added, and \bar{u} does not need to be computed.

As the desired velocity is obtained, an additional force should be added on the node to correct the velocity. Without loss of generality, we discuss on the momentum equation in x -direction:

$$\rho \frac{\partial u}{\partial t} + \rho u \frac{\partial u}{\partial x} + \rho v \frac{\partial u}{\partial y} = -\frac{\partial p}{\partial x} + \eta \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) + f_x + f_{add,x} \quad (28)$$

in which $f_{add,x}$ refers to the artificially added force.

Using an FVM discretization, the momentum equation can be finally organized as a linear equation including the velocities of the node and its neighbors [46].

$$a_p^m u^m = \sum a_{nb}^m u_{nb}^m + b^m + b_{add}^m \quad (29)$$

The superscript m in Eq. (29) refers to the m th iteration in SIMPLER algorithm, and b_{add} corresponds to the term f_{add} in Eq. (28), which appears only in the nodes satisfying conditions (1) or (2) discussed above. u_{nb} and a_{nb} denote the velocities on neighboring nodes and the corresponding coefficients, respectively.

SIMPLE-family algorithms execute several inner iterations in each time step. During the iterations, the additional force is used to adjust the velocity until it is close enough to the desired value. In one iteration, therefore, no rigorous value of the adjusting force is required if only it takes a negative feedback effect on the velocity. Accordingly, in the m th iteration in SIMPLER, we add the additional force directly in the discretized equation as:

$$b_{add}^m = b_{add}^{m-1} + r(\bar{u}^{m-1} - u^{m-1}) \sum a_{nb}^m \quad (30)$$

The second term on the right-hand side is used to modify the discretized additional force during iterations. Here we followed two principles to establish this term: First, it should take a negative feedback effect on the velocity. Second, it should have the same dimension with the other terms in the discretized momentum equation. In the second right-hand term of Eq. (30), r is a relax parameter specified as 0.5 in the present study. During SIMPLER iterations, the discretized additional force can update itself and get approach to the value corresponding to the desired velocity. In the first iteration, the discretized term is given as:

$$b_{add}^1 = r(\bar{u}^0 - u^0) \sum a_{nb}^m, \quad (31)$$

in which the superscript 0 denotes either the velocity in the previous time step or the preset velocity specified in the first time step.

The procedure of the SIMPLER+IBM is summarized here. Starting from a given velocity field, the following steps are conducted in one iteration.

- (1) Compute the coefficients in the discretized momentum equation from the current velocities without considering the pressure gradient and artificial force for IBM;
- (2) Compute the desired velocity (u, v) for the nodes inside the solid or around the phase boundary;
- (3) Compute the discretized IBM force using Eq. (30) or 31 and add it to the discretized momentum equation;
- (4) Compute the pseudo velocities (\hat{u}, \hat{v}) from the discretized momentum equation without the pressure gradient term;
- (5) Solve the pressure equation according to the pseudo velocities obtained in Step (4), to obtain a pressure field p^* ;
- (6) Compute the pressure gradient term from p^* and add it to the discretized momentum equation, then solve the momentum equation to get a velocity field (u^*, v^*);
- (7) Solve the pressure correction equation according to (u^*, v^*) and get p' ;
- (8) Modify the velocity field (u^*, v^*) by the pressure correction p' , to obtain the velocity field updated in the current iteration.

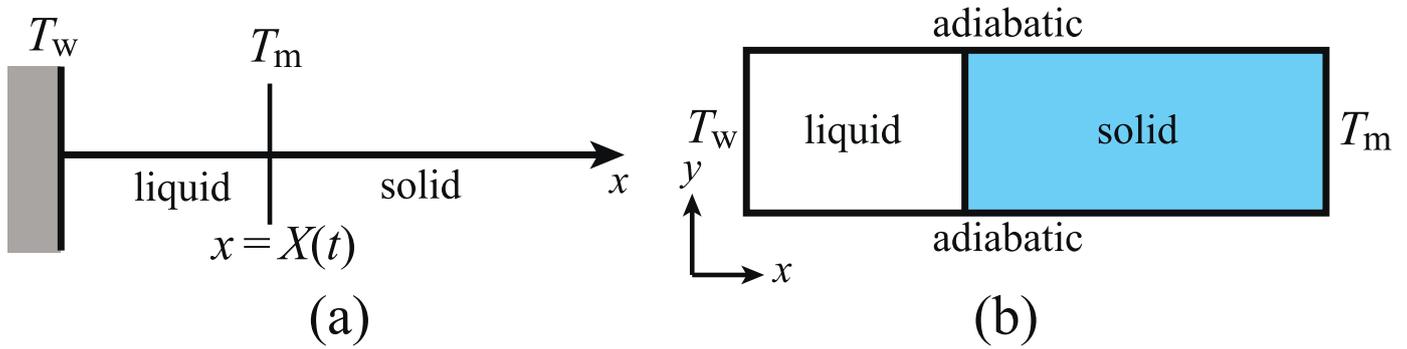


Fig. 7. Schematic view of Stefan problem: (a) description and (b) computational domain in two dimensions.

Note that, Steps (1), (4)~(8) are the sub-processes in SIMPLER algorithm [45]. Steps (2) and (3) are the IBM treatments for the effect of the solid phase. The accuracy examination of the proposed SIMPLER+IBM approach is presented in Appendix B.

4. Numerical examples

Some specific problems were studied to illustrate the application of the established numerical model for solid-liquid phase change, i.e., melting or solidification. The investigated problems include one-dimensional Stefan problem, two-dimension unstable solidification, melting with natural convection, and solidification in undercooled liquid without/with fluid flow.

4.1. One-dimensional Stefan problem

In order to assess the phase change model used in our study, a classical one dimensional Stefan problem was numerically studied. As illustrated by Fig. 7(a), a heating wall on the left increases the temperature in liquid, which drives the melting on the phase boundary. The temperature of the solid is initially given as the melting point, and the wall temperature, denoted by T_w , keeps constant.

Considering the involved heat conduction and melting, the governing equations and boundary conditions of this one-dimension Stefan problem can be expressed as:

$$\frac{\partial T}{\partial t} = \alpha_L \frac{\partial^2 T}{\partial x^2} \quad \text{for } 0 < x < X(t), \quad (32a)$$

$$T(0, t) = T_w, \quad (32b)$$

$$T(X(t), t) = T_m, \quad (32c)$$

$$\frac{dX}{dt} = \frac{k_L}{\rho L} \left(\frac{\partial T}{\partial x} \right)_{x=X}^- \quad (32d)$$

A mathematically analysis gives its exact solution describing the interface position and the temperature distribution in the liquid region [52]:

$$X(t) = 2\lambda(\alpha_L t)^{1/2} \quad (33)$$

and

$$T(x, t) = T_w - \frac{T_w - T_m}{\text{erf}\lambda} \text{erf} \frac{x}{2(\alpha_L t)^{1/2}} \quad \text{for } x < X(t), \quad (34)$$

where λ is the root of:

$$\lambda e^{\lambda^2} \text{erf}\lambda = \frac{c_{pL}(T_w - T_m)}{L\pi^{1/2}}. \quad (35)$$

The parameters used in this problem are $T_w = 2$, $T_m = 0$, $\rho = 1$, $k_L = 1$ and $c_{pL} = 1$. This one-dimensional problem can be numerically studied in a two-dimensional domain illustrated in Fig. 7(b) with adiabatic conditions specified on the two horizontal boundaries. A constant temperature of T_m was specified on the right boundary. The 2D domain covers the range from 0 to 1 in x -direction, and the initial interface position was set at $X = 0.2$. Correspondingly, the initial time t_0 , which the computation starts from, should be calculated by Eq. (33) from the initial interface position, and the initial temperature distribution was specified as $T(x, t_0)$ by Eq. (34).

Fig. 8(a) plots the interface positions with time under various grid resolutions, where the temperature gradients on the phase boundary were calculated by the linear approximation (Eq. (25)). It can be found that the numerical result converges to the exact one as the grid number increases. The same computations were then performed using quadratic approximation (Eq. (26)) for the calculation of the temperature gradients, and the results are plotted as Fig. 8(b). The comparison with Fig. 8(a) demonstrates a faster convergence rate in grid resolution. A good agreement with the exact solution was obtained even in the coarsest grid having only 10 cells in x direction.

Fig. 9 gives the temperature distributions at instances $0.05 + t_0$, $0.1 + t_0$, $0.2 + t_0$ and $0.3 + t_0$, obtained in the finest grid using quadratic approximation (Eq. (26)). The temperature distributions obtained by our numerical model fit well with the analytical solutions expressed by Eq. (34).

4.2. 2D unstable solidification

A solidification process in undercooled liquid designed by Juric and Tryggvason [1] was numerically investigated. The solidification occurs in a square domain of $(0, 4) \times (0, 4)$ from an initial solid seed placed at the center having four initial convex protrusions. The interface of the initial solid seed can be described by:

$$(x_f, y_f) = (2 + R \cos \theta, 2 + R \sin \theta), \quad (36)$$

where $R = 0.1 + 0.02 \cos(4\theta)$. The rest of the domain are initially filled with liquid phase with a constant temperature lower than the melting point.

This is a typical undercooled solidification, during which the interface may advance unstably and develop into dendritic shape [53]. It is due to the fact that a convex protrusion on the solid has more undercooled liquid surrounding it and therefore grows faster. On the crystal growth of a dendrite, it has been experimentally found by Glicksman et al. [54] that greater subcooling will result in faster growth rate and smaller tip radius. Via a linear stability analysis, Langer and Muller-Krumbhaar [17,18] established a criterion for tip-splitting instability.

