Journal of Computational Physics 475 (2023) 111827

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

An interface-resolved phase-change model based on velocity decomposition

Min Lu [a,b], Zixuan Yang [a,b,*], Guowei He [a,b]

[a] State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, China
[b] School of Engineering Sciences, University of Chinese Academy of Sciences, Beijing, 100049, China

ARTICLE INFO

Article history:
Received 14 June 2022
Received in revised form 1 November 2022
Accepted 27 November 2022
Available online 9 December 2022

Keywords:
Phase-change model
Velocity decomposition
Multi-phase flow

ABSTRACT

An interface-resolved phase-change model is proposed in the interface-capturing framework based on the coupled level-set and volume of fluid (CLSVOF) method. A velocity decomposition method is employed to ensure the numerical stability and accuracy of interface propagation. Specifically, the velocity $\mathbf{u}$ is decomposed into the potential part $\tilde{\mathbf{u}}$ associated with the phase change and the remaining rotational part $\bar{\mathbf{u}}$. The potential velocity $\tilde{\mathbf{u}}$ is computed by solving a Poisson equation. A modified momentum equation is derived to solve the rotational-part velocity $\bar{\mathbf{u}}$, which is divergence-free. The momentum equation is solved using the Computational Air-Sea Tank (CAS-Tank) developed previously by Yang et al. [1]. To evolve the interface and ensure the mass conservation, a continuous interfacial velocity $\mathbf{u}_\Gamma$ is constructed by adding the velocity $\hat{\mathbf{u}}$ associated with the volume change of liquid (or gas) to $\bar{\mathbf{u}}$, where $\hat{\mathbf{u}}$ is computed by solving a Poisson equation with constant coefficient. The proposed method is verified in the context of the two-dimensional ($2D$) droplet with constant evaporation rate, one-dimensional ($1D$) Stefan and sucking problems, $2D$ droplet evaporation at saturation temperature, $2D$ droplet evaporation below saturation temperature, and a three-dimensional ($3D$) evaporating droplet falling under the gravity. The results show that the proposed method is accurate and stable. The numerical method is also validated by simulating an evaporating droplet falling under the gravity and the numerical results are found to be in agreement with the results in the literature. The interaction of two evaporating droplets is also simulated in a $3D$ domain to show the capability of the proposed method in solving $3D$ problems.

1. Introduction

Phase change takes place in many nature and industrial applications, such as formation of raindrop, spray combustion and water boiling. Numerical methods have been developed to simulate the phase-change problems in both the interface-unresolved and interface-resolved frameworks. The interface-unresolved methods, in which the droplets are modeled as point particles or the two phases are modeled as the liquid-gas mixture, are not the focus of present study. We refer the readers to Subramaniam [2], Senocak and Shyy [3] and Liu et al. [4] for more details. In the interface-resolved framework, the liquid and gas phases are separated by liquid-gas interfaces, represented by either the interface-tracking [5–7] or interface-capturing approaches [8–12]. In interface-resolved simulation of phase-change problems, the volume source as-

* Corresponding author at: State Key Laboratory of Nonlinear Mechanics, Institute of Mechanics, Chinese Academy of Sciences, Beijing, 100190, China.
E-mail address: yangzx@imech.ac.cn (Z. Yang).

https://doi.org/10.1016/j.jcp.2022.111827
0021-9991/

sociated with the phase change is calculated at the interface. This leads to the following difficulties: (1) The source only exists at the liquid-gas interface, resulting a singular term in the governing equation, which poses challenges for numerical accuracy and stability. (2) The singular source causes velocity jump in the total velocity field $\mathbf{u}$ at grid cells containing the liquid–gas interface, but a continuous interfacial velocity $\mathbf{u}_\Gamma$ is required to evolve the interface.

In many previous studies, the volume source was treated smoothly to ensure numerical stability [13]. For example, the front-tracking method proposed by Tryggvason and coworkers [5,14] was extended to solve problems with phase change [15–17,7,18]. In their methods, the mass source associated with phase change at the interface was smoothed and incorporated into the continuity equation. A projection algorithm was adopted to calculate velocity $\mathbf{u}$. To evolve the interface of evaporating droplets, Irfan and Muradoglu [18] subtracted the velocity of phase change $\tilde{\mathbf{u}}$ from the total velocity $\mathbf{u}$, and then added the volume change velocity of liquid $\hat{\mathbf{u}}$ to construct the interfacial velocity $\mathbf{u}_\Gamma$ (i.e., $\mathbf{u}_\Gamma = \mathbf{u} - \tilde{\mathbf{u}} + \hat{\mathbf{u}}$). Following the similar philosophy, phase-change models had also been developed in the front-capturing framework. Son and Dhir [19] enabled the level-set method to perform simulations of phase change. In their method, the momentum and continuity equations were solved using the same strategy as used by Juric and Tryggvason [15], and the source was smoothed using a mollified step function. To construct the interfacial velocity $\mathbf{u}_\Gamma$ of the boiling film, they added the volume expansion velocity of gas phase $\hat{\mathbf{u}}$ to the total velocity $\mathbf{u}$ (i.e., $\mathbf{u}_\Gamma = \mathbf{u} + \hat{\mathbf{u}}$), where $\hat{\mathbf{u}}$ was calculated algebraically by dividing the interfacial mass transfer rate $\dot{M}$ with the density of gas phase ρ_g. Welch and Wilson [20] modified the volume of fluid (VOF) method to simulate the boiling flows. In their method, the source was smoothed, and the total velocity $\mathbf{u}$ was used to evolve the VOF function by a time-splitting geometric advection technique. As noted by Kharangate and Mudawar [13], although the artificial smoothing of source term improves the numerical stability, the physical authenticity is not fully preserved.

Schlottke and Weigand [21] and Sato and Niceno [22] reported the observation of spurious velocity when the source points were treated sharply within their VOF framework. As noted by Schlottke and Weigand [21], the total velocity $\mathbf{u}$ is mass weighted and therefore not volume conservative in the VOF method. To improve the volume conservation and ensure the numerical stability, they constructed a volume weighted velocity $\mathbf{u}_{\text{vol}}$ at interface cells, which consisted of liquid-part velocity $\mathbf{u}_l$ and gas-part velocity $\mathbf{u}_g$. Then, the liquid-part velocity $\mathbf{u}_l$ was adopted to transport the interface (i.e., $\mathbf{u}_\Gamma = \mathbf{u}_l$). Another effective approach to treat the source sharply was proposed by Nguyen et al. [23], in which a coupled level-set and ghost fluid method was employed to simulate the reaction flow. In their method, the source term was considered as a Dirichlet boundary condition of velocity, resulting in a velocity jump at the interface. To construct a continuous liquid velocity $\mathbf{u}_l$ and gas velocity $\mathbf{u}_g$ over the interface, $\mathbf{u}_l$ and $\mathbf{u}_g$ was extended to the gas ($\mathbf{u}_l^{\text{ghost}}$) and liquid ($\mathbf{u}_g^{\text{ghost}}$) phase, respectively, using the ghost fluid method [24]. A reaction velocity $\hat{\mathbf{u}}$ ($= -\dot{M}/\rho_l \cdot \mathbf{n}$) was then derived from the volume change of liquid phase and added into the continuous liquid velocity $\mathbf{u}_l^{\text{ghost}}$ to evolve the interface (i.e., $\mathbf{u}_\Gamma = \mathbf{u}_l^{\text{ghost}} + \hat{\mathbf{u}}$). Later, Gibou et al. [25] employed the numerical method of Nguyen et al. [23] and the Stefan solution of Gibou et al. [26] to simulate film boiling problems. However, as noted by Tanguy et al. [27], this method performed good under the limitation of low density ratio, while unrealistic mass prediction was observed for high density ratio problems. To address this issue, Tanguy et al. [27] constructed a divergence-free velocity $\bar{\mathbf{u}}$ by projecting the extrapolated $\mathbf{u}_l^{\text{ghost}}$ onto its divergence-free part using the Hodge decomposition methodology. The interfacial velocity $\mathbf{u}_\Gamma$ was then constructed by adding the volume change velocity of liquid $\hat{\mathbf{u}}$ ($= -\dot{M}/\rho_l \cdot \mathbf{n}$) into $\bar{\mathbf{u}}$. This treatment was also adopted by Villegas et al. [28] and Sahut et al. [29]. Palmore Jr and Desjardins [30] also employed a similar idea to construct the divergence-free velocity $\bar{\mathbf{u}}$ in the VOF framework for simulating the phase-change problems. The interface was advected by $\mathbf{u}_\Gamma = \bar{\mathbf{u}} - \dot{M}/\rho_l \cdot \mathbf{n}$. Bayat et al. [31] simulate the ice formation with convection effects using sharp-interface method. In their method, $\mathbf{u}_\Gamma$ is calculated from the Stefan condition and is applied as a Dirichlet boundary condition for the irregular gas-solid interface. For more details on solving parabolic problems with discontinuities on irregular domain, see Guittet et al. [32], Bochkov and Gibou [33], and Egan and Gibou [34].

Recently, another divergence-free velocity extension strategy was proposed by Malan et al. [35] in the geometric VOF method framework. In their method, after the calculation of the total velocity $\mathbf{u}$, a potential velocity $\tilde{\mathbf{u}}$ caused by phase change was subtracted from $\mathbf{u}$ to construct the divergence-free velocity field $\bar{\mathbf{u}}$. The potential velocity $\tilde{\mathbf{u}}$ was given by the solution of a Poisson equation. Thus, unlike the ghost fluid method, the velocity extrapolation of liquid or gas phase were not required. The interface was evolved using $\bar{\mathbf{u}}$ and then the VOF value of the interface cells were modified to account for the volume change caused by the phase change. Later, Scapin et al. [36] and Zhao et al. [37] adopted the similar idea to their phase-change model. In their method, the volume change was considered as an explicit velocity $\hat{\mathbf{u}}$ along the interface normal, i.e., $\hat{\mathbf{u}} = -\dot{M}/\rho \cdot \mathbf{n}$. The interface was, therefore, advected by the interfacial velocity $\mathbf{u}_\Gamma$ ($= \bar{\mathbf{u}} + \hat{\mathbf{u}}$). Due to the divergence-free velocity extension strategy proposed by Malan et al. [35] on existing incompressible two-fluid flow solver is straightforward, we attempted to adopt it to our numerical framework, in which the interface was captured by the CLSVOF method [1], and the interface was kept sharp without smoothing the density across the interface. However, spurious velocity was observed in the divergence-free velocity field $\bar{\mathbf{u}}$ when simulating the evaporation of a stationary droplet. After the numerical error accumulated for a certain time duration, the interface was found to evolve in an unphysical manner and finally the simulation diverged (see Sec. 3.1). This observation was also noted by Martinez et al. [38], and the interface capturing scheme in their work was also the CLSVOF method.

Inspired by the method of subtracting the potential velocity $\tilde{\mathbf{u}}$ associated with the phase change from the total velocity $\mathbf{u}$, we propose a new phase-change model based on the velocity decomposition method. The velocity decomposition method has been applied to solve the wave-structure interaction problems to reduce the numerical dissipation and disper-

sion [39–41]. In the present study, we apply this method to solve the phase change problem. The total velocity field $\mathbf{u}$ is decomposed into potential part $\tilde{\mathbf{u}}$ and rotational part $\bar{\mathbf{u}}$, and thus $\mathbf{u}$ is not solved directly. The potential velocity $\tilde{\mathbf{u}}$ induced solely by the phase change is given by the solution of a Poisson equation, and the rotational-part velocity $\bar{\mathbf{u}}$ is solved using the computational-fluid-dynamics (CFD) solver. As such, the divergence of the potential velocity $\tilde{\mathbf{u}}$ at the interface is included by the volume change associated with the phase change, while the rotational-part velocity $\bar{\mathbf{u}}$ is divergence-free throughout the computational domain. Besides, the interface is evolved by a continuous interfacial velocity $\mathbf{u}_\Gamma$, which is constructed by adding the velocity $\hat{\mathbf{u}}$ induced by the volume change of liquid (or gas) to the rotational-part velocity $\bar{\mathbf{u}}$.

The proposed method shows the following advantages: (1) The nonlinear interaction between the numerical error of the potential part and the rotational part is avoided, which improves the numerical accuracy and ensures the numerical stability. (2) The divergence-free velocity $\bar{\mathbf{u}}$ is obtained directly from the CFD solver rather than being reconstructed at each time step, which simplifies the implementation. (3) The interfacial velocity $\mathbf{u}_\Gamma$ is divergence-free inside the mixed cells (i.e. the cells containing both liquid and gas phases), which ensures the mass conservation during the simulation. The proposed method is verified by comparing the simulation results with the analytical solutions of several benchmark test cases and the capability of the present method for simulating the droplet evaporation under gravity in both two-dimensional ($2D$) and three-dimensional ($3D$) domains is subsequently validated.

This paper is organized as follows. In Sec. 2, the governing equations and numerical algorithms are described. The test results are presented and discussed in Sec. 3, followed by conclusions in Sec. 4.

2. Numerical method

2.1. Governing equations

In the "one-fluid" formulation, the continuity equation accounting for the phase change can be expressed as

$$\nabla \cdot \mathbf{u} = \left(\frac{1}{\rho_v} - \frac{1}{\rho_l}\right)\frac{\dot{M} S_\Gamma}{V_\Gamma}, \tag{1}$$

where $\mathbf{u}$ is the velocity, ρ is the fluid density, the subscripts v and l denote the vapor and liquid phases, respectively, $\dot{M}$ represents the mass change rate per interface area, S_Γ is the interfacial area inside the cell with interface and V_Γ is the volume of this cell. The conservative form of the momentum equation is

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla\cdot(\rho\mathbf{u}\mathbf{u}) = -\nabla p + \nabla\cdot(2\mu\mathbf{S}) + \rho\mathbf{g} + \mathbf{f}_s, \tag{2}$$

where t represents the time, μ denotes the fluid viscosity, p is the pressure, $\mathbf{S}$ is the strain-rate tensor, $\mathbf{g}$ is the gravitational acceleration, and $\mathbf{f}_s$ is the surface tension force. The transport equation of temperature is given in a conservative form as

$$\frac{\partial(\rho c_p\theta)}{\partial t} + \nabla\cdot(\rho c_p\theta\mathbf{u}) = \nabla\cdot k\nabla\theta - \left[h_{lg} + \left(c_{p,v} - c_{p,l}\right)(\theta_{sat} - \theta_\Gamma)\right]\frac{\dot{M} S_\Gamma}{V_\Gamma}, \tag{3}$$

where θ represents the temperature, c_p is the specific heat at constant pressure, and k is the thermal conductivity. The last term accounts for the jump in enthalpy due to the phase change, where h_{lg} is the latent heat. The governing equation of vapor mass fraction Y is expressed as

$$\frac{\partial Y}{\partial t} + \nabla\cdot(Y\mathbf{u}) = \lambda\nabla^2 Y, \tag{4}$$

where λ denotes the mass diffusion coefficient.