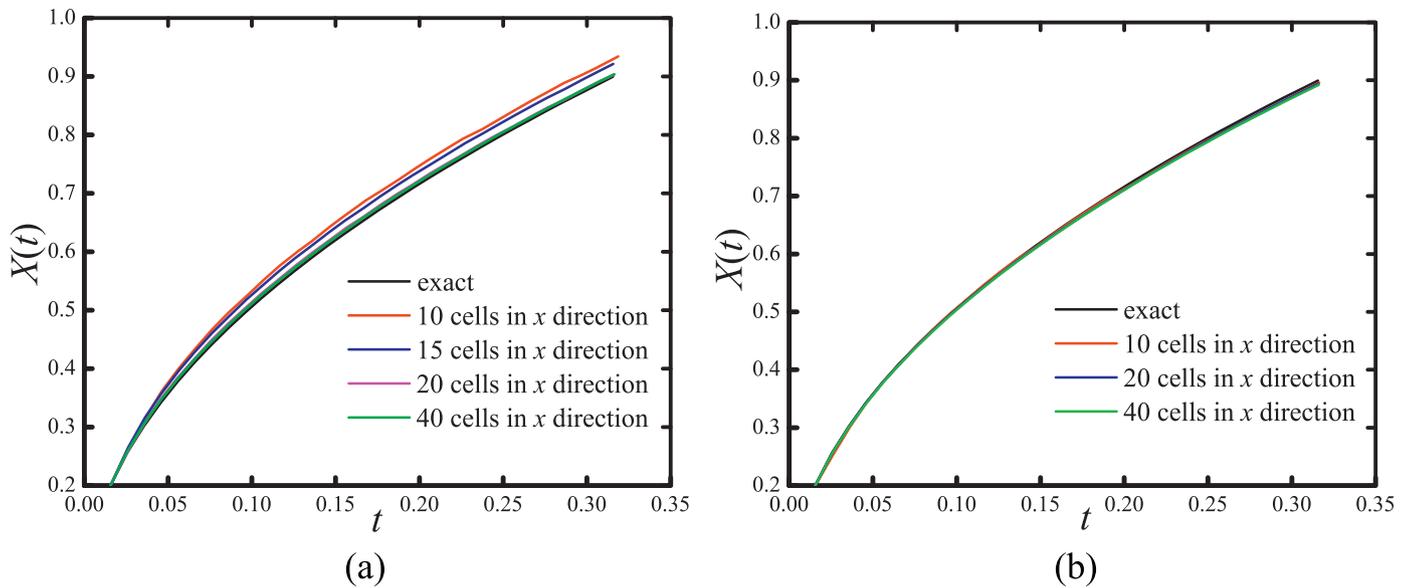


Fig. 8. Stefan problem: Interface position vs time with temperature gradients computed by: (a) linear approximation, (b) quadratic approximation.

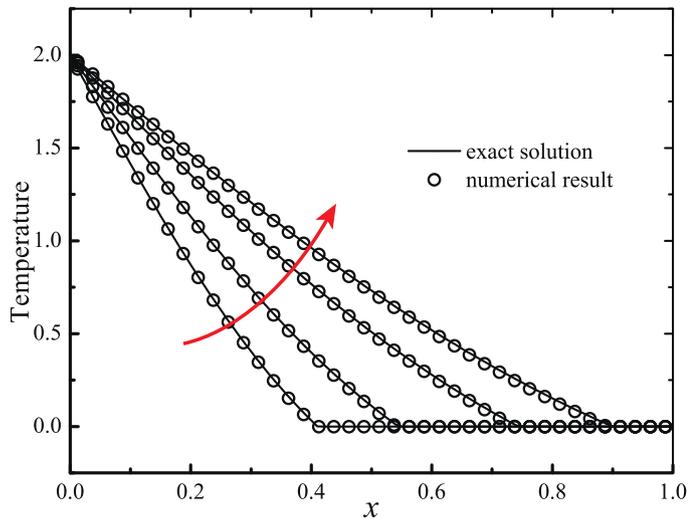


Fig. 9. Stefan problem: numerical and analytical results of temperature distribution.

4.2.1. Identical-density solidification

We first consider the case when the solid and liquid phases have the same density, which means no volume change occurs during solidification. Following the dimensionless properties used by Juric and Tryggvason [1], the solid and liquid phases were assumed to have the same density and thermal diffusivity, which were given as $\rho = 1$ and $\alpha = 1$, respectively. The parameters in the Gibbs–Thomson equation were specified as $T_m = 0$, $\epsilon_c = 0.002$ and $\epsilon = 0.002$. The initial temperatures of the liquid and solid phases were given as -0.5 and 0 , individually. The simulation for this problem was performed from time $t = 0$ to $t = 1$ under a set of grid resolutions of 200^2 , 400^2 and 600^2 . Both linear (Eq. (25)) and quadratic (Eq. (26)) approximations for the interfacial temperature gradients were used. The time step adopted for the transient simulation was chosen as $\Delta t = 0.05h/V_{\max}$, in which V_{\max} is the maximum interface moving velocity.

Fig. 10(a) shows the interface positions by the three grid resolutions at time $t = 1$, in which linear approximation was used for the calculation of the temperature gradients on the interface. All the results show a dendritic shape formed in the undercooled solidifi-

cation. Through comparison, we can find the interface approaches a certain shape with progressively finer mesh, but the grid convergence appears not satisfying. Fig. 10(b) displays the interfaces resulted by the same grid resolutions with quadratic approximation employed for the interfacial temperature gradients. In this figure we can see the interfaces by the two finer mesh are almost coincident, indicating a much better performance in grid convergence. It suggests the quadratic approximation gives more accurate results in solid-liquid phase change simulation. Therefore, for the simulations on the rest problems, we used quadratic approximation only for the temperature gradients on the phase boundary.

The solidification process (400^2 grid, quadratic approximation) is displayed as Fig. 11 with a time increment of 0.1 . We can see the four initial convex protrusions developed into dendrites, which then started to split at time around $t = 0.3$. Additionally, the solidification proceeded progressively slower. Through comparison, our result was found to be in good accordance with those reported by Udaykumar and Mao [3], Zhao and Heinrich [6], and López et al. [21].

Fig. 12 gives the temperature fields at instances of $t = 0.2$ and $t = 0.6$, in which we can see that the heat released from the solidification resulted in higher temperature in the liquid phase. It decreased the undercooling degree of the liquid around the interface, which further led to progressively decreasing rate of the solid growth. Meanwhile, we can see the temperature in the solid phase are non-uniform. It is lies in the unevenness of the interface temperature under the effect of surface energy and kinetic mobility.

From Fig. 11 we can find an evident anisotropy feature in the solid growing process. Concretely, the solid dendrites grew in the directions of the initial convex protrusions at the early stage, which are precisely the directions of the computational cell faces, i.e., x - and y - directions. Therefore, it remains possible that the grid anisotropy effect resulted in the growing feature presented in Fig. 11. In order to assess the effect of grid anisotropy, we then conducted the same simulations with the initial solid seed counter-clockwise rotated 30° and 45° . The simulations for the cases with rotated solid seeds were terminated at time $t = 0.2$ such that the insulated boundaries would not make significant differences. We specified the end time based on the result shown in Fig. 12(a), where the temperature at the boundaries almost stayed at the initial temperature of -0.5 . Fig. 13(a) and (b) present the

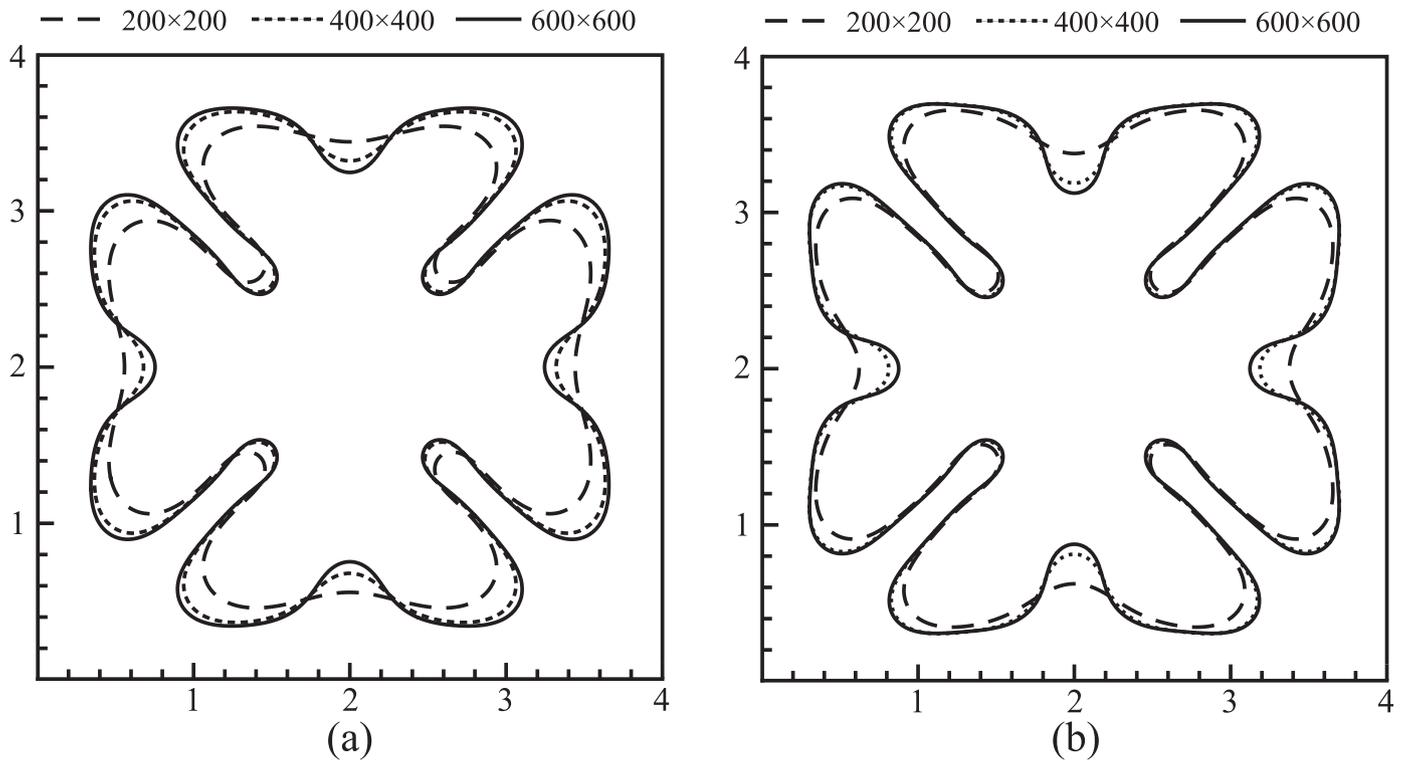


Fig. 10. The final solid-liquid interfaces of 2D unsteady solidification problem obtained in different grids. Different schemes were used in calculating the temperature gradients on the phase boundary: (a) Linear approximation; (b) Quadratic approximation.

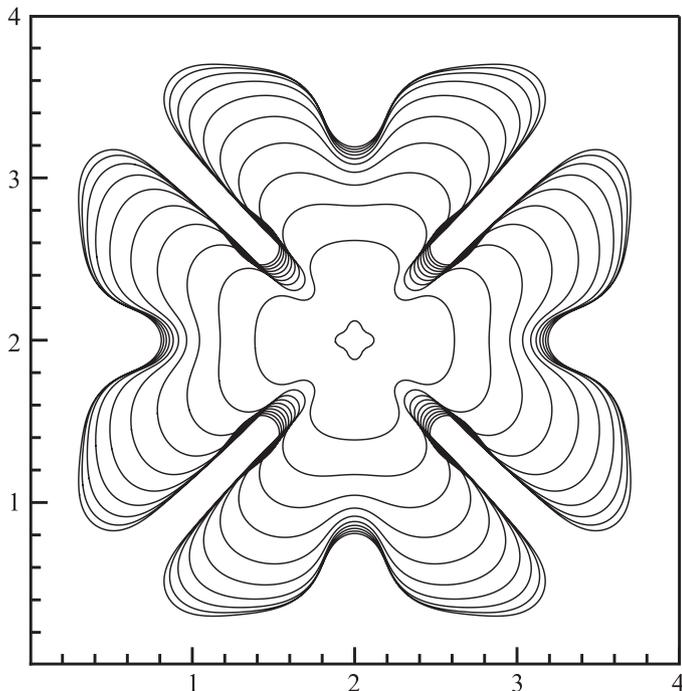


Fig. 11. Unstable solidification: interface evolution from time $t = 0$ to $t = 1$.

interface evolutions with a time increment of 0.1 in the cases with the solid seeds rotated 30° and 45° , respectively. It can be seen that the solid was growing in the directions of the initial convex protrusions regardless of the computational grid directions. Furthermore, the sizes as well as the shapes of the solid phase at $t = 0.2$ almost kept the same with the original case. For a clearer comparison, the interfaces of the cases with rotated solid seeds

were rotated back and put together in Fig. 13(c) with the original case without rotation. The interfaces by rotated solid seed are almost coincident with the one without rotation, which suggests our numerical methods have little effect in grid anisotropy. It also confirms that the shape of initial solid seed has a significant effect on the growth directions in undercooled solidification.

4.2.2. Solidification with volume expansion

The result presented in this sub-section is to illustrate the ability of the proposed numerical approaches in dealing with volume change during phase change. Based on the 2D unstable solidification problem designed by Juric and Tryggvason [1], we considered the case when density of the solid phase is smaller than that of the liquid phase. The density inequality would result in volume expansion in the process of solidification, and therefore, we solved the flow field in the liquid phase. Outflow conditions were specified on the four boundaries such that the global mass conservation could be guaranteed. Concretely, two more cases with $\rho_S/\rho_L = 0.9$ and $\rho_S/\rho_L = 0.8$ were numerically studied with the grid resolution of 400×400 .

Considering the volume variation, one needs to pay some special attentions when solving the flow field with the proposed SIMPLER+IBM. First, the liquid velocity on the interface is not zero and should be calculated by Eq. (9). It needs to be taken into account when conducting linear interpolation for liquid velocities around the interface. Second, since the velocity field is solved in the entire domain using immersed boundary method, the velocity is not divergence-free for computational cells containing interfaces. Instead, the integral-form continuity equation need to be written as:

$$\int_{CV} \nabla \cdot \mathbf{u} dV = \int_{inter} \left(1 - \frac{\rho_S}{\rho_L}\right) v_n dA \quad (37)$$

in which the subscripts *CV* and *inter* individually represent a control volume and the solid-liquid interface it contains. Indeed, the right-hand side of Eq. (37) is non-zero only in interfacial cells.

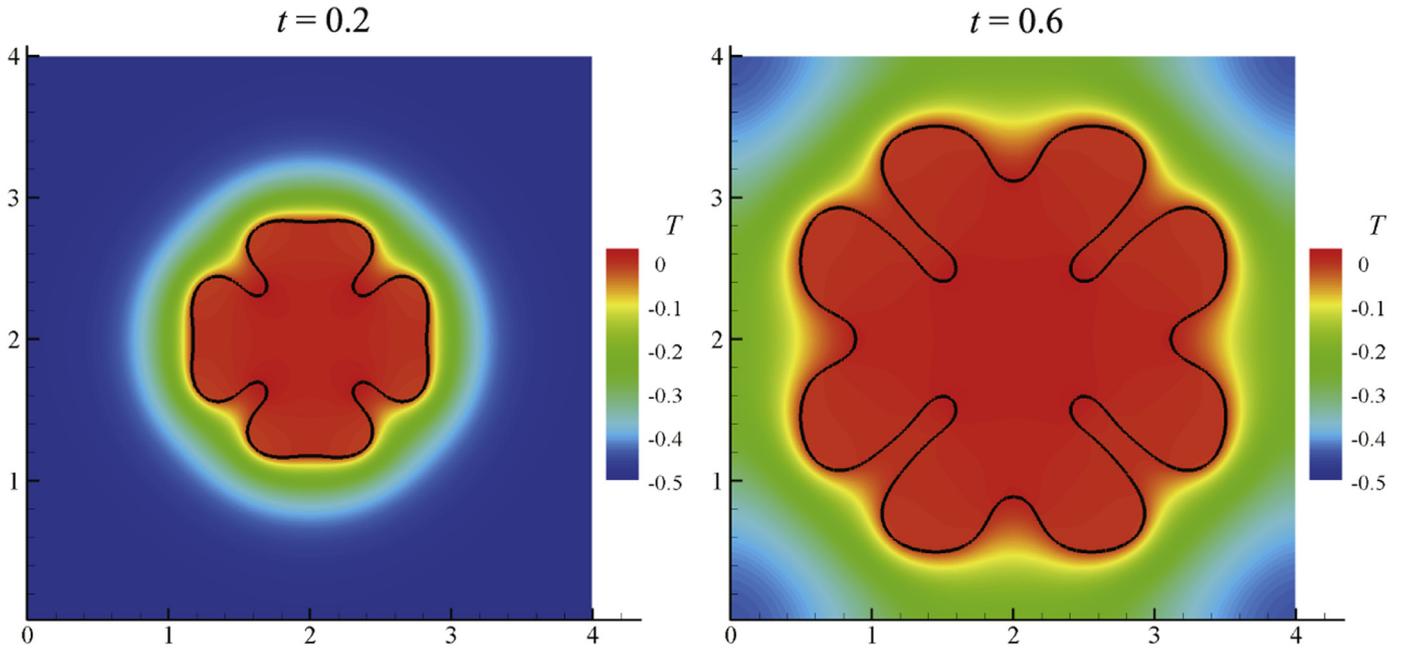


Fig. 12. 2D unstable solidification: evolution of the temperature field.

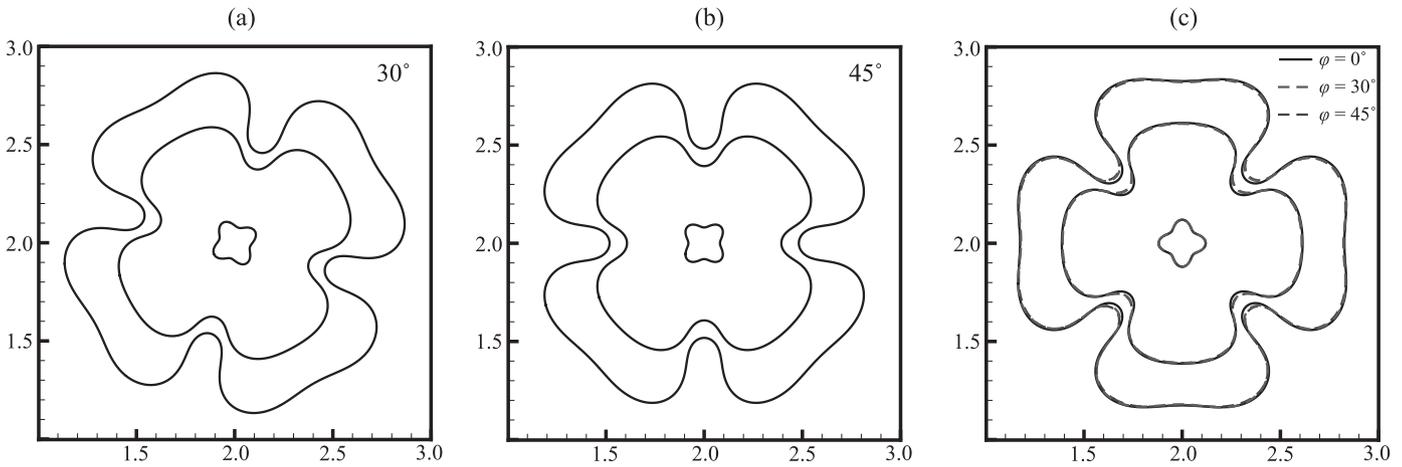


Fig. 13. Unstable solidification with the initial seed rotated: (a) Interface evolution with initial seed 30° rotated; (b) Interface evolution with initial seed 45° rotated; (c) Comparison between the interfaces at $t = 0.2$ under various rotation angles of the initial solid seed.

Fig. 14(a) shows the solid-liquid interface together with the velocity field for the case $\rho_S/\rho_L = 0.9$ at time $t = 0.2$. At the liquid side on the phase boundary, we can see the liquid flows in the direction of the solid growth, while the solid phase keeps stationary. Fig. 14 (b) plots the final interface positions at density ratios (ρ_S/ρ_L) of 0.8, 0.9 and 1.0 (identical density). Through a careful comparison one can find the solidification is in some degree restrained by the volume expansion. It lies in the outward-pointing velocity field, which drives cold liquid away from the phase boundary and therefore decrease the phase change rate. However, the effect by the density inequality appears not significant in the range investigated.

In the remaining numerical examples, we assume equal density for the two phases and neglect the volume expansion/shrink during phase change.

4.3. Melting with natural convection

Bertrand et al. [55] described a benchmark problem involving melting and natural convection. As illustrated in Fig. 15, a pure

solid material initially fills a square cavity at its melting point. The temperature on the left wall is raised to a relatively high value ($T = T_w > T_m$) at the and then keeps constant. The temperature on the right wall stays at the melting point, and the two horizontal walls are adiabatic. The solid material around the wall begins to melt under the higher temperature, dividing the square domain into solid and liquid regions. Simultaneously, natural convection forms in the liquid phase, which will then result in faster melting rate in the upper region of the cavity.

This problem can be described by three dimensionless numbers, namely Prandtl number (Pr), Stefan number (Ste) and Rayleigh number (Ra) individually defined as:

$$Pr = \frac{\nu_L}{\alpha_L}, \quad Ste = \frac{(T_w - T_m)C_{pL}}{L}, \quad Ra = \frac{g\beta H^3}{\nu_L \alpha_L} \quad (38)$$

in which β is the coefficient of thermal expansion, g is the gravity acceleration, and H denotes the side length of the square.