To ensure the energy and mass conservation across the interface, the following relations are satisfied

$$\dot{M} = \frac{k_g}{h_{lg}}\left(\frac{\partial\theta}{\partial n}\right)_g^\Gamma + \frac{k_l}{h_{lg}}\left(\frac{\partial\theta}{\partial n}\right)_l^\Gamma, \tag{5}$$

$$\dot{M}Y_l^\Gamma - \dot{M}Y_g^\Gamma + \left(\rho_g\lambda\frac{\partial Y}{\partial n}\right)^\Gamma = 0, \tag{6}$$

where Y_l^Γ and Y_g^Γ are the vapor mass fraction at the interface in liquid and gas phases, respectively. For a mono-component liquid, $Y_l^\Gamma = 1$ and the gradient of vapor mass fraction in liquid phase is zero, which means that Eq. (4) only needs to be solved in the gas phase. Thus, Eq. (6) can be rewritten as

$$\dot{M} = \frac{-\rho_v\lambda\nabla_\Gamma Y_g\cdot\mathbf{n}}{(1 - Y_g^\Gamma)}, \tag{7}$$

where ∇_Γ represents the gradient at the interface. The boundary condition for Y_g at the interface is calculated using the Clausius–Clapeyron relation, i.e.,

$$Y_g^{\Gamma} = \frac{P_{v,sat}^{\Gamma} M_l}{(P_{atm} - P_{v,sat}^{\Gamma}) M_g + P_{v,sat}^{\Gamma} M_l}, \tag{8}$$

$$P_{v,sat}^{\Gamma} = P_{atm} \exp\left[-\frac{h_{lg} M_l}{R} \left(\frac{1}{\theta_{\Gamma}} - \frac{1}{\theta_{sat}} \right) \right], \tag{9}$$

where $P_{v,sat}^{\Gamma}$ is the saturation vapor pressure corresponding to the interface temperature θ_{Γ}, θ_{sat} is the saturation temperature at ambient pressure P_{atm}, R denotes the perfect gas constant, and M_l and M_g are the molar masses of the liquid and gas, respectively. As note by Villegas et al. [28], both Eqs. (5) and (7) can be used to calculate the interfacial mass transfer rate when the interface temperature θ_{Γ} is significantly lower than the saturation temperature θ_{sat}, while the singularity appears in the denominator of Eq. (7) if the temperature of the interface approaches the saturation temperature. Thus, in the following sections, the interfacial mass transfer rate is calculated by Eq. (5) when the evaporation takes place at the saturation temperature, and Eq. (7) is adopted when the evaporation takes place below the saturation temperature.

2.2. Velocity decomposition

In the present numerical framework, the source points are sharply distributed inside the computational cells containing the interface. Due to the velocity jump at the interface cells, the total velocity field $\mathbf{u}$ cannot be used directly to evolve the interface. A continuous velocity is in demand to avoid numerical instability when the interface is evolved. Here, we develop a new phase-change model based on the velocity decomposition method, which is accurate and numerically stable.

Inspired by the method of subtracting the potential velocity $\tilde{\mathbf{u}}$ induced by the phase change from the total velocity $\mathbf{u}$ proposed by Malan et al. [35], we decompose the total velocity $\mathbf{u}$ into the potential part $\tilde{\mathbf{u}}$ and the rotational part $\bar{\mathbf{u}}$, i.e.,

$$\mathbf{u} = \tilde{\mathbf{u}} + \bar{\mathbf{u}}. \tag{10}$$

Thus, Eq. (1) can be rewritten as

$$\nabla \cdot \tilde{\mathbf{u}} = \left(\frac{1}{\rho_v} - \frac{1}{\rho_l} \right) \frac{\dot{M} S_{\Gamma}}{V_{\Gamma}}, \tag{11}$$

and

$$\nabla \cdot \bar{\mathbf{u}} = 0, \tag{12}$$

where the potential-part velocity $\tilde{\mathbf{u}}$ accounts for the volume change caused by phase change and $\bar{\mathbf{u}}$ is the remaining rotational divergence-free part. The potential velocity $\tilde{\mathbf{u}}$ is determined by the potential flow theory as

$$\begin{cases} \nabla \cdot \left(\frac{1}{\rho} \nabla \varphi \right) = \left(\frac{1}{\rho_v} - \frac{1}{\rho_l} \right) \frac{\dot{M} S_{\Gamma}}{V_{\Gamma}} \\ \tilde{\mathbf{u}} = \frac{1}{\rho} \nabla \varphi \end{cases}, \tag{13}$$

where φ is the velocity potential associated with the mass source induced by the phase change.

The potential velocity $\tilde{\mathbf{u}}$ is also the solution of the following Euler equation

$$\rho \left(\frac{\partial \tilde{\mathbf{u}}}{\partial t} + \tilde{\mathbf{u}} \cdot \nabla \tilde{\mathbf{u}} \right) = -\nabla \tilde{p}. \tag{14}$$

In order to derive the governing equation for the rotational part $\bar{\mathbf{u}}$, we subtract Eq. (14) from Eq. (2) to yield

$$\begin{aligned} \frac{\partial \left[\rho (\mathbf{u} - \tilde{\mathbf{u}}) \right]}{\partial t} + \nabla \cdot (\rho \mathbf{u}\mathbf{u}) - \nabla \cdot \left(\rho \tilde{\mathbf{u}} \tilde{\mathbf{u}} \right) - \mathbf{u} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) \right] + \tilde{\mathbf{u}} \left[\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \tilde{\mathbf{u}} \right) \right] = \\ -\nabla (p - \tilde{p}) + \nabla \cdot (2\mu \mathbf{S}) + \rho \mathbf{g} + \mathbf{f}_s. \end{aligned} \tag{15}$$

In addition, the flow field should be governed by the general form of continuity equation

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0. \tag{16}$$

Using the relation in Eq. (10), the general form of continuity equation for $\tilde{\mathbf{u}}$ is

$$\frac{\partial \rho}{\partial t} + \nabla \cdot \left(\rho \tilde{\mathbf{u}} \right) = -\nabla \cdot (\rho \bar{\mathbf{u}}). \tag{17}$$

Substituting Eqs. (10), (16) and (17) into Eq. (15) yields

$$\frac{\partial (\rho \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}} \bar{\mathbf{u}}) = -\nabla \bar{p} + \nabla \cdot \left(2\mu \bar{\mathbf{S}} \right) + \rho \mathbf{g} + \mathbf{f}_s - \nabla \cdot \left(\rho \bar{\mathbf{u}} \tilde{\mathbf{u}} \right) - (\rho \bar{\mathbf{u}} \cdot \nabla) \tilde{\mathbf{u}}, \tag{18}$$

where $\bar{p}$ and $\bar{\mathbf{S}}$ are the pressure and strain-rate tensor of the rotational-part velocity, respectively. This equation degenerates to the original momentum equation when phase change is absent (i.e., $\tilde{\mathbf{u}} = 0$). In this sense, any existing solver for two-fluid flow can be readily modified by adding the last two source terms in Eq. (18) to simulate phase-change problems.

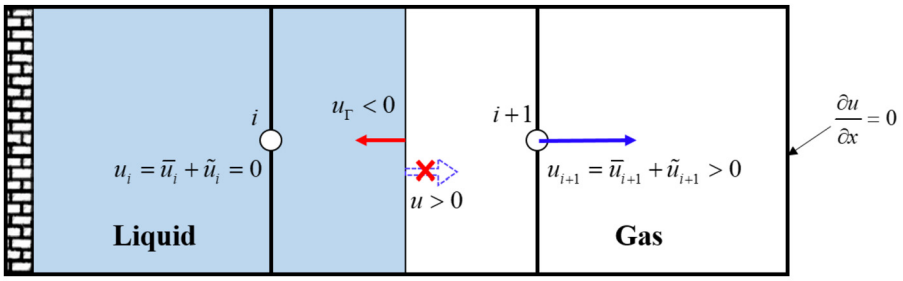


Fig. 1. Schematic diagram of the discretized velocity field for a 1*D* evaporation problem. The regression velocity $\mathbf{u}_\Gamma$ is negative due to the mass loss of the liquid phase. The total velocity $\mathbf{u}$ is positive in gas phase and is zero in liquid phase. The interpolated $\mathbf{u}$ at the interface is positive and should not be used to evolve the interface.

2.3. Interface advancement

The interface advancement of multiphase flows with phase change is challenging because of the velocity jump in $\mathbf{u}$ at the interface. This point can be recognized from Fig. 1, in which the discretized velocity field for a 1*D* evaporation problem is displayed. When evaporation occurs, the regression velocity $\mathbf{u}_\Gamma$ is negative due to the mass loss of the liquid phase. The velocity in the gas phase $\mathbf{u}_{i+1} > 0$, while the velocity in the liquid phase $\mathbf{u}_i = 0$ due to the existence of solid wall on the left domain boundary. As such, the interpolated $\mathbf{u}$ at the interface is in the opposite direction of $\mathbf{u}_\Gamma$. Therefore, $\mathbf{u}$ should not be used to evolve the interface. To overcome this issue, Scapin et al. [36] constructed a smooth interface propagation velocity $\mathbf{u}_\Gamma$ as

$$\mathbf{u}_\Gamma = \bar{\mathbf{u}} + \hat{\mathbf{u}}, \tag{19}$$

and

$$\hat{\mathbf{u}} = -\frac{\dot{M}}{\rho_l}\mathbf{n}, \tag{20}$$

where $\hat{\mathbf{u}}$ is the velocity associated with the volume change of liquid phase induced by the phase change.

The interfacial velocity $\mathbf{u}_\Gamma$ is then used to evolve the VOF function ψ. Following the philosophy of Scapin et al. [36], we evolve both LS function ϕ and VOF function ψ by $\mathbf{u}_\Gamma$, i.e., the following transport equations of ϕ and ψ are solved:

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\phi \mathbf{u}_\Gamma) = \phi \nabla \cdot \mathbf{u}_\Gamma, \tag{21}$$

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\psi \mathbf{u}_\Gamma) = 0. \tag{22}$$

However, the interface evolves in an unphysical manner if $\hat{\mathbf{u}}$ is constructed by Eq. (20) in our numerical framework, because $\dot{M}$ is sharply distributed at the interface cells. This point is shown in Sec. 3.1. Thus, to evolve the interface reasonably in the present numerical framework, we must reconstruct a interface regression velocity $\hat{\mathbf{u}}$, which is continuous at the interface. Based on the Gauss's law, the net flux over the interface is equivalent to the source inside the control volume. Thus, to ensure the mass conservation, the source points on the interface are moved into the liquid (or gas) phase at $\mathbf{x}_{\Gamma'}$, which is defined as

$$\mathbf{x}_{\Gamma'} = \mathbf{x}_\Gamma + \Delta l \mathbf{n}, \tag{23}$$

where Δl is the distance between the relocated source points and the interface. The position of the relocated source points is chosen according to the boundary condition. Specifically, the source points are shifted into the liquid phase (i.e., Δl is negatively valued) if the gas can leave the computational domain freely, otherwise the source points are shifted into the gas phase (i.e., Δl is positively valued). For example, as implemented in Sec. 3.1, the source points are relocated inside the liquid because the gas can leave the computation domain freely. A schematic diagram of the implementation is shown in Fig. 2, in which the solid blue line is the liquid-gas interface and the value of $\dot{M}$ is computed at the interface (red square) through Eq. (13). The positions of these source points are then moved inside the liquid (red dot) along the normal direction of the interface to calculate $\hat{\mathbf{u}}$. On the contrary, as implemented in Sec. 3.2.1, the source points are shifted to the gas when the liquid can leave the computational domain freely. According to the Gauss's law, the velocity $\hat{\mathbf{u}}$ induced by volume change in Eq. (19) can be found using the following potential flow solution

$$\begin{cases} \nabla \cdot \nabla \varphi' = \frac{-\dot{M} S_\Gamma}{V_\Gamma} \\ \hat{\mathbf{u}} = \frac{1}{\rho'} \nabla \varphi' \end{cases}, \tag{24}$$

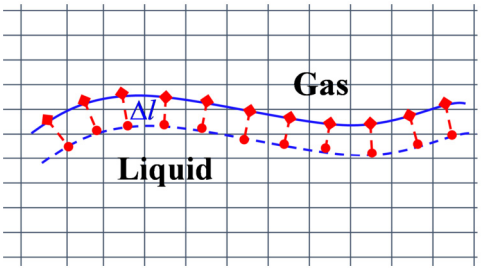


Fig. 2. Schematic diagram of the source points. The blue solid line is the liquid-gas interface and the blue dashed line is plotted by shifting the interface to the liquid phase along the normal direction. The potential velocity $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$ are induced by mass source located at the square red solid points at the interface and the circle red solid points in the liquid phase, respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where φ' denotes the potential field of source associated with the volume change of liquid (or gas), and ρ' is an effective density depending on where the mass source is relocated. Specifically, ρ' is set to either ρ_v or ρ_l if the mass source is relocated in the vapor or liquid phase, respectively.

We note here that $\tilde{\mathbf{u}}$ and $\hat{\mathbf{u}}$ are two different velocity fields, though they are both induced by the mass transfer rate $\dot{M}$ associated with the phase change. The potential velocity of $\tilde{\mathbf{u}}$ is associated with the volume change of the whole domain, and the volume source points are located at the liquid-gas interface. Thus, the velocity jump is captured over the cells with interface, which preserves the physical authenticity of phase change. However, such a velocity jump in $\mathbf{u}$ cannot provide accurate interface velocity as shown in Fig. 1. To avoid the velocity jump over the interface for the interface advection, we decompose the potential-part velocity $\tilde{\mathbf{u}}$ from the total velocity $\mathbf{u}$. The remaining rotational-part velocity $\bar{\mathbf{u}}$ is continuous, while it is not associated with the volume change. To construct the interface regression velocity, we shift the source points from the interface to liquid side along the normal direction, resulting in a new potential velocity $\hat{\mathbf{u}}$ given by Eq. (24), which is determined by the volume loss of the liquid phase. Because the source point is away from the interface, the induced $\hat{\mathbf{u}}$ is continuous at the interface. Thus, the interface is accurately evolved using $\mathbf{u}_\Gamma$ $(=\bar{\mathbf{u}}+\hat{\mathbf{u}})$.