Boussinesq approximation was adopted in our simulation for the buoyancy force. Thus the volume force appearing in the mo-

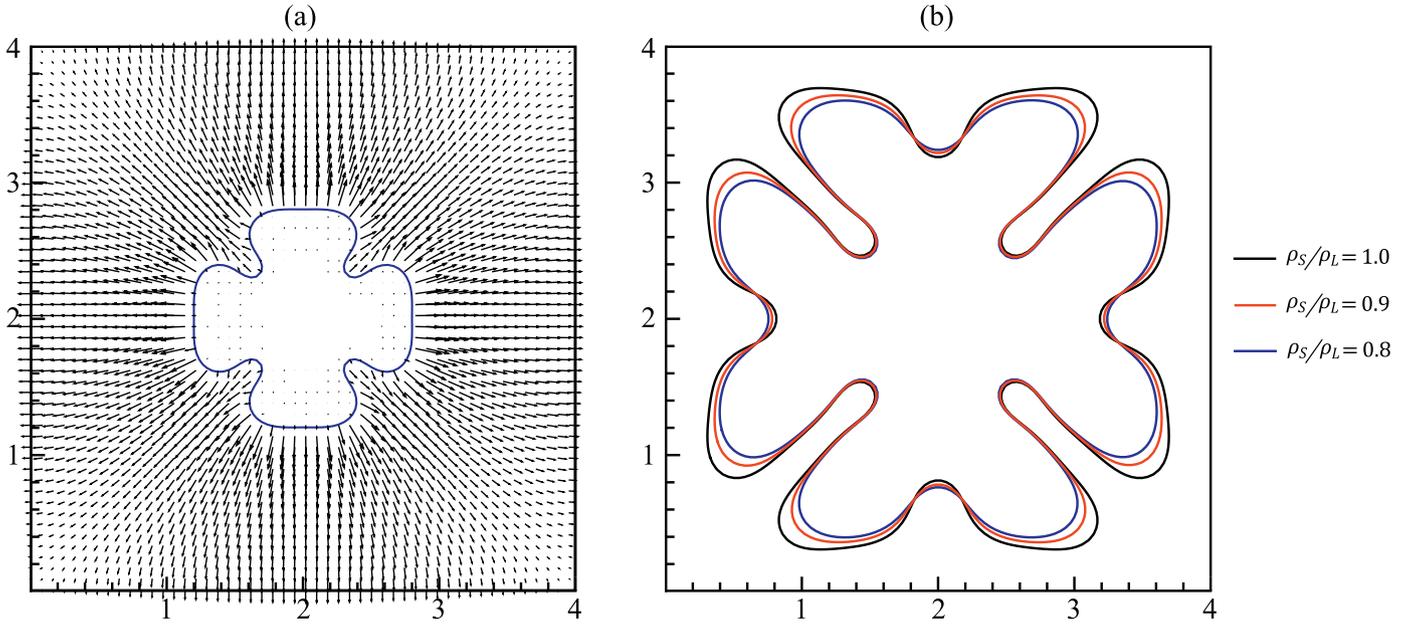


Fig. 14. 2D unstable solidification with unequal densities of solid and liquid phases: (a) Phase boundary and velocity field of the case with $\rho_s/\rho_l = 0.9$ at $t = 0.2$; (b) Comparison of phase boundaries under different density ratios.

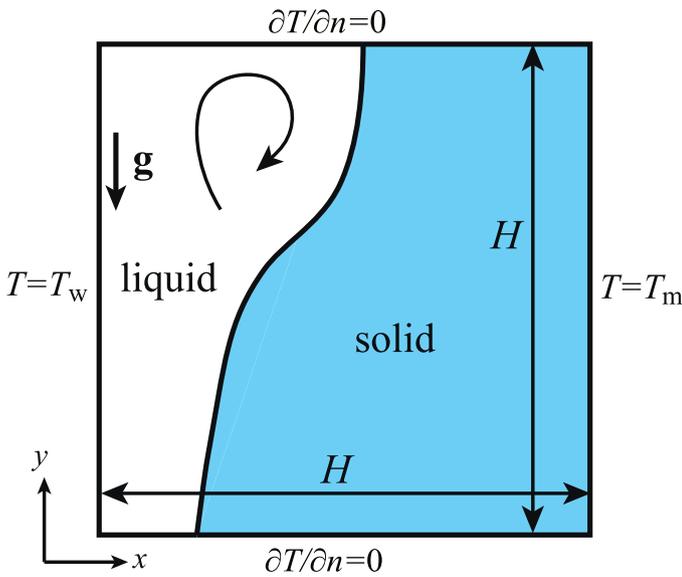


Fig. 15. Schematic of melting problem with natural convection.

mentum equation (Eq. (8)) should be expressed as:

$$\mathbf{f} = -\rho\beta(T - T_m)\mathbf{g}. \quad (39)$$

A comprehensive numerical investigation was performed by Mencinger [28] under various combinations of dimensionless numbers, in which an enthalpy method was applied in an adaptive moving grid. It was then studied by Huang et al. [10,11] using a lattice Boltzmann method. Here we used the numerical model presented in Section 3 for the simulation on a case with dimensionless numbers $Pr = 0.02$, $Ste = 0.01$, $Ra = 2.5 \times 10^{-4}$. In the present study, the square cavity was divided by grids of 40×40 , 64×64 , 100×100 and 160×160 for the resolution test, and the result will be compared with those in previous studies. The detailed parameters adopted in our simulation are summarized in Table 1.

The dimensionless time, namely Fourier number was defined as $Fo = \alpha_L t/H^2$. The simulations covered the period ranging from

Table 1

Parameters used in simulating melting with natural convection.

| Parameter | Value | Unit |
|-------------|------------------------------|--------------------|
| k_L | 60 | W/m·K |
| c_{pL} | 200 | J/kg·K |
| ρ | 7500 | kg/m ³ |
| L | 6.0×10^4 | J/kg |
| η | 6.0×10^{-3} | N·s/m ² |
| β | $\frac{8}{3} \times 10^{-4}$ | 1/K |
| H | 0.1 | m |
| $T_w - T_m$ | 3.0 | K |
| g | 1.0 | m ² /s |

$Fo = 0$ to $Fo = 30$. Fig. 16 (a) compares the phase boundaries at $Fo = 20$ obtained in the four grid resolutions. We can see the resulted interfaces by various grid resolutions are almost coincident except for the one by the coarsest grid. The interface positions at some instances obtained by the 100×100 uniform grid are plotted in Fig. 16 (b), in which the results by Mencinger [28] and by Huang and Wu [11] are included as well. Through comparison we can see that the interface positions obtained by our numerical methods confirm closely with those reported in literature. It should be noted that, by our model, the same result can be reached in a coarser grid. The interface by Huang and Wu [11] are obtained in a 128×128 grid; and the interface by Mencinger et al. [28] was achieved by an adaptive grid which dynamically generates much finer mesh around the phase boundary.

Fig. 17 gives the interface positions, temperature and velocity fields at three instances at $Fo = 4.0$, $Fo = 10.0$ and $Fo = 20.0$. It can be seen, velocity vectors appear only in the liquid phase. Meanwhile, the interface evolution shows an evident impact of the natural convection on the melting. The temperature contours and the phase boundary at $Fo = 4.0$ are almost parallel to the heating wall, indicating that the conduction played as the dominant role in heat transfer at the early stage. The natural convection was then enhanced as the liquid region continued expanding. In the liquid phase, some vortexes were formed under the effect of the temperature difference, which progressively merged into a larger one

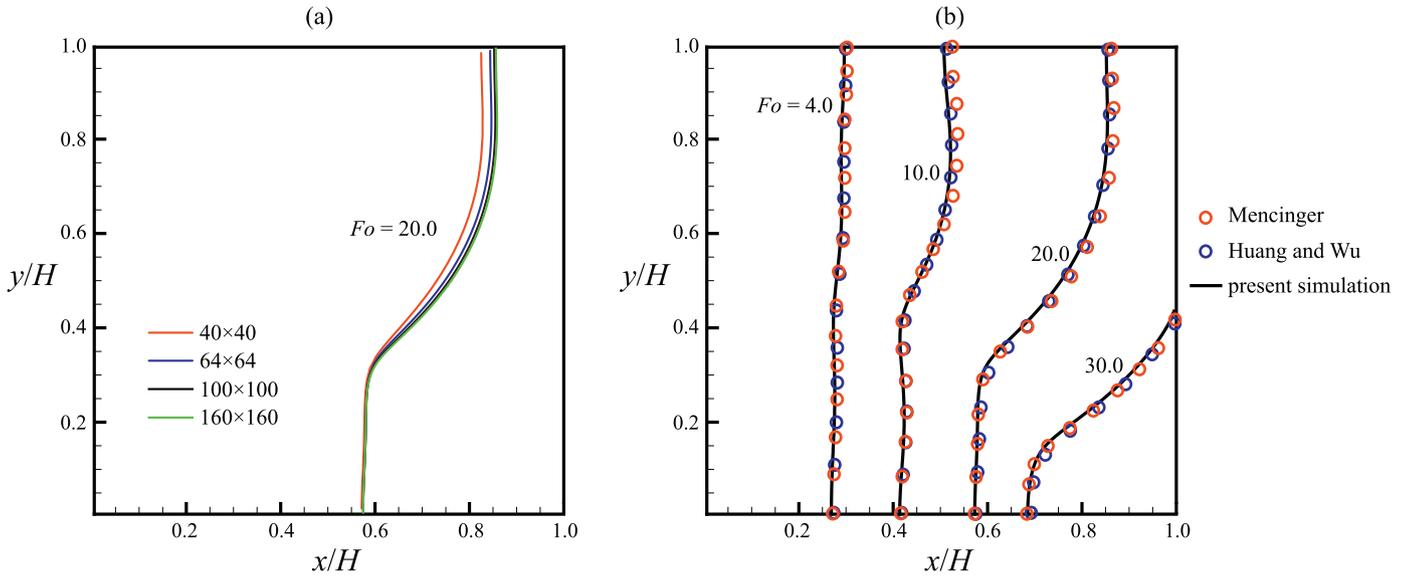


Fig. 16. Phase boundary of melting with natural convection: (a) Grid independent test; (b) Comparison of phase boundary with previous studies.