In fact, from a continuum perspective, the velocity at the interface is not provided by $\tilde{\mathbf{u}}$ due to the singularity there, while $\tilde{\mathbf{u}}$ is correct for the rest of the computational domain. To address this issue, plenty of works suggested to find the accurate interface velocity at the interface by calculating an interface regression velocity $\hat{\mathbf{u}}$ based on mass conservation (Irfan and Muradoglu [18], Malan et al. [35], Scapin et al. [36], Zhao et al. [37]), and thus $\hat{\mathbf{u}}$ is accurate at the interface. In other words, the real velocity consists of $\mathbf{u}_\Gamma$ $(=\bar{\mathbf{u}}+\hat{\mathbf{u}})$ at the interface and $\mathbf{u}$ $(=\bar{\mathbf{u}}+\tilde{\mathbf{u}})$ in the remainder of the flow field. From a discretized point of view, due to the velocity jump of $\tilde{\mathbf{u}}$ at the interface, the interpolated $\tilde{\mathbf{u}}$ cannot be used to evolve the interface, while $\hat{\mathbf{u}}$ is accurate at the interface (see Fig. 1). Therefore it is appropriate to use $\mathbf{u}_\Gamma$ $(=\bar{\mathbf{u}}+\hat{\mathbf{u}})$ to evolve the interface. However, $\mathbf{u}_\Gamma$ should not be used to evolve the momentum equation, because it does not give the real velocity in the remainder of the flow field. The most accurate treatment is to set $\mathbf{u}_\Gamma$ as an enforced condition at the interface to evolve the momentum equation, as proposed by Bayat et al. [31]. In our numerical framework, in consideration of the algorithm simplicity, we do not consider the velocity of the interface $\mathbf{u}_\Gamma$ in momentum transport. Instead, we directly use $\tilde{\mathbf{u}}$, which is defined at cell faces, to conduct interpolation. When the interpolation of $\tilde{\mathbf{u}}$ needs to be performed based on the stencil points on different sides of the interface, a third-order CUI scheme is used to avoid numerical oscillations (see Sec. 2.5). A similar treatment is applied by Irfan and Muradoglu [18], Malan et al. [35], Scapin et al. [36], Zhao et al. [37], and the results are reasonable.

From the above description about the algorithms, it is understood that we solve four Poisson equations in each time step, one for $\tilde{\mathbf{u}}$, two for $\bar{\mathbf{u}}$ (RK2 scheme), and one for $\hat{\mathbf{u}}$. In the method of Malan et al. [35] and Scapin et al. [36], if the RK2 scheme is also used to evolve the momentum equation, three Poisson equations need to be solved, including two for $\mathbf{u}$ and one for $\tilde{\mathbf{u}}$. We solve one more Poisson equation to generate the velocity for evolving the interface. Otherwise, the simulation results are incorrect (see Fig. 9 and the corresponding discussions). In this regard, it is this additional Poisson equation that allows us to simulation the phase-change problems in the CLSVOF framework. In other words, in the present framework based on the CLSVOF method, solving an additional Poisson equation of $\hat{\mathbf{u}}$ is necessary. In addition, the tolerance for Eq. (24) is not required to be as small as that for calculating $\bar{\mathbf{u}}$ and $\tilde{\mathbf{u}}$, because $\hat{\mathbf{u}}$ only accounts for the mass conservation without any direct participation in the evolution of the flow field. Based on our test results, solving the Poisson equation of $\hat{\mathbf{u}}$ occupies 17.6% of the total simulation cost.

The numerical algorithm for solving Eqs. (21) and (22) are detailed by Sussman [42]. After the interface advances to the next time step, the physical properties are updated by the LS function as

$$\eta = \eta_g + (\eta_l - \eta_g) H(\phi), \tag{25}$$

where η can be either the density ρ, viscosity μ, specific heat capacity c_p, or thermal conductivity k, and $H(\phi)$ is the Heaviside function, defined as

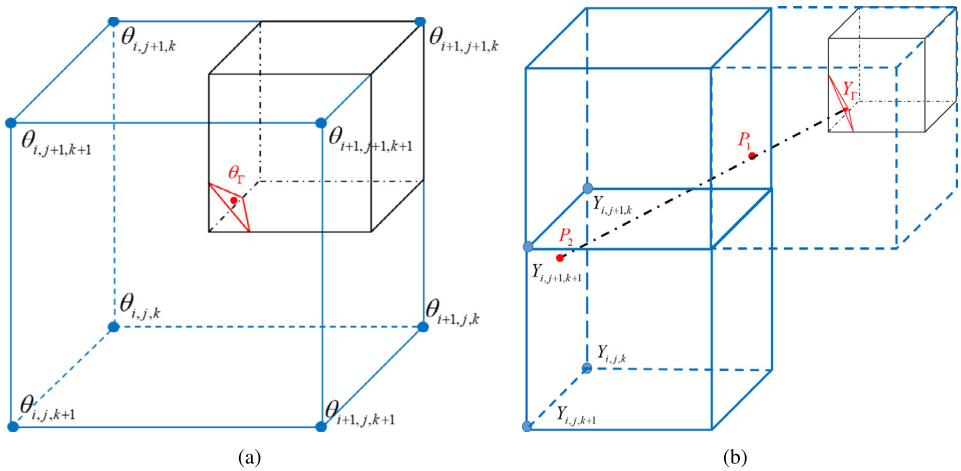


Fig. 3. Schematic diagram of calculating (a) the temperature at the interface θ_Γ and (b) the vapor mass fraction gradient at the interface $\nabla_\Gamma Y_g$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$H(\phi) = \begin{cases} 0 & \phi \le 0 \\ 1 & \phi > 0 \end{cases}. \tag{26}$$

2.4. Calculation of interfacial mass transfer rate and surface area

According to Eq. (1), the interfacial mass transfer rate $\dot{M}$, the interfacial area inside one cell S_Γ and the volume of cell V_Γ are needed to calculate the source term. The interfacial mass transfer rate $\dot{M}$ can be calculated by either Eq. (5) or Eq. (7) in a fully coupled system. Here, we solve Eq. (7) and thus the vapor mass fraction at the interface Y_Γ and the gradient of vapor mass fraction near the interface $\nabla_\Gamma Y_g$ are needed to calculate the source term. Based on the Clausius-Clapeyron relation given by Eqs. (8) and (9), the vapor mass fraction at the interface Y_Γ is a function of the temperature at the interface θ_Γ. In order to calculate θ_Γ, a linear interpolation scheme is utilized and a schematic diagram for the linear interpolation is shown in Fig. 3(a). As shown, if an interface element is located inside the computational cell $(i+1, j+1, k)$, the temperature at the interface θ_Γ can be calculated as

$$\begin{cases} x_d = \frac{x_\Gamma - x_i}{x_{i+1} - x_i};\ y_d = \frac{y_\Gamma - y_j}{y_{j+1} - y_j};\ z_d = \frac{z_\Gamma - z_k}{z_{k+1} - z_k} \\ \theta_\Gamma = \ \theta_{i,j+1,k+1}\left[(1-x_d)\, y_d z_d\right] + \theta_{i+1,j+1,k+1}\left(x_d y_d z_d\right) \\ \quad + \theta_{i,j,k+1}\left[(1-x_d)\,(1-y_d)\, z_d\right] + \theta_{i+1,j,k+1}\left[x_d\,(1-y_d)\, z_d\right] \\ \quad + \theta_{i,j+1,k}\left[(1-x_d)\, y_d\,(1-z_d)\right] + \theta_{i+1,j+1,k}\left[x_d y_d\,(1-z_d)\right] \\ \quad + \theta_{i,j,k}\left[(1-x_d)\,(1-y_d)\,(1-z_d)\right] + \theta_{i+1,j,k}\left[x_d\,(1-y_d)\,(1-z_d)\right]. \end{cases} \tag{27}$$

An embedded boundary method applied by Zhao et al. [37] is then used to calculate the vapor mass fraction gradient $\nabla_\Gamma Y_g$ at the interface. As shown in Fig. 3(b), we draw a line along the normal direction of the interface towards the gas phase. The intersections of the normal line with the cell faces are two stencil points P_1 and P_2, which are marked by red points. The vapor mass fractions at these two intersections are calculated from the nodal values marked by the blue points using the bilinear interpolation scheme. Taking the calculation of Y_{P2} as an example, the value of Y_{P2} is given by

$$Y_{P2} = Y_{i,j,k}\left[(1-y_d)\,(1-z_d)\right] + Y_{i,j+1,k}\left[y_d\,(1-z_d)\right] + Y_{i,j,k+1}\left[(1-y_d)\, z_d\right] + Y_{i,j+1,k+1}\left(y_d z_d\right), \tag{28}$$

where the definitions of y_d and z_d are provided in Eq. (27) with $(x_\Gamma, y_\Gamma, z_\Gamma)$ being the coordinates of point Y_{P2}. The calculation of Y_{P1} can be deduced accordingly. After Y_{P1} and Y_{P2} are known, the vapor gradient at the interface is given by

$$\left(\frac{\partial Y}{\partial n}\right)_\Gamma = \frac{1}{d_{\Gamma-2} - d_{\Gamma-1}}\left[\frac{d_{\Gamma-2}}{d_{\Gamma-1}}\left(Y_\Gamma - Y_{P1}\right) - \frac{d_{\Gamma-1}}{d_{\Gamma-2}}\left(Y_\Gamma - Y_{P2}\right)\right], \tag{29}$$

where $d_{\Gamma-1}$ and $d_{\Gamma-2}$ are the distances from interface to the two intersections, respectively, and $(x_\Gamma, y_\Gamma, z_\Gamma)$ are the coordinates of the interface center provided by the CLSVOF algorithm. After that, the vapor mass fraction Y_Γ at the interface is calculated through Eqs. (8) and (9). The quadratic extrapolation approach can also be used to find the vapor gradient at the interface [43,44].

It should be noted that in Sec. 3.2, the vapor mass fraction is not coupled in the system and thus the interfacial mass transfer rate $\dot{M}$ is calculated from Eq. (5). Because the phase change takes place at the saturation temperature in the test cases of Sec. 3.2, the temperature gradient inside the water is 0 and thus the mass transfer rate only depends on the temperature gradient in the gas phase. This can be calculated through Eq. (29) with Y being replaced by θ.

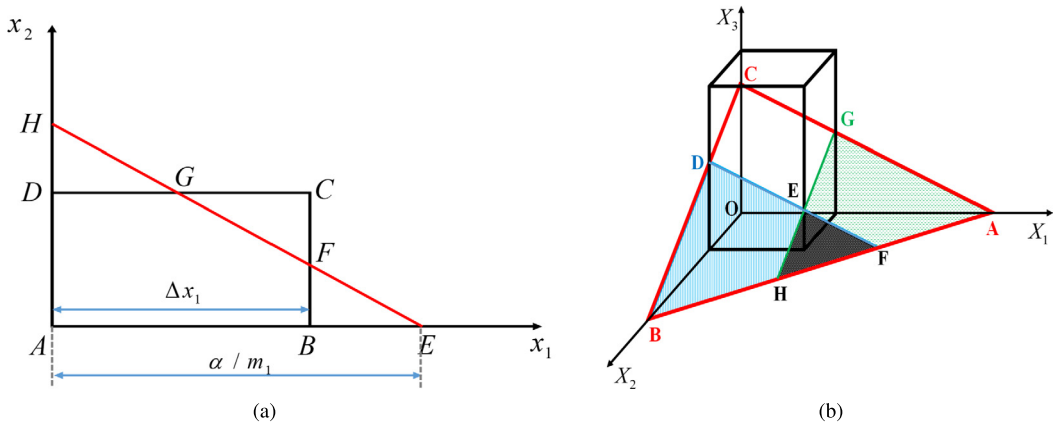


Fig. 4. Schematic diagram of the calculation of the interface area inside the computational cell cut by the interface in (a) 2*D* and (b) 3*D* computational domains. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Based on the concept of Gueyffier et al. [45], and Scardovelli and Zaleski [46], we develop an algorithm to find the interfacial area S_Γ in the CLSVOF framework. For simplicity, we start from considering a specific 2*D* case as shown in Fig. 4(a). Assuming that a unitary computational cell $ABCD$ with sides $\Delta x_i = 1$ $(i = 1, 2)$ is cut by the interface EH with normal vector $\mathbf{m} = (m_1, m_2)$. The origin of the Cartesian coordinate is located at A. The interface is represented by the following equation

$$m_1 x_1 + m_2 x_2 = \alpha, \tag{30}$$

where α is the distance from the origin to the interface. We normalize Eq. (30) to ensure $\alpha = m_1 + m_2$ (the values of α, m_1 and m_2 are computed after the interface is advanced). The points at which the interface intersects with x_1 and x_2 are α/m_1 and α/m_2, respectively. The interface length S_Γ inside the computational cell is given by

$$S_\Gamma = \sqrt{\left(\frac{\alpha}{m_1}\right)^2 + \left(\frac{\alpha}{m_2}\right)^2}\left[1 - H(\alpha - m_1\Delta x_1)\frac{\alpha - m_1\Delta x_1}{\alpha} - H(\alpha - m_2\Delta x_2)\frac{\alpha - m_2\Delta x_2}{\alpha}\right], \tag{31}$$

where $\sqrt{(\alpha/m_1)^2 + (\alpha/m_2)^2}$ is the length of EH. Since the triangles AEH, BEF and DGH are geometrically similar, the lengths of EF and GH can be computed by multiplying the length of EH with the ratio of the sides BE to AE and DH to AH, respectively. This is shown in the last two terms within the square brackets on the right-hand side of Eq. (31). The Heaviside function is adopted to ensure that the length of EF (or GH) is subtracted from EH when E (or H) is located outside the computational cell. This method can be extended to the calculation of the interfacial area S_Γ in a 3*D* domain. We skip the derivation and provide the formula as follows,

$$\begin{aligned} S_\Gamma = & \frac{\alpha^2\sqrt{(m_1m_2)^{-2} + (m_1m_3)^{-2} + (m_2m_3)^{-2}}}{2} \\ & \times \left[1 - \sum_{i=1}^{3} H(\alpha - m_i\Delta x_i)\left(\frac{\alpha - m_i\Delta x_i}{\alpha}\right)^2\right. \\ & + H(\alpha - m_1\Delta x_1 - m_2\Delta x_2)\left(\frac{\alpha - m_1\Delta x_1 - m_2\Delta x_2}{\alpha}\right)^2 \\ & + H(\alpha - m_1\Delta x_1 - m_3\Delta x_3)\left(\frac{\alpha - m_1\Delta x_1 - m_3\Delta x_3}{\alpha}\right)^2 \\ & \left. + H(\alpha - m_2\Delta x_2 - m_3\Delta x_3)\left(\frac{\alpha - m_2\Delta x_2 - m_3\Delta x_3}{\alpha}\right)^2\right], \end{aligned} \tag{32}$$

where $\alpha^2\sqrt{(m_1m_2)^{-2} + (m_1m_3)^{-2} + (m_2m_3)^{-2}}/2$ represents the area of triangle ABC in Fig. 4(b). The negative terms inside the square brackets are associated with the areas outside the grid cell. For example, triangles BDF and AGH are located outside the cell and should be subtracted from the area of triangle ABC. However, the area of triangle EFH marked by black is subtracted twice and we need to add it back. Thus, the last three positive terms within the square brackets account for the areas which have been subtracted twice. The relations in Eqs. (31) and (32) are applicable for all of the possibilities during the simulation, though we only show two specific cases in Fig. 4. We note that a similar algorithm was proposed

earlier by Soh et al. [47], while the implementation of our algorithm is more straightforward in the present numerical framework.

2.5. Temporal and spatial discretization

A second-order Runge-Kutta (RK2) scheme is used to advance the rotational velocity $\bar{\mathbf{u}}$, temperature θ and vapor mass fraction Y from time step n to $n+1$, and the finite difference scheme is used for spatial discretization. The flow quantities are defined on a staggered Cartesian grid to avoid checkerboard pressure field [48,49]. All scalars are discretized at cell centers, while the velocity components are discretized at cell faces. A third-order total variation diminishing (TVD) scheme and a second-order central difference scheme are employed to compute the advection term and diffusion term, respectively. For more details of the flow solver, we refer the readers to Yang et al. [1]. Here, we only introduce the spatial discretization of the last two derived terms in Eq. (18). Taking the calculation of $\nabla\cdot\left(\rho\bar{u}\tilde{\mathbf{u}}\right)_{i+1/2,j,k}$ and $\left[\left(\rho\bar{\mathbf{u}}\cdot\nabla\right)\tilde{u}\right]_{i+1/2,j,k}$ as examples, the discretization is given by

$$\begin{aligned}-\nabla\cdot\left(\rho\bar{u}\tilde{\mathbf{u}}\right)_{i+1/2,j,k} = &-\frac{\rho'_{i+1,j,k}\bar{u}'_{i+1,j,k}\tilde{u}_{i+1,j,k}-\rho'_{i,j,k}\bar{u}'_{i,j,k}\tilde{u}_{i,j,k}}{\Delta x}\\ &-\frac{\rho'_{i+1/2,j+1/2,k}\bar{u}'_{i+1/2,j+1/2,k}\tilde{v}_{i+1/2,j+1/2,k}-\rho'_{i+1/2,j-1/2,k}\bar{u}'_{i+1/2,j-1/2,k}\tilde{v}_{i+1/2,j-1/2,k}}{\Delta y}\\ &-\frac{\rho'_{i+1/2,j,k+1/2}\bar{u}'_{i+1/2,j,k+1/2}\tilde{w}_{i+1/2,j,k+1/2}-\rho'_{i+1/2,j,k-1/2}\bar{u}'_{i+1/2,j,k-1/2}\tilde{w}_{i+1/2,j,k-1/2}}{\Delta z},\end{aligned} \tag{33}$$

and

$$\begin{aligned}-\left[\left(\rho\bar{\mathbf{u}}\cdot\nabla\right)\tilde{u}\right]_{i+1/2,j,k} = &-\rho_{i+1/2,j,k}\bar{u}_{i+1/2,j,k}\frac{\tilde{u}'_{i+1,j,k}-\tilde{u}'_{i,j,k}}{\Delta x}\\ &-\rho_{i+1/2,j,k}\bar{v}_{i+1/2,j,k}\frac{\tilde{u}'_{i+1/2,j+1/2,k}-\tilde{u}'_{i+1/2,j-1/2,k}}{\Delta y}\\ &-\rho_{i+1/2,j,k}\bar{w}_{i+1/2,j,k}\frac{\tilde{u}'_{i+1/2,j,k+1/2}-\tilde{u}'_{i+1/2,j,k-1/2}}{\Delta z},\end{aligned} \tag{34}$$

respectively. The prime denotes a quantity calculated using the CUI scheme [50], while the advection velocities are calculated using linear central interpolation scheme. The discretization of the derived terms in other directions is similar to that in the x-direction with certain replacement in the indices.

The numerical implementations for the temperature θ and vapor mass fraction Y are also crucial for accurate simulations of phase change problems [26,37,35,31,36]. In the proposed method, the transport equations of temperature and mass fraction are evolved using the RK2 method as

$$\left(\rho c_p\theta\right)^l = \left(\rho c_p\theta\right)^{l-1} + \left(\alpha^l G^{l-1} - \beta^l G^{l-2}\right)\Delta t, \tag{35}$$

$$(Y)^l = (Y)^{l-1} + \left(\alpha^l H^{l-1} - \beta^l H^{l-2}\right)\Delta t, \tag{36}$$

where $l=1,2$ corresponds to the two substeps of RK2 method. The values of α^l and β^l are given as $\alpha^1=1$ and $\beta^1=0$, and $\alpha^2=\beta^2=0.5$. In Eqs. (35) and (36), G and H are the summation of advection and diffusion terms for temperature and vapor mass fraction equations, respectively, viz,

$$G = -\nabla\cdot\left(\rho c_p\theta\mathbf{u}\right) + \nabla\cdot k\nabla\theta - \left[h_{lg} + \left(c_{p,v}-c_{p,l}\right)\left(\theta_{sat}-\theta_\Gamma\right)\right]\frac{\dot{M}S_\Gamma}{V_\Gamma}, \tag{37}$$

$$H = -\nabla\cdot\left(Y\mathbf{u}\right) + \lambda\nabla^2 Y. \tag{38}$$

The temperature equation is evolved using a consistent scheme, the advection and diffusion terms are discretized using a third-order CUI scheme and a second-order central difference scheme, respectively. We refer the readers to Lu et al. [51] for more details on the temperature solver. Here, we describe the numerical method for solving the vapor mass fraction equation. The advection term of Eq. (4) is discretized using a first-order upwind scheme [22] as

$$\begin{aligned}-\left[\nabla\cdot\left(Y\mathbf{u}\right)\right]_{i,j,k} = &-\frac{Y_{i+1/2,j,k}u_{i+1/2,j,k}-Y_{i-1/2,j,k}u_{i-1/2,j,k}}{\Delta x}\\ &-\frac{Y_{i,j+1/2,k}v_{i,j+1/2,k}-Y_{i,j-1/2,k}v_{i,j-1/2,k}}{\Delta y}\\ &-\frac{Y_{i,j,k+1/2}w_{i,j,k+1/2}-Y_{i,j,k-1/2}w_{i,j,k-1/2}}{\Delta z},\end{aligned} \tag{39}$$

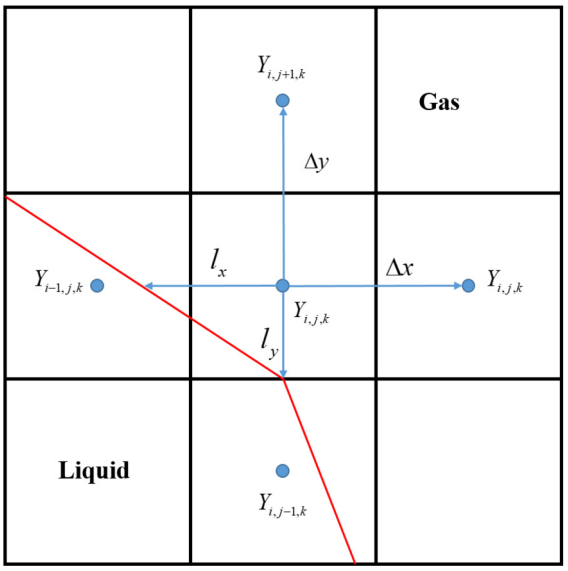


Fig. 5. Stencil points for calculating the derivation of vapor mass fraction $\partial Y/\partial x_i$ near the interface when the Dirichlet boundary condition is applied at the interface. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

where the vapor mass fraction at the cell faces is chosen according to the velocity there. For example, $-\partial(Yu)/\partial x$ is calculated as

$$-\left[\frac{\partial(Yu)}{\partial x}\right]_{i,j,k} = \begin{cases} -\left(Y_{i,j,k}u_{i+1/2,j,k} - Y_{i-1,j,k}u_{i-1/2,j,k}\right)/\Delta x & u_{i-1/2} \geq 0 \\ -\left(Y_{i+1,j,k}u_{i+1/2,j,k} - Y_{i,j,k}u_{i-1/2,j,k}\right)/\Delta x & u_{i-1/2} < 0 \end{cases}. \tag{40}$$

However, if the interface is located between i and $i+1$ as shown in Fig. 5, the calculation of $-\partial(Yu)/\partial x$ becomes

$$-\left[\frac{\partial(Yu)}{\partial x}\right]_{i,j,k} = \begin{cases} -\left(Y_{i,j,k}u_{i+1/2,j,k} - Y_{\Gamma}u_{i-1/2,j,k}\right)/l_x & u_{i-1/2} \geq 0,\ \phi_{i,j,k}\phi_{i-1,j,k} < 0 \\ -\left(Y_{i+1,j,k}u_{i+1/2,j,k} - Y_{i,j,k}u_{i-1/2,j,k}\right)/\Delta x & u_{i-1/2} < 0,\ \phi_{i,j,k}\phi_{i-1,j,k} \geq 0 \end{cases}, \tag{41}$$

where l_x is estimated using a height function as

$$l_x = \frac{\left|\phi_{i,j,k}\right|}{\left|\phi_{i-1,j,k}\right| + \left|\phi_{i,j,k}\right|}. \tag{42}$$

As the interface approaches to the grid point (i, j, k), l_x becomes extremely small. To ensure the robustness of the solver, we set $l_x = 0.1\Delta x$ when $l_x \leq 0.1\Delta x$, and the numerical results show that this approximation is reasonable. The diffusion term of Eq. (4) is discretized using a second-order scheme for non-uniform grid spacing [52,53,18,22,31]. Taking the discretization in the x-direction as an example, the value of $\left(\partial^2 Y/\partial x^2\right)_{i,j,k}$ is calculated as

$$\left(\frac{\partial^2 Y}{\partial x^2}\right)_{i,j,k} = \begin{cases} \frac{2Y_{\Gamma}}{l_x(\Delta x + l_x)} - \frac{2Y_{i,j,k}}{\Delta x l_x} + \frac{2Y_{i+1,j,k}}{\Delta x(\Delta x + l_x)} & \phi_{i,j,k}\phi_{i-1,j,k} < 0 \\ \frac{2Y_{i,j,k}}{\Delta x(\Delta x + l_x)} - \frac{2Y_{i,j,k}}{\Delta x l_x} + \frac{2Y_{\Gamma}}{l_x(\Delta x + l_x)} & \phi_{i,j,k}\phi_{i+1,j,k} < 0 \\ \frac{Y_{i+1,j,k} + Y_{i-1,j,k} - 2Y_{i,j,k}}{\Delta x^2} & \text{others} \end{cases}. \tag{43}$$

The implementation for other directions can be deduced accordingly.

We note that in Sec. 3.2, the phase change takes place at the saturated temperature and thus the temperature at the interface is fixed at θ_{sat} as a Dirichlet boundary condition for the air phase. In this situation, the temperature inside the air phase is solved using the same algorithm as that for the vapor mass fraction.

2.6. Boundary conditions and independent non-dimensional parameters

In the proposed method, the CFD solver only accounts for the rotational-part velocity $\tilde{\mathbf{u}}$, while the boundary conditions are satisfied by the total velocity $\mathbf{u}$ ($=\bar{\mathbf{u}} + \tilde{\mathbf{u}}$). Thus, the boundary condition for $\tilde{\mathbf{u}}$ should be derived based on the boundary conditions of $\mathbf{u}$. For example, due to the volume change, the outflow condition is adopted at some domain boundaries for phase-change problems. For a conventional CFD solver, the outflow condition at the right domain boundary for u at $i = nx + 1/2$ is given as

$$u_{nx+3/2,j,k} = u_{nx-1/2,j,k}. \tag{44}$$

We note that the commonly used "continuative approximation" is adopted here to implement the outflow condition. The numerical results in the following sections show that this approximation is sufficiently accurate for the problems in our study. The improvement of the outflow condition is not the focus of present work, and we refer the reads to Dong et al. [54] for more details.

To derive the boundary conditions for $\bar{u}$ when the outflow condition is employed for u, we substitute Eq. (10) into Eq. (44) to yield

$$\bar{u}_{nx+3/2,j,k} = \bar{u}_{nx-1/2,j,k} + \tilde{u}_{nx-1/2,j,k} - \tilde{u}_{nx+3/2,j,k}, \tag{45}$$

where $\tilde{u}_{nx+3/2,j,k}$ is obtained from the outflow condition of the potential velocity $\tilde{u}$, which is given at $i = nx + 1/2$ as

$$\tilde{u}_{nx+3/2,j,k} = \tilde{u}_{nx-1/2,j,k}. \tag{46}$$

Similarly, the boundary condition for interfacial velocity u_Γ at $i = nx + 1/2$ is given as

$$u_{\Gamma nx+3/2,j,k} = u_{\Gamma nx-1/2,j,k}. \tag{47}$$

Substituting Eq. (19) into Eq. (47) yields

$$\hat{u}_{nx+3/2,j,k} = \hat{u}_{nx-1/2,j,k} + \bar{u}_{nx-1/2,j,k} - \bar{u}_{nx+3/2,j,k}, \tag{48}$$

where $\bar{u}_{nx+3/2,j,k}$ is obtained from the boundary condition of $\bar{\mathbf{u}}$. Other types of boundary condition can be derived accordingly.

The relevant non-dimensional parameters for the study are given as

$$\begin{aligned} Re &= \frac{\rho_g u_{ref} l_{ref}}{\mu_g}, \quad We = \frac{\rho_g u_{ref}^2 l_{ref}}{\sigma}, \quad Fr = \frac{u_{ref}}{\sqrt{g l_{ref}}}, \\ Sc &= \frac{\mu_g}{\rho_g \lambda}, \quad Pr = \frac{\mu c_P}{k}, \quad St = \frac{c_{p,g}(\theta_\infty - \theta_{sat})}{h_{lg}}, \quad \gamma_\eta = \frac{\eta_l}{\eta_g}, \end{aligned} \tag{49}$$

where Re, We, Fr, Sc, Pr and St are the Reynolds number, Weber number, Froude number, Schmidt number, Prandtl number, and Stefan number, respectively, σ is the surface tension coefficient, u_{ref} and l_{ref} are characteristic velocity and length scale, respectively, and γ_η is the ratio of physical property η of liquid and gas phases.

2.7. Overall procedure

The overall procedure of the proposed algorithm is summarized as follows:

Step 1. Initialize the velocity $\mathbf{u}$, temperature θ, vapor mass fraction Y and the physical properties η.

Step 2. Calculate the interfacial mass transfer rate $\dot{M}$ and interface area S_Γ using Eqs. (5), (6), (31) and (32).

Step 3. Calculate the potential velocity $\tilde{\mathbf{u}}$ associated with the phase change by Eq. (13), the rotational-part velocity $\bar{\mathbf{u}}$ by the modified momentum equation Eq. (18), and the volume change velocity $\hat{\mathbf{u}}$ of liquid by Eq. (24).

Step 4. Solve the transport equations of temperature and vapor mass fraction using the total velocity $\mathbf{u}$ $(= \tilde{\mathbf{u}} + \bar{\mathbf{u}})$, and the transport equations of LS and VOF functions using the interfacial velocity $\mathbf{u}_\Gamma$ $(= \hat{\mathbf{u}} + \bar{\mathbf{u}})$.

Step 5. Update the physical properties according to the interface position by Eq. (25) and go to step 2.

The algorithm is also outlined in Fig. 6 for clarity.