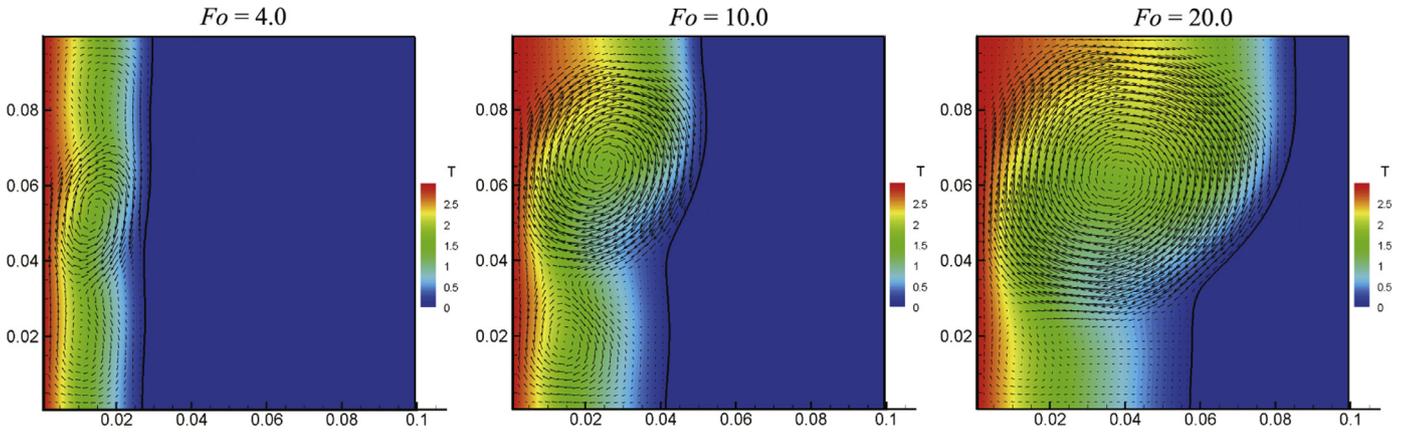


Fig. 17. Melting with natural convection: phase boundaries, velocity and temperature fields at Fo numbers of 4.0, 10.0 and 20.0.

(see Fig. 17 at $Fo = 20.0$). During this period, hot liquid continuously flowed upwards while colder liquid flowed downwards, which resulted in the faster melting rate at the upper part of the cavity.

The average Nusselt number, which is defined by Eq. (40), versus Fourier number was plotted in Fig. 18, in which the results by Mencinger [28] and Huang et al. [11] are included. The Nusselt number obtained in our simulation shows an excellent agreement with those in the two previous studies. The high Nusselt number at the early stage lies in the initial thin liquid film between the phase boundary and the heating wall. It dropped rapidly with the increase of the liquid thickness. No significant variation can be found on the Nusselt number when $Fo > 10$ even though the liquid region continued expanding. It suggests that the natural convection was producing an effect increasing the heat transfer during this period.

$$Nu = \frac{\int_{wall} q dy}{k_L(T_w - T_m)} = \frac{\int_{wall} \frac{\partial T}{\partial x} dy}{T_w - T_m} \quad (40)$$

4.4. Crystal growth with forced convection

The numerical results in Section 4.4 demonstrated that the solid-liquid interface may develop into dendrites during solidification into undercooled liquid. In some previous studies [4,5,15,27], the fluid flow in the liquid phase was found to have significant

Table 2
Properties of metal tin.

| | Solid | Liquid |
|---------------------------|--------------|--|
| Thermal conductivity | 61 W/m · K | 34 W/m · K |
| Specific heat | 228 J/kg · K | 255 J/kg · K |
| Viscosity | | 1.93×10^{-3} N · s/m ² |
| Average density | | 7145 kg/m ³ |
| Melting point | | 505 K |
| Latent heat | | 60300 J/K |
| Gibbs–Thomson coefficient | | 5.39×10^{-7} m · K |

impact on the dendritic shape formed during solidification. This characteristic will result in different micro structures of a material. Here we designed a problem of solidification into undercooled melt with liquid flowing over a solid layer. As Fig. 19 shows, the solid grows at the bottom with new undercooled liquid flowing in from a part of the top boundary and flowing out from the right boundary. The concerned region is 400µm in width and 200µm in height. In the present problem, we used properties of metal tin, which were summarized in Table 2.

The initial temperature of the solid phase was given as the melting point, and the initial liquid temperature was given as 455 K, i.e., 50 K lower than the melting point except for a 10µm thermal layer around the interface, in which the temperature linearly varies from 455 K to the melting point. The temperature at

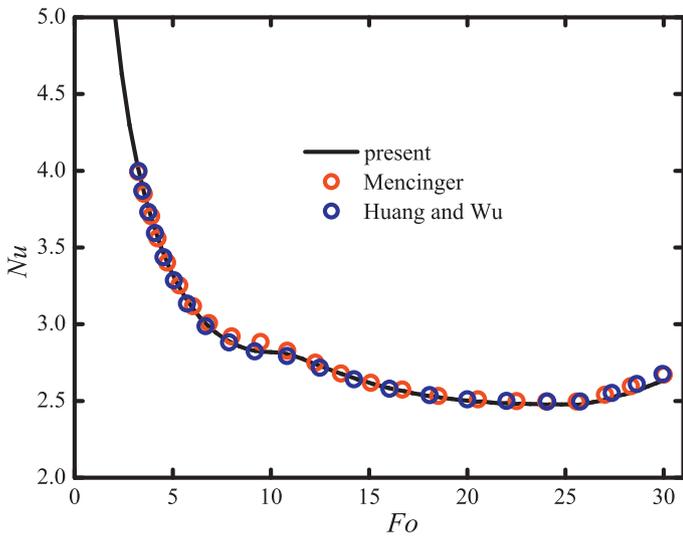


Fig. 18. Melting with natural convection: Comparison of Nusselt number with previous studies.

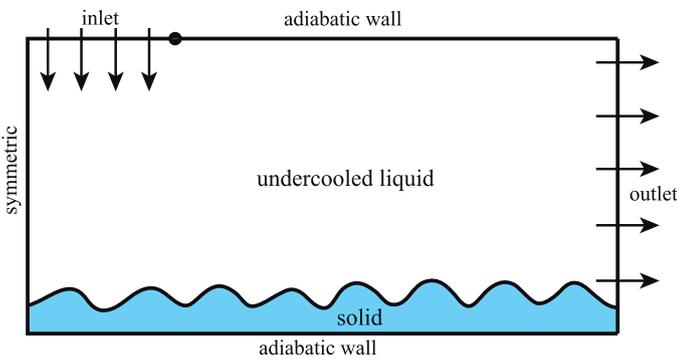


Fig. 19. Schematic of crystal growth problem with forced convection.

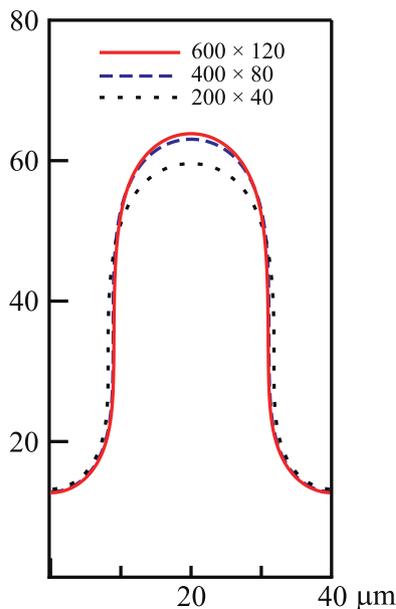


Fig. 20. Crystal growth with forced convection: test on grid resolution.

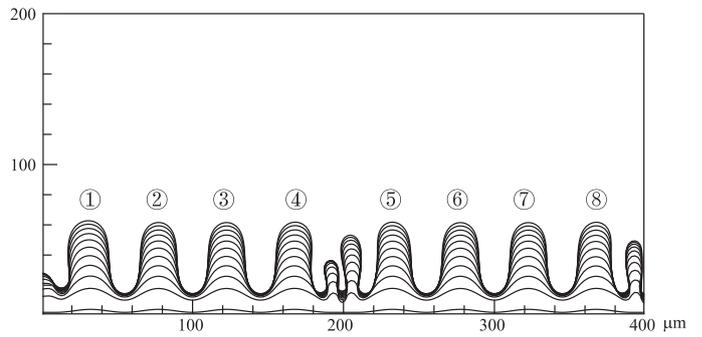


Fig. 21. 2D crystal growth with convection: interface evolution without flow.

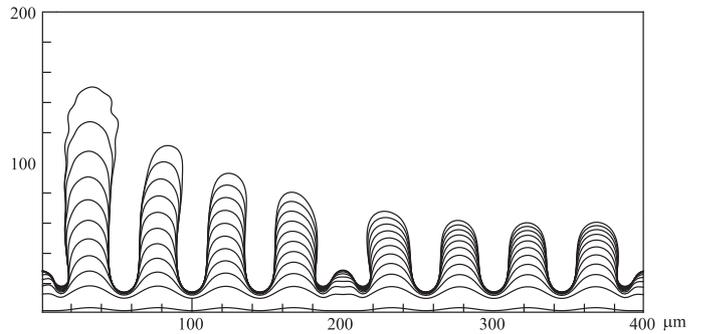


Fig. 22. 2D crystal growth with convection: interface evolution with inlet velocity of 0.5 m/s.

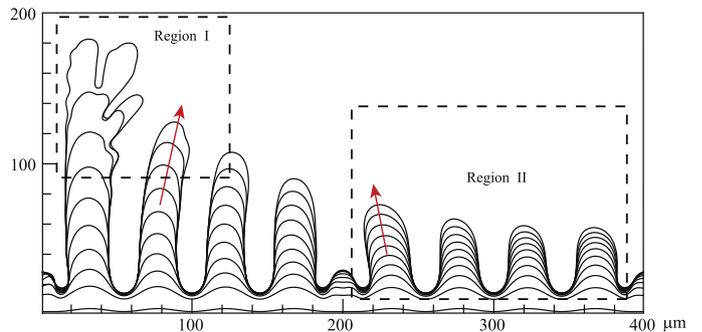


Fig. 23. 2D crystal growth with convection: interface evolution with inlet velocity of 0.8 m/s.

the inlet was specified 455 K as well. Natural convection was not taken into account owing to the small scale of the concerned domain. In order to trigger the unstable solidification, a wave-shape interface was given for the initial solid layer covering the bottom boundary. Eight convex protrusions having a same height of 2μm were unevenly distributed. The solidification processes within 1 ms under different inflow velocities were numerically investigated.

In order to optimize the grid size for the simulation, we first conducted a computation in a smaller domain having a width of 50μm without considering the fluid flow. Only one convex protrusion was given on the initial solid layer in this region. The test problem was simulated by grid resolutions of 40 × 200, 80 × 400 and 120 × 600, and the obtained solid-liquid interfaces are compared in Fig. 20. It can be seen that the results by the two finer meshes are in good accordance. We therefore considered the intermediate grid is sufficient for the grid-independence requirement, and the corresponding grid size, 0.5μm, was used in the simulations in the entire domain.