3. Results

3.1. 2D droplet with constant evaporation rate

The numerical simulation of phase change is complicated due to the coupling of momentum, heat and mass transfer. Thus, before the validation of the fully coupled system, we first test the accuracy of the interface propagation by decoupling the effect of temperature θ and vapor mass fraction Y on the momentum equation. This is accomplished by prescribing a constant interfacial mass flux $\dot{M}$, and the transport equations of temperature and vapor mass fraction are not solved. The computational domain for this test case is depicted in Fig. 7. As shown, a $2D$ droplet is surrounded by gas. The density ratio between the liquid and gas is γ_ρ=8.333 and the gravity is neglected. The interfacial mass transfer rate is fixed at $\dot{M} = 1$. Thus, the droplet radius is expected to reduce steadily at a constant rate. The size of the computational domain is $4D_0$ in both x- and y-directions, where D_0 is the initial diameter of the droplet. The origin of the coordinates is located at the left bottom corner of the computational domain and the center of the droplet is located at $(x, y) = (2D, 2D)$. The outflow condition is applied at all boundaries. The computational domain is discretized using 256×256 number of grid points. Analytically, the diameter of the droplet as a function of time t is governed by the follow ordinary differential equation,

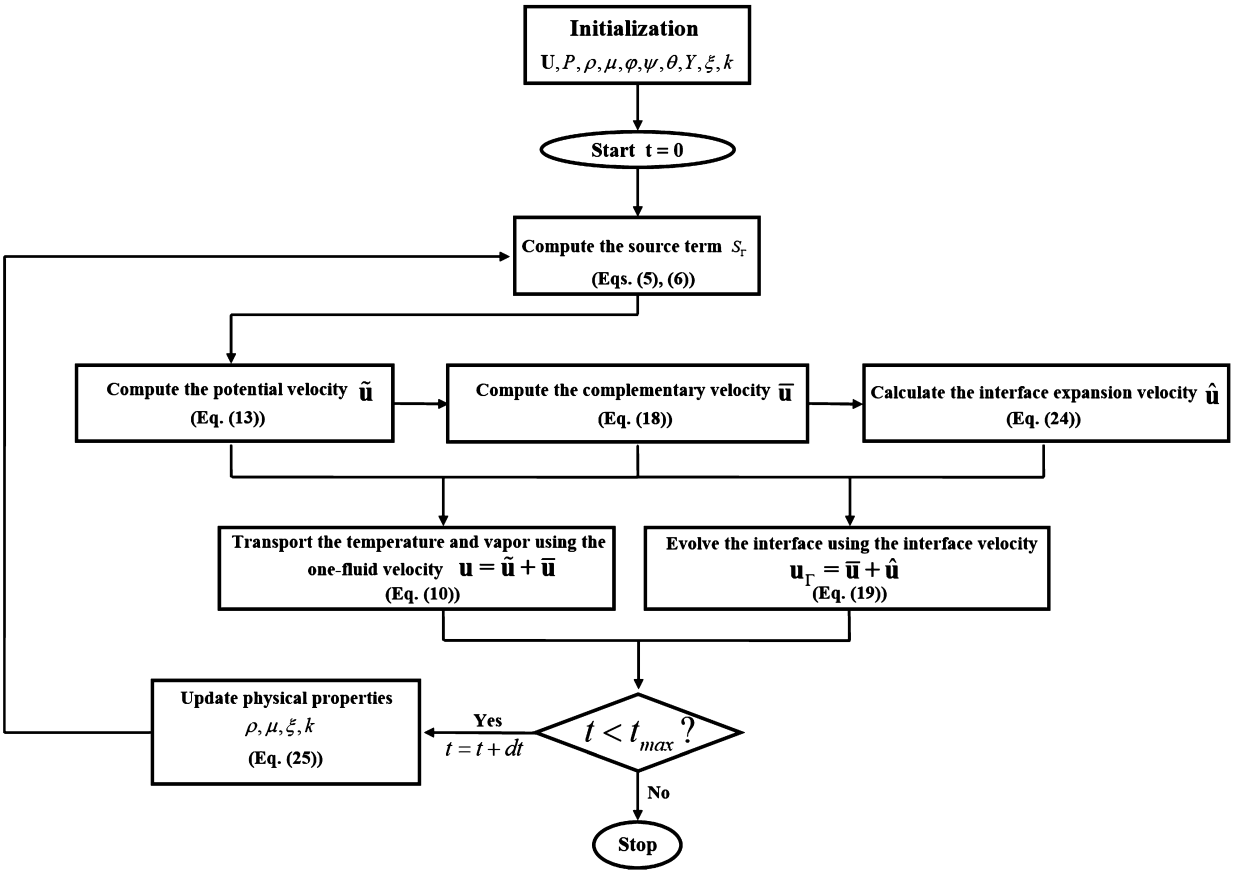


Fig. 6. Diagram illustration of the overall procedure of the proposed algorithm.

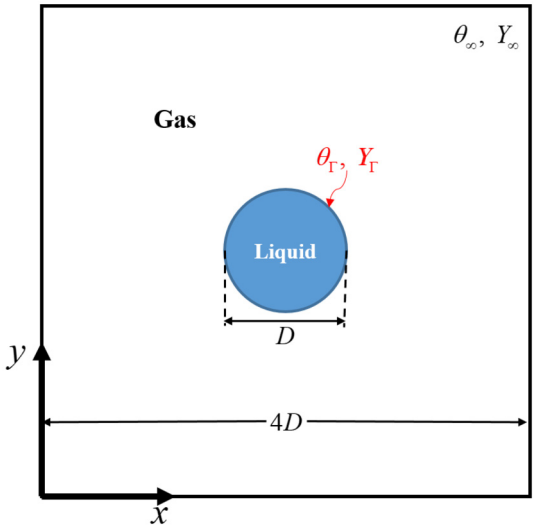


Fig. 7. Computational domain for the test case of evaporation of a $2D$ droplet.

$$\frac{D(t)}{D_0} = 1 - \left(\frac{2\dot{M}}{D_0\rho_l}\right)t. \tag{50}$$

In this test case, the theoretical value of the rotational-part velocity $\bar{\mathbf{u}}$ is zero, and the interface motion is solely determined by $\hat{\mathbf{u}}$ from Eq. (24). Here, we relocate the source point inside the liquid phase, thus $\hat{\mathbf{u}}$ is associated with the volume loss of liquid phase (i.e., $\rho' = \rho_v$) in this test case. Fig. 8 compares different parts of the decomposed velocity. It is seen from

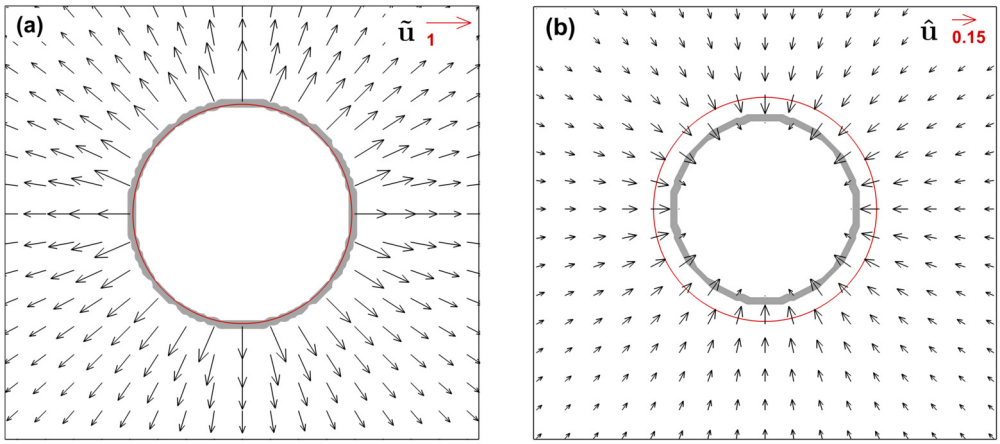


Fig. 8. The decomposed velocity vectors at $t = 0.1T$ for the test case of a $2D$ droplet with a constant evaporation rate. (a) The velocity vectors of $\tilde{\mathbf{u}}$ induced by the source term on the interface; (b) The reconstructed interface regression velocity vectors of $\hat{\mathbf{u}}$ induced by the rearranged source points inside the droplet. The solid red line represents the interface, and the gray band shows the location of source points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Fig. 8(a) that the Stephan flow velocity $\tilde{\mathbf{u}}$ (Eq. (13)) induced by the evaporation is discontinuous at the interface due to the existence of the source points there. For a conventional CFD solver, the interface is then transported using the total velocity $\mathbf{u}$ ($=\tilde{\mathbf{u}}+\bar{\mathbf{u}}$), in which the rotational part $\bar{\mathbf{u}}$ is 0 and such that $\mathbf{u}=\tilde{\mathbf{u}}$. As a result, the total velocity $\mathbf{u}$ is discontinuous. If the interface is transported directly using $\mathbf{u}$, the volume of the droplet is enlarged in an unexpected manner. Thus, to evolve the interface accurately, we calculate a regression velocity $\hat{\mathbf{u}}$ of the interface by Eq. (24). Because the source points are reallocated inside the interface in the liquid phase based on the mass loss of liquid, $\hat{\mathbf{u}}$ is continuous across the liquid-gas interface. The interface is then advanced using the interface velocity $\mathbf{u}_\Gamma$ ($=\hat{\mathbf{u}}+\bar{\mathbf{u}}$). The regression velocity field is displayed in Fig. 8(b). Although there exist discontinuities in $\hat{\mathbf{u}}$ inside the liquid phase, it is proper to be used to evolve the interface because the evolution of the LS and VOF functions is only determined by $\hat{\mathbf{u}}$ near the interface.

As noted in Sec. 2.3, the interface regression velocity $\hat{\mathbf{u}}$ can also be calculated by Eq. (20) as applied in the algebraical VOF framework of Scapin et al. [36]. To test the feasibility of this method in the CLSVOF framework, we use Eq. (20) to calculate the regression velocity of interface at the cell center and apply this velocity onto the corresponding cell faces. The results are shown in Fig. 9. It is evident that the interface geometry is more reasonable when $\hat{\mathbf{u}}$ is computed by Eq. (24), while the circular shape is not well preserved when $\hat{\mathbf{u}}$ is calculated using Eq. (20). This is because the source points exist sharply in the interface cells in our numerical framework as shown in Fig. 8(a). Thus, Eq. (20) does not give a continuous divergence-free regression velocity $\hat{\mathbf{u}}$ around the interface cells. The velocity extrapolation schemes proposed by Tanguy et al. [27] and Palmore Jr and Desjardins [30] have the potential to address this issue. In their method, the extrapolated velocity also needs to be projected onto its divergence-free part through iteration as done in Eq. (24) to ensure the mass conservation. Another mass conservative approach to evolve the interface is proposed by Malan et al. [35] in the geometrical VOF framework. In their method, the rotational-part velocity $\bar{\mathbf{u}}$ is constructed to evolve the VOF function. Because $\bar{\mathbf{u}}$ is divergence-free, the volume change is transferred to a source term of the VOF function (i.e., $\partial\psi/\partial t+\nabla\cdot(\psi\bar{\mathbf{u}})=\dot{M}S_\Gamma/V_\Gamma/\rho'$) to ensure the mass conservation for phase-change problems. However, this approach cannot be applied to the CLSVOF method because the level-set function ϕ is not a conservative quantity. Therefore, in the present study, equation (24) is employed to calculate $\hat{\mathbf{u}}$ for evolving the VOF and level-set functions.

To further demonstrate the accuracy of the proposed method, we conduct the simulation using different number of grid points ranging from 32×32 to 256×256 and compare the evolution of droplet diameter with the analytical solution in Fig. 10(a). It is seen that the numerical results convergence to the analytical solution as the grid is refined. To quantify the numerical accuracy, we define a numerical error E of the diameter as

$$E=\frac{\left|D_s^t-D_a^t\right|}{D_a^t}, \tag{51}$$

where the subscripts s and a denote simulation and analytical results, respectively. Fig. 10(b) depicts the numerical error at $t=2T$ and $t=3T$. It is observed that the accuracy of the numerical solution is close to the second order, which remains the same as the accuracy of the interface capturing scheme [42].

For comparison purpose, we also implement the numerical method proposed in the geometric VOF framework by Malan et al. [35] into our CLSVOF framework. In their method, the governing equation for the phase change is also given by Eqs. (1) and (2), and the governing equation of total velocity $\mathbf{u}$ is solved. To construct a divergence-free velocity $\bar{\mathbf{u}}$, they compute a volume change velocity, which accounts for the anti-effect of phase change. The anti-effect velocity $-\tilde{\mathbf{u}}$ is obtained from solving an additional Poisson equation. By adding $-\tilde{\mathbf{u}}$ to the total velocity $\mathbf{u}$, the divergence-free velocity $\bar{\mathbf{u}}$ ($=\mathbf{u}-\tilde{\mathbf{u}}$) for

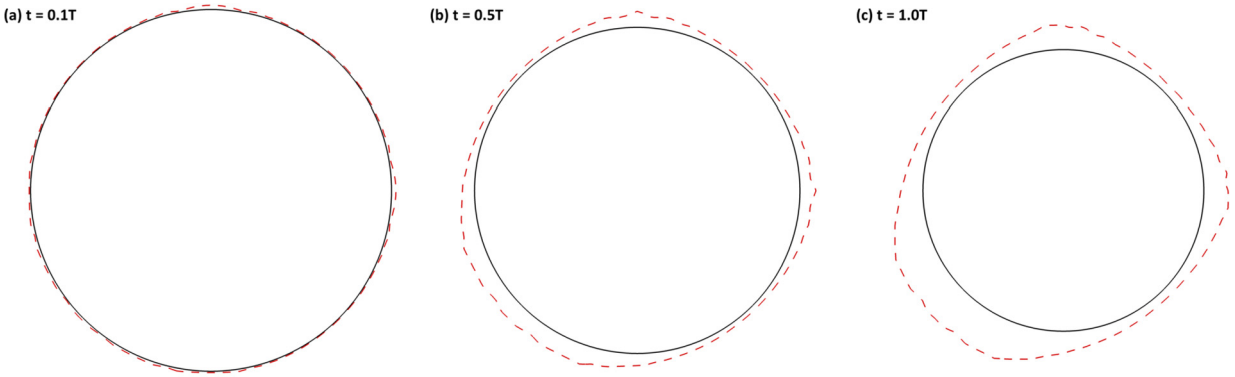


Fig. 9. Instantaneous droplet geometry at (a) $t = 0.1T$, (b) $t = 0.5T$, and (c) $t = 1.0T$ by using Eq. (20) (red dash line) and Eq. (24) (black solid line) to compute the interfacial velocity $\mathbf{u}_\Gamma$, respectively. The interfacial mass transfer rate is prescribed to $\dot{M} = 1$. The domain size is $4D_0 \times 4D_0$, discretized using 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

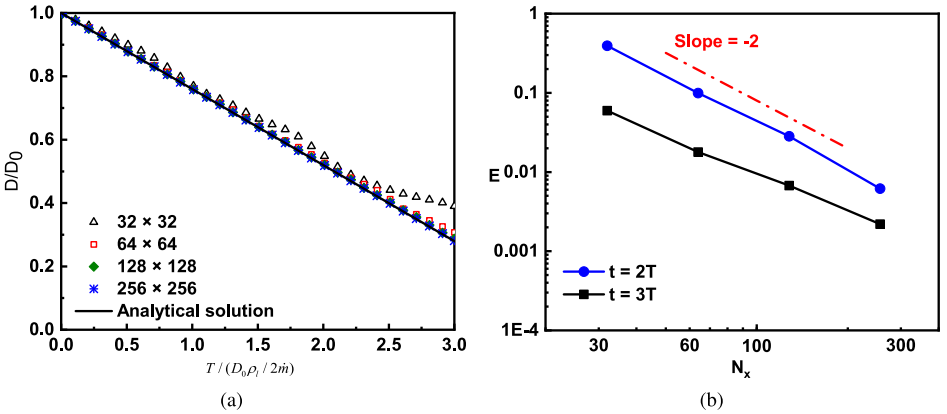


Fig. 10. Numerical results for the test case of a $2D$ droplet with constant evaporation rate using different number of grid points. (a) Temporal evolution of the droplet diameter. (b) The numerical error E as function of the number of grid points. The blue dots line and black squares line show the results at $t = 2T$ and $3T$, respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

the whole domain is found. To evolve the interface, the loss of liquid volume is also transferred into a interface regression velocity $\hat{\mathbf{u}}$ by Eq. (24). Thus, the interface can be transported by the interfacial velocity $\mathbf{u}_\Gamma$ ($=\bar{\mathbf{u}}+\hat{\mathbf{u}}$). For notational simplicity, we name this numerical strategy as $S2$ and the numerical strategy proposed in Sec. 2 as $S1$. For further illustration, a schematic diagram for these two strategies is shown in Fig. 11. It is seen that the main difference between these two strategies lies in the solver for the momentum equation. In the proposed strategy $S1$, the momentum equations are solved for the rotational part $\bar{\mathbf{u}}$, which is divergence-free with $\nabla \cdot \bar{\mathbf{u}} = 0$. In strategy $S2$, the momentum equations are solved for the total velocity $\mathbf{u}$, of which the divergence is determined by the mass change rate as $\nabla \cdot \mathbf{u} = (1/\rho_v - 1/\rho_l)\dot{M}S_\Gamma/V_\Gamma$.

Fig. 12(a) compares the interface geometry obtained from $S1$ and $S2$. It is seen that the interface is distorted in an unphysical manner at $t = 0.1T$ when $S2$ is adopted. On the contrary, the interface can evolve precisely by applying $S1$. The reason for the unsatisfactory interface propagation of $S2$ can be found by looking into the rotational-part velocity $\bar{\mathbf{u}}$, of which the vectors obtained from the two strategies are compared in Fig. 12(b). Theoretically, the rotational-part velocity $\bar{\mathbf{u}}_{\text{theo}}$ for this specific test is zero since the flow is only driven by the phase change and this is satisfied by $S1$ as shown in the left half of Fig. 12(b). However, spurious velocity is observed in the right half of Fig. 12(b) when $S2$ is employed. This is because if the total velocity $\mathbf{u}$ is evolved directly without applying the decomposition, its numerical error $E_{\mathbf{u}}$ consists of the errors in both rotational and potential parts, which have nonlinear interactions with each other. As a result, the subtraction of $\tilde{\mathbf{u}}$ from the total velocity $\mathbf{u}$ does not fully eliminate the numerical error, i.e.,

$$\bar{\mathbf{u}}_{\text{num}} = (\mathbf{u} + E_{\mathbf{u}}) - (\tilde{\mathbf{u}} + E_{\tilde{\mathbf{u}}}) = \bar{\mathbf{u}}_{\text{theo}} + (E_{\mathbf{u}} - E_{\tilde{\mathbf{u}}}) \neq \bar{\mathbf{u}}_{\text{theo}}, \tag{52}$$

because $(E_{\mathbf{u}} - E_{\tilde{\mathbf{u}}}) \neq 0$. The subscripts 'num' and 'theo' denote the numerical and theoretical values, respectively. This leads to the spurious velocity in $\bar{\mathbf{u}}$. The numerical error in $\bar{\mathbf{u}}$ accumulates with simulation time and the level-set function is then transported in an unphysical manner when the magnitude in $\bar{\mathbf{u}}$ reaches a certain level. As a result, the interface propagates inaccurately and in return amplifies the numerical error. This observation is reasonable because the normal direction of the interface is acquired from the level-set function in our CLSVOF framework and the error in the interface

	step 1	step 2	step 3	step 4
$S1$	$\begin{cases}\nabla\cdot\left(\frac{1}{\rho}\nabla\varphi\right)=\left(\frac{1}{\rho_v}-\frac{1}{\rho_l}\right)\frac{\dot{M}S_\Gamma}{V_{cell}}\\ \tilde{\mathbf{u}}=\frac{1}{\rho}\nabla\varphi\end{cases}$	$\begin{cases}\frac{D(\rho\bar{\mathbf{u}})}{Dt}=-\nabla\bar{p}+\nabla\cdot 2\mu\bar{S}+\rho\mathbf{g}+\mathbf{f}_s\\ \qquad\qquad +\nabla\cdot(\rho\bar{\mathbf{u}}\tilde{\mathbf{u}})-(\rho\bar{\mathbf{u}}\cdot\nabla)\tilde{\mathbf{u}}\\ \nabla\cdot\bar{\mathbf{u}}=0\end{cases}$	$\begin{cases}\nabla\cdot\nabla\varphi'=\frac{-\dot{M}S_\Gamma}{V_{cell}}\\ \hat{\mathbf{u}}=\frac{1}{\rho'}\nabla\varphi'\end{cases}$	$\mathbf{u}_\Gamma=\bar{\mathbf{u}}+\hat{\mathbf{u}}$
$S2$		$\begin{cases}\frac{D(\rho\mathbf{u})}{Dt}=-\nabla p+\nabla\cdot 2\mu S+\rho\mathbf{g}+\mathbf{f}_s\\ \nabla\cdot\mathbf{u}=\left(\frac{1}{\rho_v}-\frac{1}{\rho_l}\right)\frac{\dot{M}S_\Gamma}{V_{cell}}\end{cases}$		$\begin{cases}\bar{\mathbf{u}}=\mathbf{u}-\tilde{\mathbf{u}}\\ \mathbf{u}_\Gamma=\bar{\mathbf{u}}+\hat{\mathbf{u}}\end{cases}$

Fig. 11. Diagram illustration of the overall procedure using different numerical strategies. $S1$ represents the numerical method using the velocity decomposition method proposed in Sec. 2, while $S2$ employs the numerical algorithm proposed in the literature without velocity decomposition.

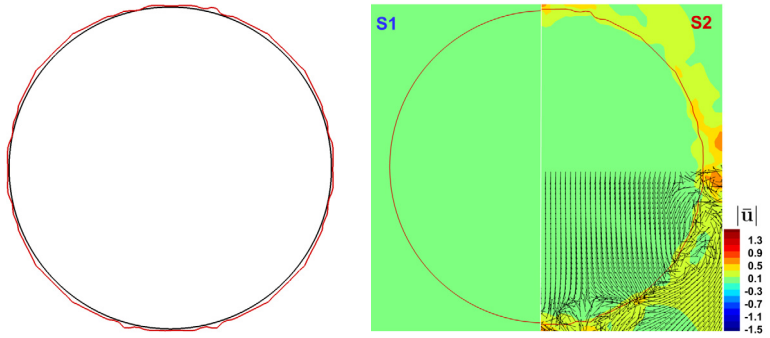


Fig. 12. Numerical results of $S1$ and $S2$ for the test case of $2D$ droplet with constant evaporation rate at $t=0.1T$. (a) The interface position of $S1$ (black line) and $S2$ (red line). (b) Contours of the norm of the rotational-part velocity $\tilde{\mathbf{u}}$ and the velocity vectors of $\bar{\mathbf{u}}$ given by $S1$ (left) and $S2$ (right). The velocity vectors are only shown in the lower half of the computational domain. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

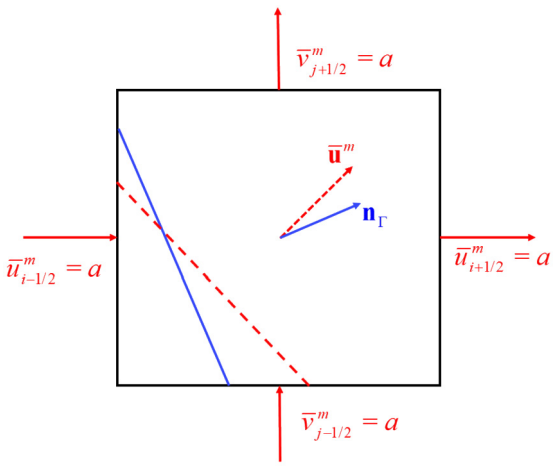


Fig. 13. Schematic diagram of the unphysical interface propagation induced by the numerical error in $\tilde{\mathbf{u}}$ within the CLSVOF framework when $S2$ is employed for the phase-change problem.

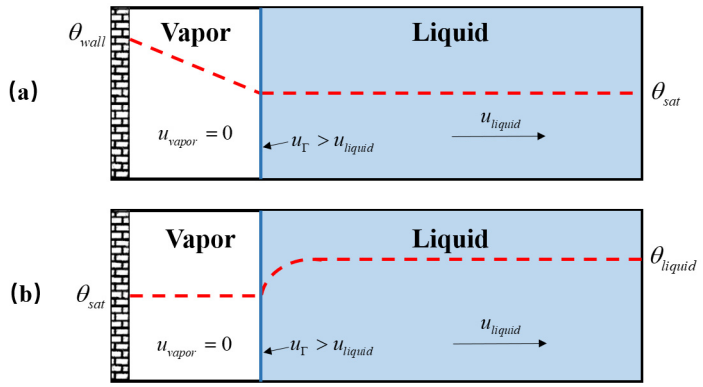


Fig. 14. Schematic diagram for the 1D (a) Stefan and (b) sucking problems. The solid wall is set for the left boundary while the liquid can leave freely from the right boundary. The vapor is generated at the interface and the saturation temperature θ_{sat} is fixed at the interface.

normal direction leads to the inaccurate interface propagation. A schematic diagram is shown in Fig. 13 to demonstrate this point. As shown, the interface should propagate along the normal direction if $\tilde{\mathbf{u}} = 0$. Nevertheless, due to the numerical error in the solution of the Poisson equation, the existing spurious velocity changes the normal direction of the interface. We note here that, unlike the CLSVOF framework, the shape of the interface is solely determined by VOF function in the geometric VOF framework. Although the level-set method can provide an accurate representation of the interface normal direction, the VOF method seems to be less sensitive to the error in the velocity direction, and as such the accurate interface representation is achieved by using the numerical method proposed by Malan et al. [35]. Therefore, in the framework of VOF method, the interface is more robust to the presence of spurious velocity. The advantages of different interface-capturing schemes are not the focus of the present study. The objective of present study is to develop an accurate phase-change model in the CLSVOF framework, and the results of this test case indicate that the velocity decomposition strategy is effective.

3.2. Phase change at saturation temperature

In this section, we further verify the coupling of temperature and momentum equations. The temperature at the interface θ_Γ is fixed at the saturation temperature θ_{sat}. Thus, the interfacial mass transfer rate $\dot{M}$ is computed by Eq. (5). The vapor mass fraction is not considered in this section.

3.2.1. The Stefan and sucking problem

The Stefan problem is a benchmark test case to validate the phase-change model at the saturation temperature θ_{sat}. The problem setup is depicted in Fig. 14(a). Initially, the liquid phase is at rest at the saturation temperature θ_{sat}. A vapor layer is located between the liquid phase and the left wall. The temperature of the interface is fixed at θ_{sat} and the temperature of left wall is fixed at θ_{wall} ($> \theta_{sat}$). The thermal diffusivity of gas and Stefan number are $\alpha_g = k_g/\rho_g c_{p,g} = 10$ and $St = 0.05$, respectively. The outflow condition is set at the right-side boundary. Due to the temperature gradient between the heated wall and the interface, phase change takes place, causing the expansion of the vapor layer. The liquid leaves the computational domain at the right-side boundary. The analytical solution of the interface location at an arbitrary time t is given as

$$X_\Gamma(t) = 2\beta\sqrt{\alpha_g t}, \tag{53}$$

where β is the solution of the following transcendental equation

$$\beta \exp(\beta^2)\mathrm{erf}(\beta) = \frac{St}{\sqrt{\pi}}. \tag{54}$$

The numerical results for different numbers of grid points are plotted in Fig. 15. It is seen that the numerical results agree with the analytical solution at all grid resolution under test.

In the above Stefan problem, only diffusion term of temperature equation is verified while the advection term is zero everywhere. To further examine the temperature solver, the sucking problem is tested. This case was previously studied by Welch and Wilson [20]. The setup of the sucking problem is shown in Fig. 14(b). The temperature of vapor and the left wall is fixed at θ_{sat} while the temperature of liquid θ_l is higher than the saturation temperature. The phase change takes place at the interface and the water is pushed out of the domain across the right-side boundary. The governing parameters for this problem are $St = 0.2$, $\alpha_g = 1$, $\alpha_l = 0.2$, $\gamma_k = 2$ and $\gamma_\rho = 10$. The analytical solution of the interface position versus time for this problem is also given by Eq. (53), while β is given by the solution of a different transcendental equation as

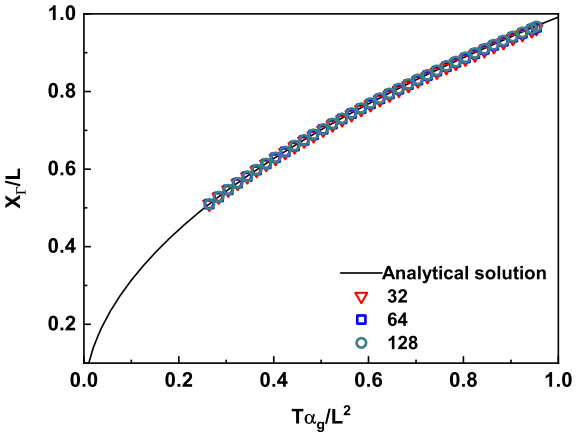


Fig. 15. Evolution of interface location for Stefan problem. The solid line represents the analytical solution. The thermal diffusivity of gas and Stefan number are $\alpha_g = 10$, $St = 0.05$, respectively. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

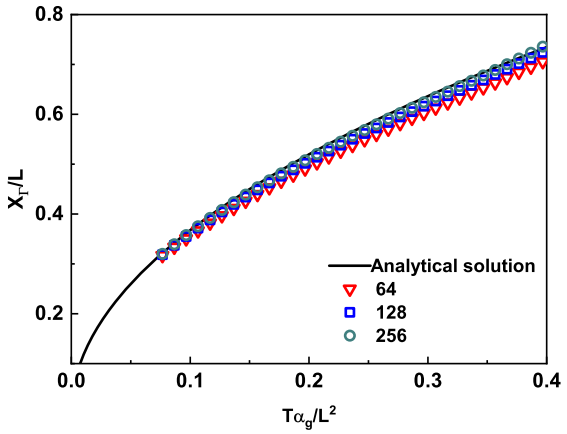


Fig. 16. The evolution of interface location for sucking problem. The black line is the analytical solution. The governing parameters are $St = 0.2$, $\alpha_g = 1$, $\alpha_l = 0.2$, $St = 0.2$, $\gamma_k = 2$, and $\gamma_\rho = 10$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

$$\exp(\beta^2)\mathrm{erf}(\beta)\left[\beta - \frac{St\gamma_k\sqrt{\alpha_g}\exp(-\beta^2\frac{\alpha_g}{{\gamma_\rho}^2\alpha_l})}{\sqrt{\pi\alpha_l}\mathrm{erfc}(\beta\frac{\sqrt{\alpha_g}}{\gamma_\rho\sqrt{\alpha_l}})}\right] = 0. \tag{55}$$

Fig. 16 compares the numerical results for different grid resolution with the analytical solution. It is seen that the numerical results converge to the analytical solution when the number of grid points approaches 256.