The simulations covered the processes from 0 to 1 ms under three cases where the inlet velocities were zero (no fluid flow),

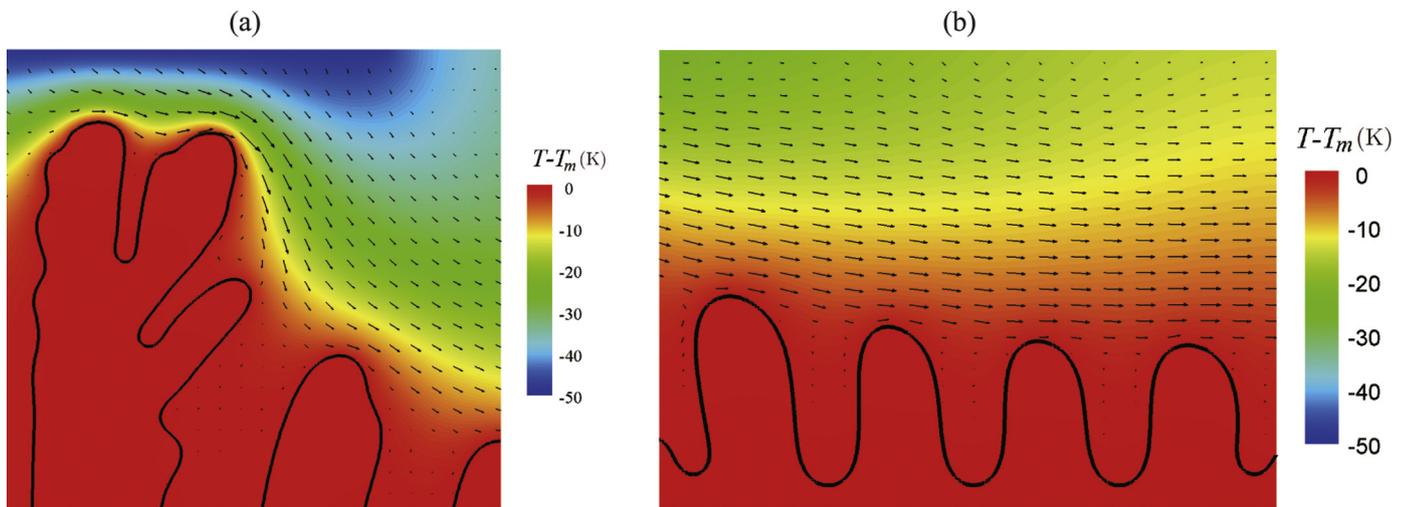


Fig. 24. Velocity and temperature fields with inlet velocity of 0.8 m/s: (a) in the local region I, and (b) in the local region II.

0.5 m/s and 0.8 m/s. We first carried out the simulation without considering the fluid flow with insulated condition specified on the entire top boundary. The interface evolution resulted from our simulation is displayed as Fig. 21, where we can find a unstable feature on the interface advancement. The convex parts on the initial solid were growing evidently faster than concave ones. Particularly, the growths on the concave parts of the solid boundary were almost halted at 0.1 ms. Consequently, the eight initial convex protrusions resulted in eight main finger-shape crystal arms. The result shows a similar feature with the problem presented in Section 4.4. Furthermore, we can see two additional smaller crystal arms formed between the 4th and the 5th main crystal arms. It is owing to a larger distance between the two protrusions, which provides enough space for new dendrites to form.

Figs. 22 and 23 display the interface evolutions when the inlet velocities were given as 0.5 m/s and 0.8 m/s, respectively. It should be noted that for the case with the inlet velocity of 0.8 m/s the simulation was terminated at 0.9 ms when the solid almost reached the top boundary. Through comparison with Fig. 21, we can see the liquid flow brought out evident influences on the sizes as well as the shapes of the crystal arms. It is probably the most obvious effect that the fluid flow was accelerating the growth rates of the crystal arms around the inlet. It lies in the new undercooled liquid supplemented from the inlet, which provides an advantageous situation for solidification with lower liquid temperature. Particularly, for the case with inlet velocity of 0.8 m/s (see Fig. 23), secondary dendrites were formed on the 1st crystal arm in the later period simulated.

Furthermore, we can see the growth directions of the crystal arms were inclined with liquid flowing over them. This feature is more significant in the case with the inlet velocity of 0.8 m/s. As Fig. 23 shows, some crystal arms (such as the 2nd and the 3rd arms) showed right-inclining feature in growth direction, while others (such as the 5th to the 8th arms) grew in left-inclining directions. A crystal arm prefers to grow to the colder side due to the greater phase change rate. However, the right- and left-inclining growth features lies in different heat transfer mechanisms. Fig. 24(a) shows the local fields of temperature and velocity at 0.9 ms in region I marked in Fig. 23 with the tips of the 1st and the 2nd crystal arms included. It can be seen that the tip of the 1st crystal arm was surrounded by the coldest liquid flowing from the inlet. It resulted in the fastest growth rate of the dendrite. More importantly, the locally large subcooling of the liquid increased the instability of the solid-liquid interface, which leads

to tip-splitting and forms secondary dendrites. The faster growth rate of the 1st crystal arm, however, released more heat on the left side of the 2nd crystal arm. Under this effect, the tip of the 2nd crystal arm grew faster at its right side where the liquid temperature was relatively lower, thus forming a right-inclining growth direction.

Through a careful comparison among the 5th to the 8th crystal arms, we can find the growth rate is decreasing successively from upstream to downstream. The temperature and velocity fields in the local region II are displayed as Fig. 24(b). It shows that the temperature of the liquid above these crystal arms is increasing along the flow direction. Here we can see another impact the liquid flow made on the solidification. The heat released from the upstream crystal arms was transported downstream and restrained the growths of the crystal arms there. It also explains the reason for the inclining of the growth direction of the 5th crystal arm. Since the 5th crystal tip had colder liquid in the upstream direction located at its left than in the downstream direction located at its right, the crystal arm grew in a left-inclining direction where the liquid temperature was lower.

5. Conclusion

A sharp-interface model combining VOSET and IBM methods was built up for the simulation on solid-liquid phase change. A VOF-based interface tracking method was developed to deal with solid growth/reduction occurring in solidification/melting. Level-set function was applied for the calculation of interface normal and curvature, and for the interpolation of temperature and velocity around the phase boundary. In order to handle the interaction between the solid and liquid phases, an immersed boundary method was incorporated into SIMPLER algorithm, in which the artificial forces are added directly in the discretized momentum equation.

After being validated by a set of test problems, the fundamental numerical approaches were combined to simulate both melting and solidification processes. The simulation on the solidification into undercooled liquid resulted in a dendritic shape, and the resulted phase boundary of the identical-density case is quantitatively consistent with previous studies. The result on the unequal-density case shows a slight influence by the volume expansion in reducing the rate of phase change. Via this problem, our computational methods were illustrated to be free of grid anisotropy. The comparisons on the two schemes in normal probe technique indicated that quadratic approximation is prior to linear approximation in convergence of grid resolution. The results of the melting

problem show evident influences of the natural convection to the melting rate as well as to the heat transfer characteristic. Without the use of local mesh refinement [28], our approaches achieved the result in the same level of accuracy by a mesh size twice of the one used by a front-tracking method [11]. Finally, a crystal growth with forced convection was investigated, and the results suggested remarkable impacts of the fluid flow on the shapes of the dendrites formed during solidification.

The proposed VOSET-IBM method can provide a much simpler way than front tracking method in implementation, and owing to a sharp description on the phase boundary, it gives preferable accuracy to diffused models like enthalpy method. This model has been successfully applied for both melting and solidification problems with the effects by fluid flow and density difference between phases taken into consideration.

Acknowledgements

This work is supported by China Postdoctoral Science Foundation (2018M6333506), the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No.51721004), and the 111 Project (B16038). Kong Ling would like to appreciate Dr. Yongliang Feng for his helpful suggestions.

Appendix A. Validation of the proposed interface tracking method

In order to assess the interface tracking method presented in Section 3.2, we studied some standard problems in which interfaces are tracked with given velocity fields. The investigated problems are list as follows:

- (1) *Translation test*: In a $(0, 1) \times (0, 1)$ square, a circle with a radius of 0.15 initially centered at $(0.25, 0.25)$ shifts under a steady and uniform velocity of $\mathbf{u} = (1, 1)$. After a period of 0.5, the translated circle should be centered at $(0.75, 0.75)$ maintaining its original size and shape.
- (2) *Zalesak rotation test* [50]: In a $(0, 1) \times (0, 1)$ square, a slotted circle initially centered at $(0.5, 0.75)$ shifts under a steady uniform-vorticity field $\mathbf{u} = (-y + 0.5, x - 0.5)$. During a period of $T = 2\pi$, the slotted circle should be rotated back to its original position.
- (3) *Vortex-in-a-box test*: In a $(0, 1) \times (0, 1)$ square, a circle having a radius of 0.15 is initially centered at $(0.5, 0.75)$, and the velocity field is given by a stream function $\psi(t) = \frac{1}{\pi} \sin^2(\pi x) \sin^2(\pi y) \cos(\pi t/T)$, where T was specified as 6.0 in the present study. Under this velocity field, the initial circle will experience a deformation process during $0 < t < T/2$ and then recovered during $T/2 < t < T$. At time $t = T$ when the interface finished a whole period, it should return to its initial position.

In order to estimate the derivation of the tracked interface from the accurate one, we calculated an L_1 -norm error which is defined as:

$$\text{Error} = \sum |C_{i,j} - C_{i,j}^{\text{exact}}| \Delta V_{i,j}, \quad (41)$$

where the superscript *exact* refers to the volume fractions corresponding to the exact result for a specific problem, and $\Delta V_{i,j}$ is the area of cell (i, j) . Those problems were simulated under different mesh sizes, and the L_1 -norm errors will be compared to estimate the rate of convergence.

On those three standard test problems, velocity fields were given directly on the cell vertices. Actual simulations shows that accurate results requires the use of smaller time steps, especially when the interface experience a large deformation in the vortex-in-a-box problem. We adopted a CFL number of about 0.1 in the translation and rotation tests, and 0.025 for the vortex-in-a-box

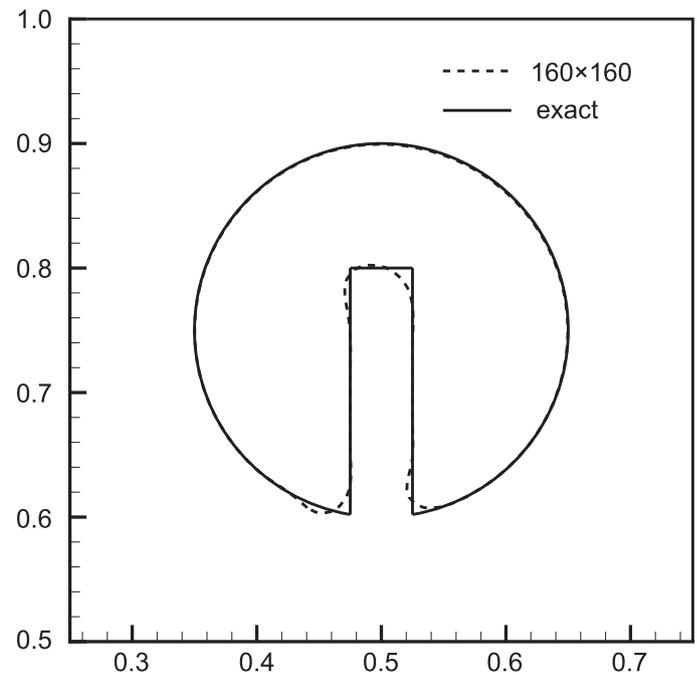


Fig. 25. Interface position of slotted disk after an entire rotation period.

problem. The L_1 -norm errors as well as the estimated convergence rate were summarized as Table 3. As can be seen, the numerical error will be decreased by using finer grid in all the three problems. The rate of convergence of our interface tracking method was estimated to be slightly lower than 2.0.