We note here that the numerical results obtained from $S1$ and $S2$ are close in the one-dimensional (1D) problems. This is because the orientation of the interface cannot be changed in 1D problems. Thus, both $S1$ and $S2$ can predict the evolution of the interface accurately, which indicates that the unsatisfactory results of $S2$ in Sec. 3.1 are mainly caused by the inaccurate prediction of the interface normal direction.

3.2.2. Evaporation of a 2D droplet at saturation temperature

Thus far, only 1D cases are tested for phase change at the saturation temperature θ_{sat}. To further verify the proposed phase-change model, we conduct a simulation of the evaporation of a $2D$ droplet. Initially, a droplet with the saturation temperature θ_{sat} is located at the centroid of the computational domain, surrounded by hot gas ($\theta_g > \theta_{sat}$). The phase change is triggered by the normal temperature gradient at the interface. As such, the droplet is expected to evaporate steadily and the temperature of the gas θ_g near the droplet decreases with time. The computational domain and boundary condition for this problem remains the same as those for the prescribed evaporation rate problem in Sec. 3.1. The thermal diffusivity of gas and the density ratio are $\alpha_g = 1$ and $\gamma_\rho = 10$, respectively. The temperature of the droplet is fixed to θ_{sat} and the temperature of the domain boundaries is θ_∞ ($> \theta_{sat}$). We conduct three cases, in which the latent heat h_{lg} varies, resulting in various values of the St number. The characteristic length scale and time scale are D_0 and ${D_0}^2/\alpha_g$, respectively. The simulation is conducted using 256×256 grid points.

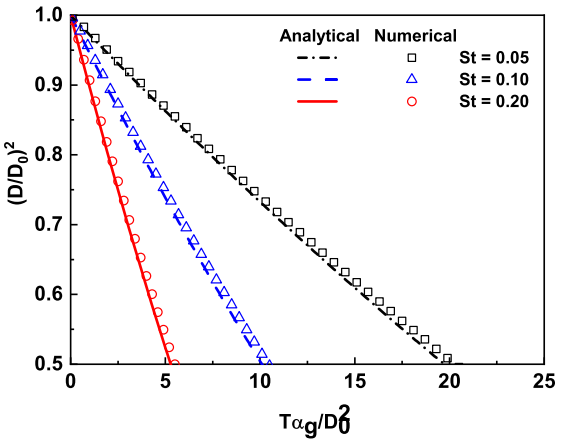


Fig. 17. Evolutions of drop diameter for different St number for the test case of $2D$ droplet evaporation at the saturation temperature θ_{sat}. The symbols are the results of the numerical simulation while the lines represent the analytical solutions. The governing parameters are $\alpha_g = 1$ and $\gamma_\rho = 10$. The computational domain size is $4D_0 \times 4D_0$, discretized evenly by 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

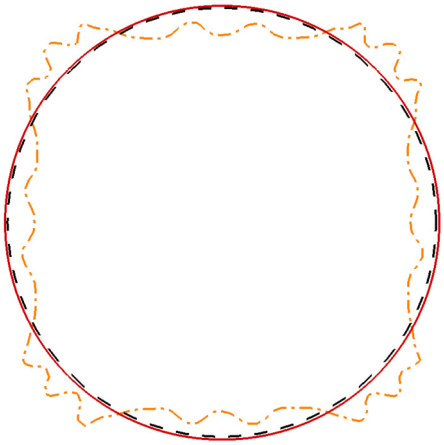


Fig. 18. The geometry of the droplet interface at t=0.4T for the test case of $2D$ droplet evaporation at the saturation temperature θ_{sat} using numerical strategies of $S1$ (red solid line), $S2$ (yellow dash dot line), and the analytical solution (black dashed line), respectively. The governing parameters are $St = 0.2$, $\alpha_g = 1$, and $\gamma_\rho = 10$. The computational domain is $4D_0 \times 4D_0$, discretized evenly by 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Fig. 17 shows the numerical results for different St numbers. It is observed that the drop diameter decreases faster at a higher St number. This observation follows the D^2 law and can be expressed analytically through the following equation

$$\frac{dD^2}{dt} = -\frac{8\alpha_g}{\gamma_\rho}\frac{\ln(1+St)}{\ln(L/\sqrt{d^2})}, \tag{56}$$

where L is the length of the computational domain. The analytical solutions for different St numbers are also depicted in Fig. 17. It is evident that the numerical results agree with the analytical solution. The results shown in Sec. 3.2 indicate that the proposed model is accurate for phase change at the saturation temperature θ_{sat} while the mass fraction of vapor is not coupled.

We have also tested the performance of strategy $S2$ on the problem of droplet evaporation at saturation temperature θ_{sat}. The interface geometry at $t = 0.4T$ is shown in Fig. 18. The numerical result based on $S1$ and the analytical result are superimposed for comparison. It is evident from the figure that $S2$ causes unphysical deformation of the droplet, while the result of $S1$ is close to the analytical solution. This test result indicates that $S1$ is more suitable than $S2$ for the present numerical framework.

3.3. Phase change below saturation temperature

In this section, we further consider the verification of the fully coupled system, in which the temperature is significantly lower than the saturation temperature θ_{sat}. The interfacial mass transfer rate $\dot{M}$ is mainly determined by the local gradient

Table 1
Physical properties of fluid system used in Sec. 3.3.1.

	ρ (kg/m^3)	μ (Pa·s)	c_p (J/kg K)	κ (W/m K)	M (kg/kmol)	h_{lg} (J/kg)	σ (N/m)	θ_{sat} (K)	λ (m^2/s)
Air	1.2	1.79×10^{-5}	1006	0.026	29.0	2.33×10^6	7.2×10^{-4}	373.15	2.33×10^{-5}
Liquid	10	1.14×10^{-3}	4186	0.006	18.0				

of the vapor mass fraction at the interface as given by Eq. (7). The phase change takes place when the gradient of the vapor mass fraction is non-zero. The heat in both liquid and air phases is absorbed by the interface as a compensation of phase change, and the temperature at the interface decreases. The decrease in temperature further influences the vapor mass fraction at the interface (see Eqs. (8) and (9)), and thus the momentum, heat and mass transfers are fully coupled.

3.3.1. Evaporation of a 2D droplet below saturation temperature

In this test case, the evaporation of a $2D$ liquid droplet below the saturation temperature θ_{sat} is simulated. The computational domain of this test case remains the same as that shown in Fig. 7, and $N_x \times N_y = 256 \times 256$ grid points are used for discretization. The Dirichlet boundary condition is set for both temperature θ and vapor mass fraction Y, and the outflow condition is employed for the velocity $\mathbf{u}$ at the four boundaries. Initially, the temperature of the entire computational domain is the dry bulb temperature $\theta_{db} = 313$ K. The temperature at the boundaries is also fixed at $\theta_\infty = 313$ K. The vapor mass fraction at the domain boundaries Y_∞ is a function of relative humidity of air Ψ and dry bulb temperature θ_{db}. The relative humidity is set to $\Psi = 50\%$ in this test case, resulting a vapor mass fraction of $Y_\infty = 2.28 \times 10^{-2}$ at the boundary. The vapor mass fraction of gas is set to $Y_g = 2.28 \times 10^{-2}$ for the initial condition. The vapor mass fraction at the interface Y_Γ varies according to the interface temperature θ_Γ. The specific value is computed by Eq. (8), and θ_Γ is set as the Dirichlet boundary condition at the interface using the method mentioned in Irfan and Muradoglu [18]. The vapor mass fraction in the liquid phase Y_l is not considered and thus imposes no impact on the simulation. The physical properties are listed in Table 1, which remain the same as the test cases by Irfan and Muradoglu [18] and Zhao et al. [37]. The length scale and time scale for this problem are D_0 and ${D_0}^2/\lambda$, respectively. The phase change takes place when the vapor mass fraction in the gas Y_g is lower than that at the interface Y_Γ. During the evaporation, the temperature at the interface θ_Γ decreases due to the heat loss associated with the phase change until the wet bulb temperature θ_{wb} is attained. Theoretically, the wet bulb temperature θ_{wb} is also a function of dry bulb temperature θ_{db} and the relative humidity in the air Ψ, and can be read from the psychrometric chart.

Fig. 19 shows the successive snapshots of the temperature and vapor mass fraction fields. It is seen that the vapor mass fraction Y decreases along with the decrease of the temperature θ. The temperature profile along the center line of the computational domain in the x-direction is shown in Fig. 20. It is seen that the temperature θ inside the droplet almost reaches the wet bulb temperature θ_{wb} at $t = 22T$, which is consistent with the results of Zhao et al. [37]. To further validate our solver, we compare the wet bulb temperature θ_{wb} under different conditions with the data read from the psychrometric chart in Fig. 21. It is seen that the numerical results agree with the data given by the chart. The test results shown in this section indicate that the numerical method proposed in Sec. 2 is also accurate for a fully coupled system.

We have also tested the capability of $S2$ for simulating this case. The temperature contours, interface geometry, and the velocity vectors at $t = 0.2T$ based on $S1$ and $S2$ are compared in Fig. 22. It is seen from Fig. 22(a) that the circular shape is better preserved by using $S1$, while the interface based on $S2$ is less satisfactory due to the spurious velocity of $\tilde{\mathbf{u}}$ shown in the right part of Fig. 22(b). Fig. 22(c) compares the velocity field of the Stephan flow. It is seen that the jump of $\tilde{\mathbf{u}}$ is well captured at the interface when $S1$ is employed, while the unsatisfactory interface geometry based on $S2$ leads to unexpected velocity vectors near the interface.

3.3.2. Falling of a 2D evaporating droplet under gravity

In this section, we consider a more general case in which the droplet is driven by the gravity during evaporation. Initially, the droplet is at rest and embedded in the dry air ($Y_g = 0$), and the temperature of the whole computational domain is set as dry bulb temperature θ_{db} ($= 371$ K). The droplet falls down and meanwhile, the phase change occurs due to the vapor mass fraction gradient between the droplet interface and the surrounding dry air. The heat of both liquid and air is absorbed by the interface as a compensation of the phase change, and as a result, the temperature around the interface decreases during the droplet falling. The size of the computational domain is $L_x \times L_y = 4D_0 \times 16D_0$, discretized using $N_x \times N_y = 128 \times 512$ grid points. To present the results, the origin of the coordinate is located at the left bottom corner of the computational domain, and the droplet is released at $(x, y) = (2D_0, 14.4D_0)$. The physical properties are listed in Table 2.

Table 2
Physical properties of fluid system used in Sec. 3.3.2.

	ρ (kg/m^3)	μ (Pa·s)	c_p (J/kg K)	κ (W/m K)	M (kg/kmol)	h_{lg} (J/kg)	σ (N/m)	θ_{sat} (K)	λ (m^2/s)
Air	2	3.88×10^{-6}	1006	0.026	29.0	2.4×10^6	1.0×10^{-5}	373.15	1.94×10^{-6}
Liquid	10	7.75×10^{-5}	4186	0.006	18.0				

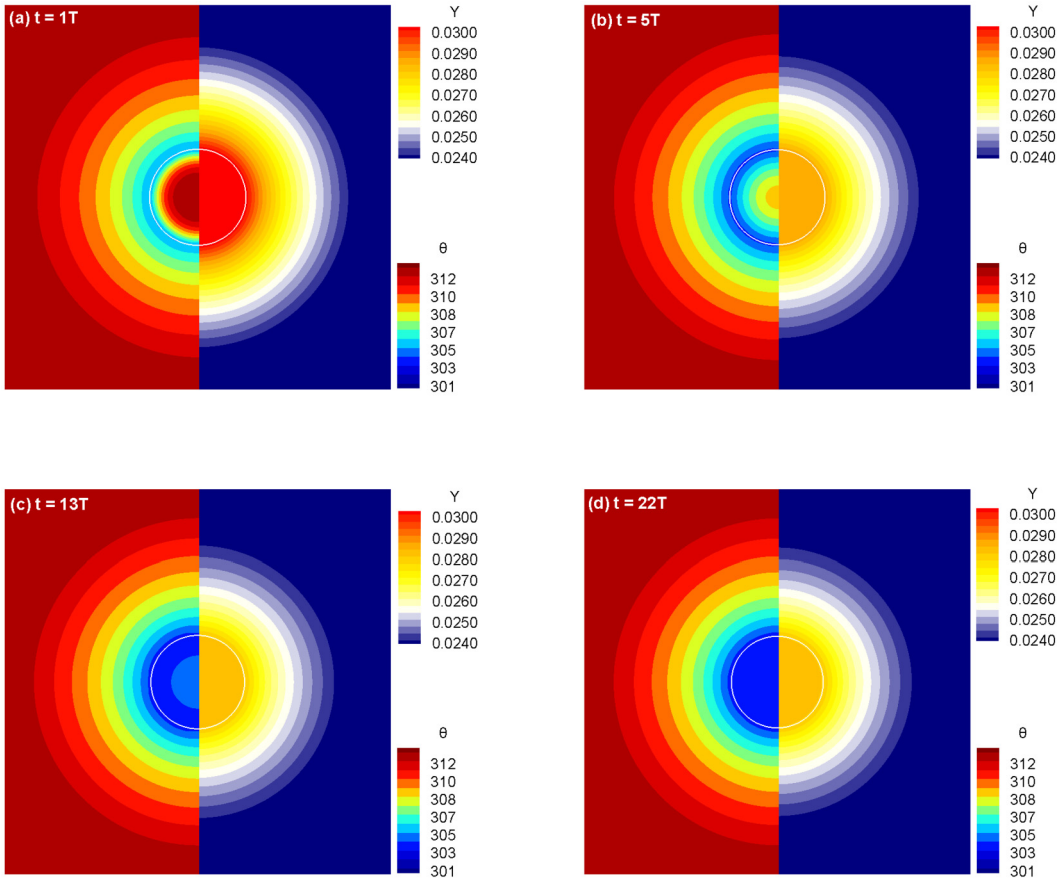


Fig. 19. Instantaneous geometry of a droplet and contours of temperature θ and vapor mass fraction Y during the simulation of the evaporation below the saturation temperature θ_{sat} at (a) $t = 1T$, (b) $t = 5T$, (c) $t = 13T$ and (d) $t = 22T$. The temperature contours are partially plotted on the left and the vapor mass fraction contours are shown on the right side of the figures. The interface is represented by a white solid line. The dry bulb temperature and the relative humidity are $\theta_{db} = 313$ K and $\Psi = 50\%$, respectively. The computational domain size is $4D_0 \times 4D_0$, discretized evenly by 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

Fig. 23 shows successive snapshots of the droplet position, temperature and the vapor mass fraction fields. It is seen that the temperature inside the droplet decreases with time and a low temperature zone is formed in the wake of the droplet. Meanwhile, the vapor mass aggregates in the wake of the droplet and two "tails" are observed in both the vapor mass fraction and temperature fields, an observation that is consistent with Irfan and Muradoglu [18]. To further quantify the motion of the droplet, we extract the coordinates of the droplet bottom and show the evolution of its position in Fig. 25. The results are compared with the results of Irfan and Muradoglu [18] and Zhao et al. [37]. It is observed that the droplet position is lower than that reported in Irfan and Muradoglu [18] and is higher than that given by Zhao et al. [37]. The difference in the evolution of the droplet position can be attributed to the numerical strategies. The interface capturing scheme is the front-tracking method in Irfan and Muradoglu [18], while the VOF method is employed by Zhao et al. [37]. Besides, the work of Zhao et al. [37] is based on the finite-volume method within a $2D$ axis-symmetric framework,while both Irfan and Muradoglu [18] and the present study are based on the finite-difference method in a $2D$ Cartesian domain. We note that the advantages of difference numerical schemes are not the focus of this study. The main objective of this plot is to show that our numerical results based on the velocity decomposition method are reasonable.