Fig. 25 gives the interface at $t = T$ of the rotation test by the 160×160 grid, in which the exact interface exact solution is included for comparison. It can be observed that the derivation appears mostly around the sharp corners on the interface. Fig. 26 shows the interface positions of the vortex-in-a-box problem tracked by the 160×160 grid. At $t = T/2$ when the deformation process was completed, the interface was stretched into a thin stripe without being split up. The interface at $t = T$ is given as the right figure along with the exact circular interface, from which we can see the interface was almost recovered to its initial position.

Appendix B. Validation of SIMPLER+IBM

A numerical study was conducted on a lid-driven flow in a semicircular cavity to assess the SIMPLER+IBM method presented in Section 3.4. As sketched in Fig. 27, the fluid takes a semicircular domain with a radius of 0.5 and the flow is driven by the top wall moving at a given velocity. By the use of triangular cells, Glowinski et al. [51] performed a comprehensive numerical investigation on this problem with Reynolds numbers ranging from 500 to 6600, providing a set of benchmark solutions. We carried out a numerical simulation on the case with $Re = 1000$ using a 200×100 Cartesian grid. A level-set function was given as:

$$\phi = 0.5 - \sqrt{(x - 0.5)^2 + (y - 0.5)^2} \quad (42)$$

to identify the fluid and solid regions.

The velocities at the central lines of the computational domain, i.e., $u(0.5, y)$ and $v(x, 0.25)$, are plotted in Fig. 28, where the result by Glowinski et al. [51] is included for comparison. Apparently, the velocity distributions obtained by our SIMPLER+IBM method confirm closely with that reported by Glowinski et al. Furthermore, the computation was conducted using two relax parameters appearing in Eq. (30), i.e., $r = 0.5$ and $r = 0.3$. However, the resulting difference can be hardly detected in Fig. 28, which suggests the

Table 3
Numerical error and rate of convergence of test problems for interface tracking methods.

| Grid | Translation | | Zalesak rotation | | Vortex-in-a-box | |
|-----------|-----------------------|-------|-----------------------|-------|------------------------|-------|
| | Error | R.O.C | Error | R.O.C | Error | R.O.C |
| 40 × 40 | 6.47×10^{-4} | – | 1.21×10^{-2} | – | 2.59×10^{-2} | – |
| 80 × 80 | 2.23×10^{-4} | 1.59 | 3.03×10^{-3} | 2.00 | 5.28×10^{-3} | 2.29 |
| 160 × 160 | 5.38×10^{-5} | 2.05 | 1.08×10^{-3} | 1.49 | 1.725×10^{-3} | 1.61 |

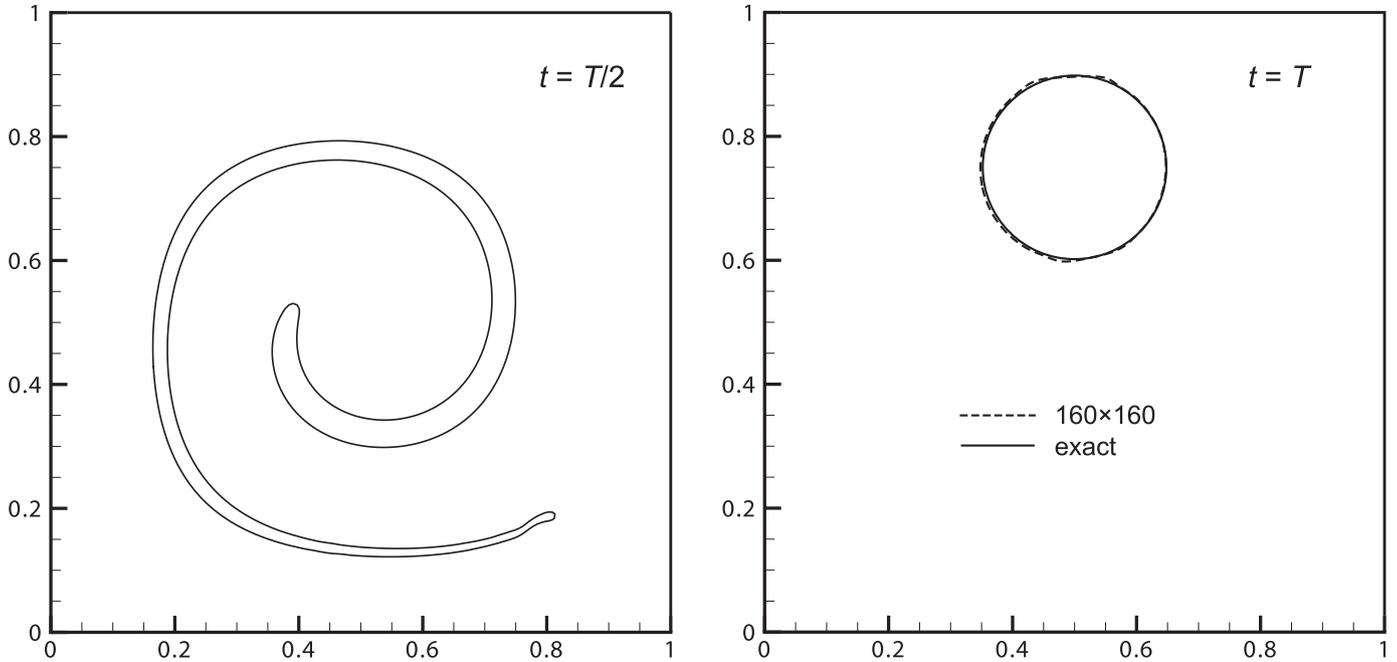


Fig. 26. Interface positions of vortex-in-a-box problem at $t = T/2$ and $t = T$.

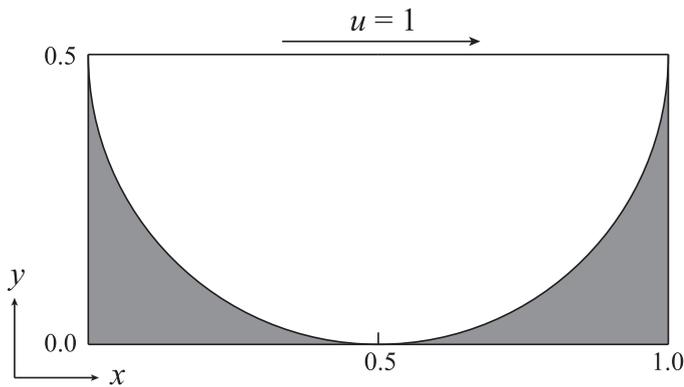


Fig. 27. Lid-driven flow in a semicircular cavity.

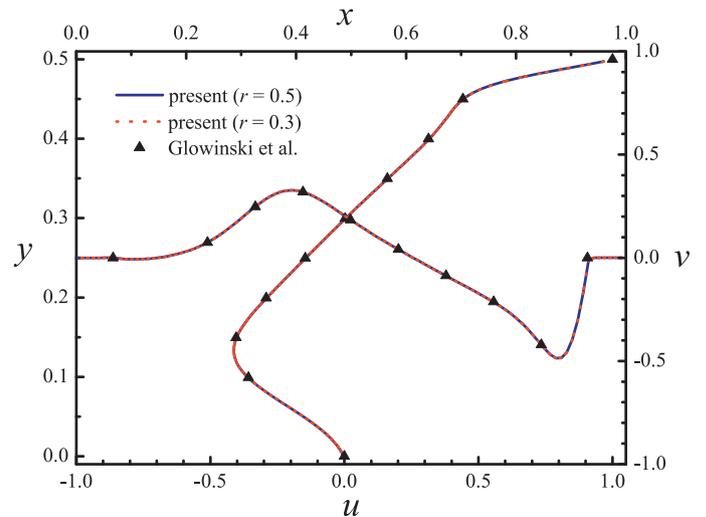


Fig. 28. Velocity distributions on the horizontal and vertical central lines of the cavity.

value of r does not affect the result as long as the convergence solution can be obtained.

Fig. 29 shows the simulated streamlines as well as the velocity field in a local region. From the streamlines, a secondary vortex can be observed forming in the left half of the computational domain, which is in good agreement with the result by Glowinski et al. (see Fig. 4 in Ref. [51]). In the local velocity field we can find velocity vectors only in the fluid region, indicating the solid phase was successfully maintained stationary by the proposed SIMPLER+IBM. Another important requirement for an immersed boundary method for incompressible flows is that the velocity in the entire computational domain should be divergence free. The

numerical result shows the divergence of the velocity field calculated by Eq. (43) fell to the order of 10^{-10} , indicating a divergence-free velocity field resulted by the proposed approach. The result displayed in this section shows the SIMPLER+IBM method described in Section 3.4 can accurately solve incompressible flow

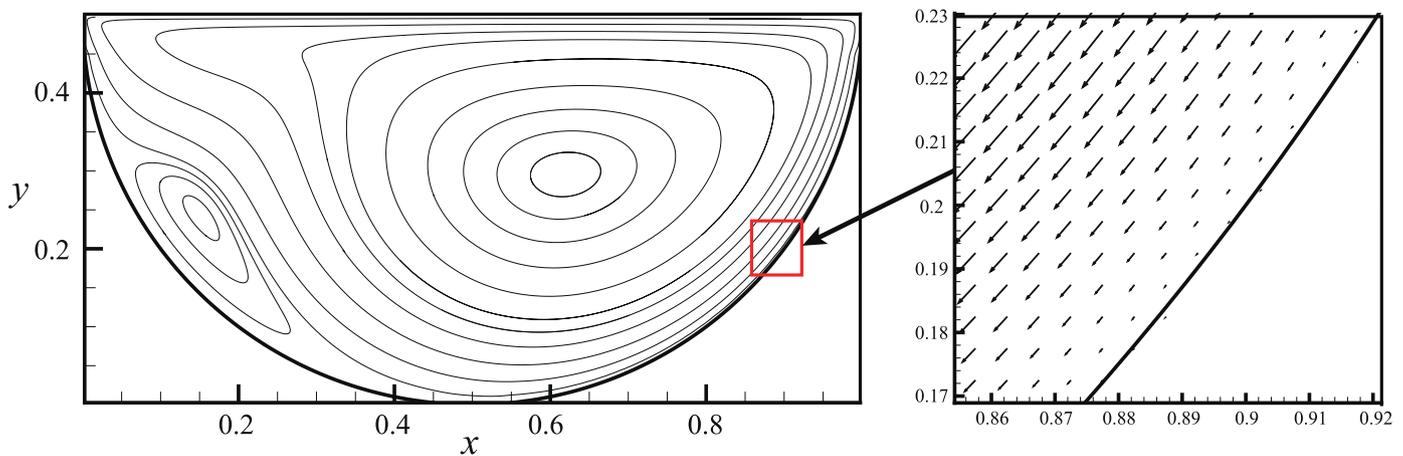


Fig. 29. Streamlines and a local velocity field of the lid-driven flow in a semi-circular cavity.

with stationary phase described by level-set function.

$$\text{Divergence} = \frac{\int_{CV} \nabla \cdot \mathbf{u} dV}{\int_{CV} dV} \cong \frac{(u_{i+1/2,j} - u_{i-1/2,j} + v_{i,j+1/2} - v_{i,j})h}{h^2} \quad (43)$$

References

- [1] Juric D, Tryggvason G. A front-tracking method for dendritic solidification. *J Comput Phys* 1996;123:127–48.
- [2] Udaykumar HS, Mittal R, Shyy W. Computation of solid-liquid phase fronts in the sharp interface limit on fixed grids. *J Comput Phys* 1999;155:535–74.
- [3] Udaykumar HS, Mao L. Sharp-interface simulation of dendritic solidification of solutions. *Int J Heat Mass Transfer* 2002;45:4793–808.
- [4] Al-Ravahi N, Tryggvason G. Numerical simulation of dendritic solidification with convection: two-dimensional geometry. *J Comput Phys* 2002;180:471–96.
- [5] Al-Ravahi N, Tryggvason G. Numerical simulation of dendritic solidification with convection: three-dimensional flow. *J Comput Phys* 2004;194:677–96.
- [6] Zhao P, Heinrich JC. Numerical approximation of a thermally driven interface using finite elements. *Int J Numer Methods Eng* 2003;56:1533–47.
- [7] Zhao P, Heinrich JC, Poirier DR. Numerical simulation of crystal growth in three dimensions using a sharp-interface finite element method. *Int J Numer Methods Eng* 2007;71:25–46.
- [8] Zhao P, Heinrich JC, Poirier DR. Dendritic solidification of binary alloys with free and forced convection. *Int J Numer Methods Fluids* 2005;49:233–66.
- [9] Hu Y, Shi Q, De Almeida VF, Li X. Numerical simulation of phase transition problems with explicit interface tracking. *Chem Eng Sci* 2015;128:92–108.
- [10] Huang R, Wu H, Cheng P. A new lattice Boltzmann model for solid-liquid phase change. *Int J Heat Mass Transfer* 2013;59:295–301.
- [11] Huang R, Wu H. An immersed boundary-thermal lattice Boltzmann method for solid-liquid phase change. *J Comput Phys* 2014;277:305–19.
- [12] Tryggvason G, Scardovelli R, Zaleski S. *Direct numerical simulations of gas-liquid multiphase flows*. New York: Cambridge University Press; 2011.
- [13] Osher S, Sethian J. Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations. *J Comput Phys* 1988;79:12–49.
- [14] Chen S, Merriman B, Osher S, Smereka P. A simple level set method for solving Stefan problems. *J Comput Phys* 1997;135:8–29.
- [15] Tan L, Zabarav N. A level set simulation of dendritic solidification with combined features of front-tracking and fixed-domain methods. *J Comput Phys* 2006;211:36–63.
- [16] Rauschenberger P, Criscione A, Eisenschmidt K, Kintea D, Jakirlić S, Tuković v, Roisman IV, Weigand B, Tropea C. Comparative assessment of volume-of-fluid and level-set methods by relevant to dendritic ice growth in supercooled water. *Comput Fluids* 2013;79:44–52.
- [17] Langer JS, Muller-Krumbhaar H. Stability effects in dendritic crystal growth. *J Cryst Growth* 1977;42:11–14.
- [18] Langer JS, Muller-Krumbhaar H. Theory of dendritic growth. I. Elements of a stability analysis. *Acta Metall* 1978;26:1681–7.
- [19] Ghoneim A. A new technique for numerical simulation of dendritic solidification using a meshfree interface finite element method. *Int J Numer Methods Eng* 2016;107:813–52.
- [20] Hirt CW, Nichols BD. Volume of fluid (VOF) method for the dynamics of free boundaries. *J Comput Phys* 1981;39:201–26.
- [21] López J, Gómez P, Hernández J. A volume of fluid approach for crystal growth simulation. *J Comput Phys* 2010;229:6663–72.
- [22] López J, Gómez P, Hernández J, Faura F. A two-grid adaptive volume of fluid approach for dendritic solidification. *Comput Fluids* 2013;86:326–42.
- [23] Francois MM, Cummins SJ, Dendy ED, Kothe DB, Sicilian JM, Williams MW. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *J Comput Phys* 2006;213:141–73.
- [24] Tong AY, Wang Z. A numerical method for capillarity-dominant free surface flows. *J Comput Phys* 2007;221:506–23.
- [25] Wang Z, Yang J, Koo B, Stern F. A coupled level set and volume-of-fluid method for sharp interface simulation of plunging breaking waves. *Int J Multiphase Flow* 2009;35:227–46.
- [26] Rauschenberger P, Weigand B. A volume-of-fluid method with interface reconstruction for ice growth in supercooled water. *J Comput Phys* 2015;282:98–112.
- [27] Pal D, Bhattacharya J, Dutta P, Chakraborty S. An enthalpy model for simulation of dendritic growth. *Numer Heat Transfer Part B* 2006;550:5978.
- [28] Mencinger J. Numerical simulation of melting in two-dimensional cavity using adaptive grid. *J Comput Phys* 2004;198:243–64.
- [29] Ulvrová M, Labrosse S, Coltice N, Råback P, Tackley PJ. Numerical modelling of convection interacting with a melting and solidification front: application to the thermal evolution of the basal magma ocean. *Phys Earth Planet Inter* 2012;51–66. 206–207
- [30] Karagadde S, Bhattacharya A, Tomar G, Dutta P. A coupled VOF-IBM-enthalpy approach for modeling motion and growth of equiaxed dendrites in a solidifying melt. *J Comput Phys* 2012;231:3987–4000.
- [31] Wu W, Zhang S, Wang S. A novel lattice Boltzmann model for the solid-liquid phase change with the convection heat transfer in the porous media. *Int J Heat Mass Transfer* 2016;104:675–87.
- [32] Beckermann C, Diepers HJ, Steinbach I, Karma A, Tong X. Modeling melt convection in phase-field simulations of solidification. *J Comput Phys* 1999;154:468–96.
- [33] Do-Quang M, Amberg G. Simulation of free dendritic crystal growth in a gravity environment. *J Comput Phys* 2008;227:1772–89.
- [34] Zhao Y, Zhao CY, Xu ZC. Numerical study of solid-liquid phase change by phase field method, for incompressible two-phase flow. *Comput Fluids* 2018;164:94–101.
- [35] Karma A, Rappel WJ. Quantitative phase-field modeling of dendritic growth in two and three dimensions. *Phys Rev E* 1998;57:4323–49.
- [36] Amberg G. Semisharp phase field method for quantitative phase change simulations. *Phys Rev Lett* 2003;91(26):26550–1.
- [37] Sun DL, Tao WQ. A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows. *Int J Heat Mass Transf* 2010;53:645–55.
- [38] Ling K, Li ZH, Sun DL, He YL, Tao WQ. A three-dimensional volume of fluid & level set (VOSET) method for incompressible two-phase flow. *Comput Fluids* 2015;118:293–304.
- [39] Alexiades V, Solomon AD. *Mathematical modeling of melting and freezing process*. Washington DC: Hemisphere Publishing; 1993.
- [40] Youngs DL. *Time-dependent multi-material flow with large fluid distortion. Numerical method for fluid dynamics*. New York: Academic press; 1982.
- [41] Cummins J, Francois M, Kothe B. Estimating curvature from volume fractions. *Comput Struct* 2005;83:425–34.
- [42] López J, Hernández J, Gómez P, Faura F. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *J Comput Phys* 2004;195:718–42.
- [43] Rider WJ, Kothe DB. Reconstructing volume tracking. *J Comput Phys* 1998;141:112–52.
- [44] Slater M, Steed A, Chrysanthou Y. *Computer graphics and virtual environments: from realism to real-time*. New Jersey: Addison Wesley; 2001.
- [45] Patankar SV. *Numerical heat transfer and fluid flow*. New York: McGraw-Hill; 1980.
- [46] Tao WQ. *Numerical heat transfer*. Xian, P.R. China: Xian Jiaotong University Press; 2001.
- [47] Van-Leer B. Towards the ultimate conservative difference scheme. V. A second-order sequel to Godunov's method. *J Comput Phys* 1979;32:101–36.
- [48] Fadlun EA, Verzicco R, Orlandi P, Mohd-Yusof J. Combined immersed-boundary finite-difference methods for three-dimensional complex flow simulations. *J Comput Phys* 2000;161:35–60.

- [49] Liao CC, Lin CA. Simulations of natural and forced convection flows with moving embedded object using immersed boundary method. *Comput Methods Appl Mech Eng* 2012;58–70. 213–216
- [50] Zalesak ST. Fully multidimensional flux-corrected transport algorithms for fluids. *J Comput Phys* 1979;31:335–62.
- [51] Glowinski R, Guidoboni G, Pan TW. Wall-driven incompressible viscous flow in a two-dimensional semi-circular cavity. *J Comput Phys* 2006;216:79–91.
- [52] Carslaw HS, Jaeger JC. *Conduction of heat in solids*. 2nd ed. Oxford: Clarendon Press; 1959. p. 283–286.
- [53] Kurz W, Fisher DJ. *Fundamentals of solidification*. Switzerland: Trans Tech Publications LTD; 1986.
- [54] Glicksman ME, Schaefer RJ, Ayers JD. Dendritic growth—a test of theory. *Metall Trans A* 1976;7:1747–59.
- [55] Bertrand O, Binet B, Combeau H, Couturier S, Delannoy Y, Gobin D, Lacroix M, Quéré PL, Médale M, Mencinger J, Sasat H, Vieira G. Melting driven by natural convection. A comparison exercise: first results. *Int J Therm Sci* 1999;38:5–26.