We have also simulated this case using strategy $S2$, in which the velocity decomposition is not utilized. Fig. 24 shows successive snapshots of the droplet position, temperature and the vapor mass fraction field. It is seen that the interface evolves steadily with the temperature and vapor mass fraction being reasonably resolved. To further quantify the difference of the results between $S1$ and $S2$, the evolution of the droplet position of $S2$ is superimposed in Fig. 25. It is found that numerical result of $S2$ is closer to the result of Irfan and Muradoglu [18], which is reasonable since the velocity decomposition is also not used in the work of Irfan and Muradoglu [18]. This test result also indicates that the numerical simulation can evolve robustly when the droplet motion is dominated by an external force. This is because the magnitude of the velocity induced by the external force is greater than the spurious velocity, the normal vector of the interface is not distorted significantly in this case.

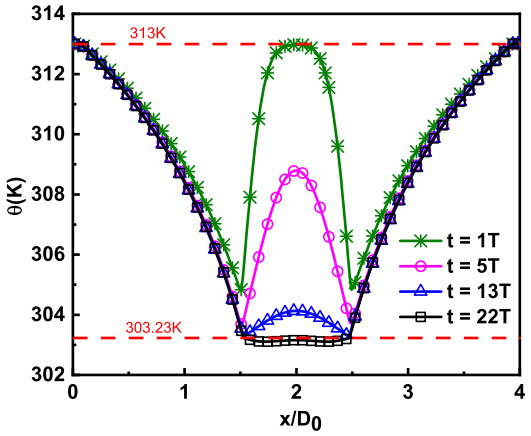


Fig. 20. Evolution of temperature distribution along the horizontal line at $y = 2D_0$ for the test case of $2D$ droplet evaporation below the saturation temperature θ_{sat}. Two red dashed lines are the dry bulb temperature θ_{db} (top) and the wet bulb θ_{wb} temperature (bottom), respectively. The dry bulb temperature, wet bulb temperature and the relative humidity are $\theta_{db} = 313$ K, $\theta_{db} = 303.23$ K, $\theta_{wb} = 303.23$ K, and $\Psi = 50\%$, respectively. The domain size is $4D_0 \times 4D_0$, discretized using 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

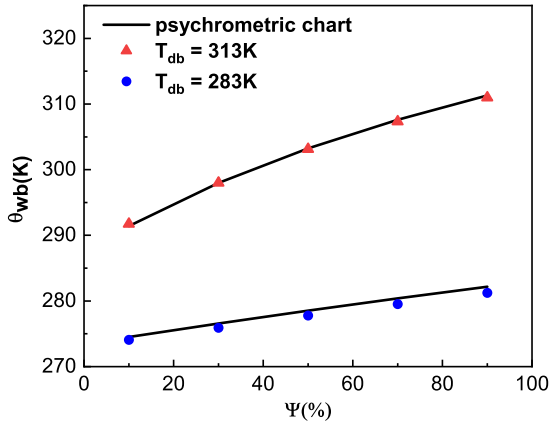


Fig. 21. Wet bulb temperature θ_{wb} corresponding to different relative humidity Ψ and dry bulb temperature θ_{db}. The relative humidity Ψ ranges from 10% to 90% and the dry bulb temperature θ_{db} is either 313 K or 283 K in different test cases. The psychometric chart data is represented by the black solid lines. The computational domain size is $4D_0 \times 4D_0$, discretized using 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

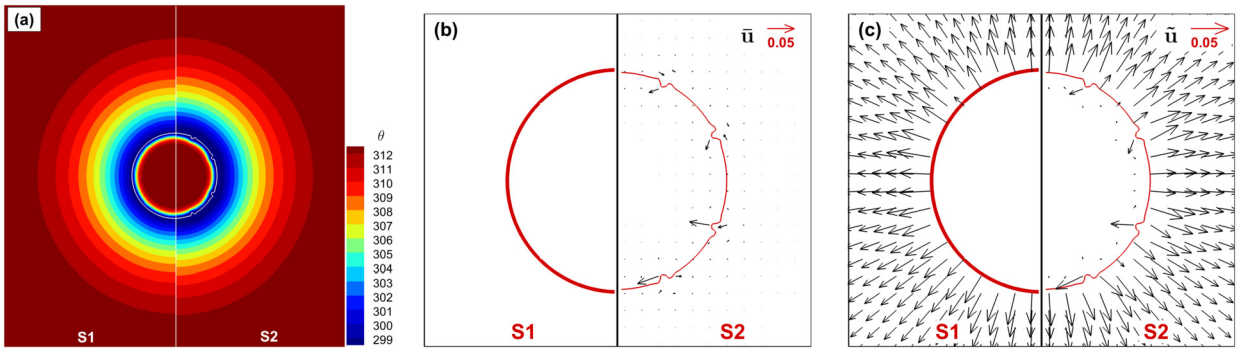


Fig. 22. Numerical results of $S1$ (left) and $S2$ (right) for test case of droplet evaporation below the saturated temperature θ_{sat} at $t = 0.2T$. (a) The geometry of the droplet, contours of temperature and vapor mass fraction; (b) The velocity vector of rotational-part velocity $\bar{\mathbf{u}}$; (c) The velocity vector of $\tilde{\mathbf{u}}$ of Stephan flow. The dry bulb temperature and relative humidity are $\theta_{db} = 313$ K and $\Psi = 10\%$, respectively. The computational domain size is $4D_0 \times 4D_0$, discretized using 256×256 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

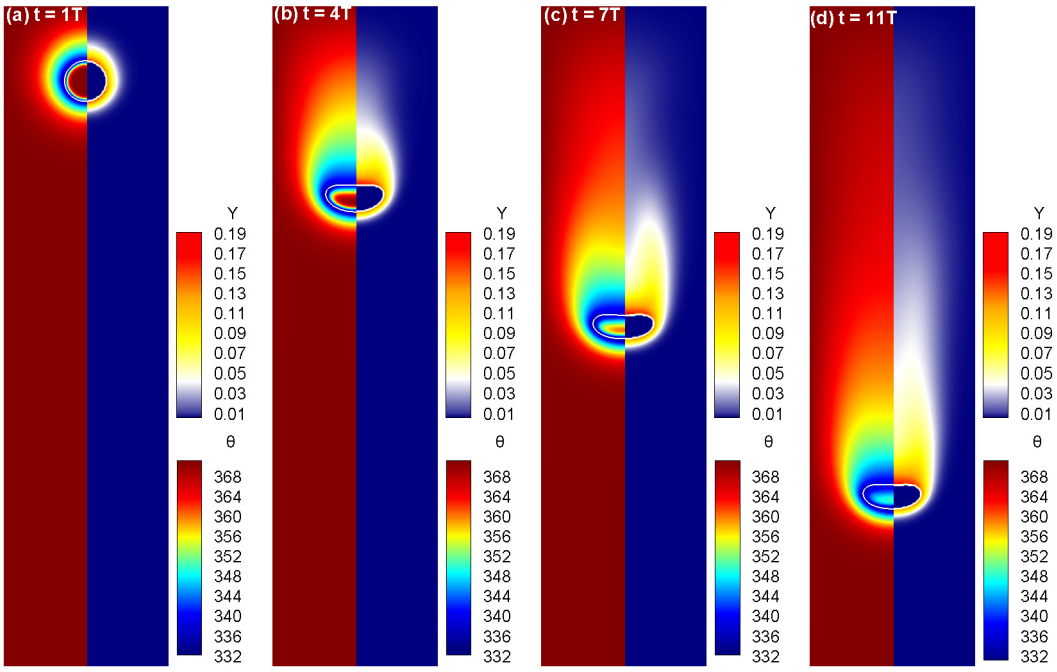


Fig. 23. Instantaneous geometry of a moving droplet (white line) and the corresponding contours of temperature (left) and vapor mass fraction (right) obtained from strategy $S1$ at (a) $t = 1T$, (b) $t = 4T$, (c) $t = 7T$ and (d) $t = 11T$. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $Sc = 1.0$, $Pr = 0.15$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 16D_0$, discretized using 128×512 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3.4. Falling of 3D evaporating droplets under gravity

3.4.1. Falling of a 3D droplet with constant evaporation rate

Before the validation of the fully coupled system in a $3D$ domain, we validate our method by testing a falling droplet with a constant evaporation rate. The computational domain size for this case is $L_x \times L_y \times L_z = 4D_0 \times 8D_0 \times 4D_0$, discretized using $N_x \times N_y \times N_z = 128 \times 256 \times 128$ grid points. The droplet is released from $(x, y, z) = (2D_0, 7.2D_0, 2D_0)$. Other parameters for the momentum solver, including the physical properties η, initial condition and boundary condition remain the same as that for the test case in Sec. 3.3.2. The mass transfer rate $\dot{M}$ is fixed at 1.05, but it is set to 0 before the bottom of the droplet reaches $Y = 5.7D_0$ to generate an initial velocity field. Then, the velocity and interface position at that time are set as the initial condition for the following evaporation simulations. The objective for this case is to test if $S1$ and $S2$ are both robust with a high evaporation rate in the problems with gravity.

The evolution of the interface geometries obtained from $S1$ and $S2$ are shown in Figs. 26. It is seen from Fig. 26(a) that the evolution of the droplet is reasonable when $S1$ is employed, while the interface in the results of $S2$ is unsatisfactory as shown in Fig. 26(b). This is because the numerical errors in $\tilde{\mathbf{u}}$ are more significant when $S2$ is utilized. The error-induced spurious velocity is not negligible compared with the velocity caused by gravity and thus alters the normal direction of the interface. As a consequence, the interface evolves in an unphysical manner. Fig. 27 compares the evolution of droplet volume V in $S1$ and $S2$. The analytical solution of a stationary spherical droplet with the same $\dot{M}$ is superimposed for comparison. It is seen that the evolution of the droplet volume obtained from $S1$ is close to the analytical solution of the stationary spherical droplet case. The slightly faster loss of volume in $S1$ than the analytical solution is reasonable because the interface area for a moving droplet is larger than the spherical droplet, and thus a moving droplet evaporates faster. However, the numerical result from $S2$ is undesirable, the volume is reduced too fast due to the undesirable interface geometry shown in Fig. 26(b).

We have also tested the time consuming for calculating $\hat{\mathbf{u}}$ based on this case. If we set the same relative tolerance for all Poisson equations to $O(10^{-6})$, it takes 110 steps for the two Poisson equations in the momentum solver (Eq. (18)) to converge, while 130 steps are needed for another two (Eqs. (13) and (24)) to converge, and thus solving Eq. (24) occupies approximately 27% of the overall simulation cost. However, based on our test, the tolerance for Eq. (24) is not required to be as small as those for other Poisson equations, because $\hat{\mathbf{u}}$ only accounts for the mass conservation, but does not directly participate in the evolution of the momentum equation. If we further reduce the tolerance for Eq. (24) to $O(10^{-2})$, the volume at the end of the simulation only changes 0.000321%, while the iteration step required for Eq. (24) decreases to approximately 75, which occupies 17.6% of the simulation cost.

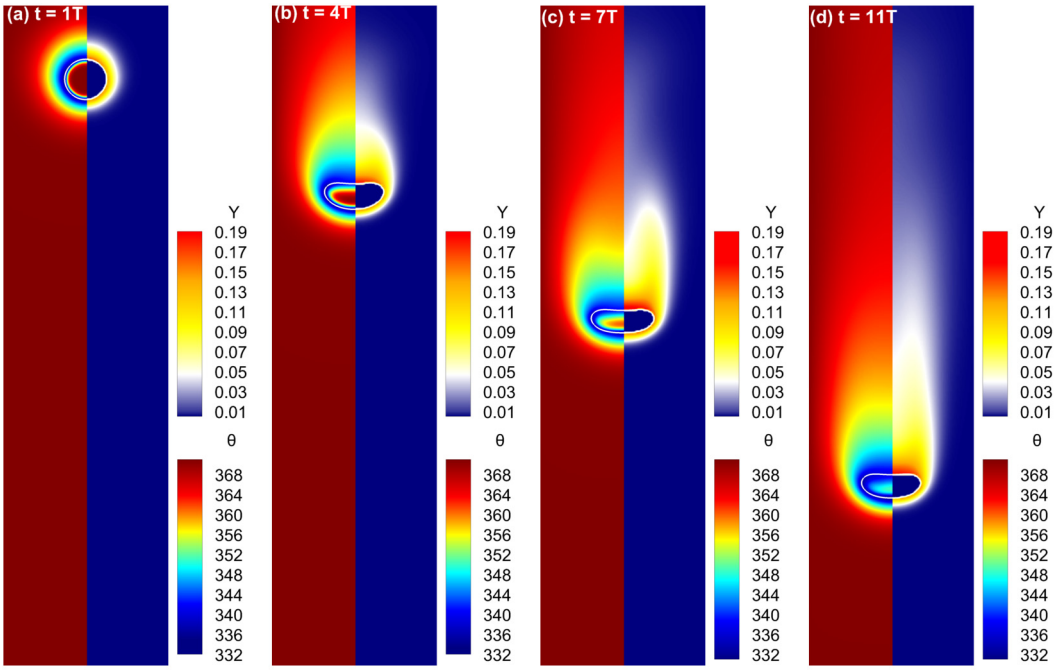


Fig. 24. The geometry of a moving droplet (white line) and the corresponding contours of temperature (left) and vapor mass fraction (right) obtained from $S2$ at (a) $t = 1T$, (b) $t = 4T$, (c) $t = 7T$ and (d) $t = 11T$. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $Sc = 1.0$, $Pr = 0.15$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 16D_0$, discretized using 128×512 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

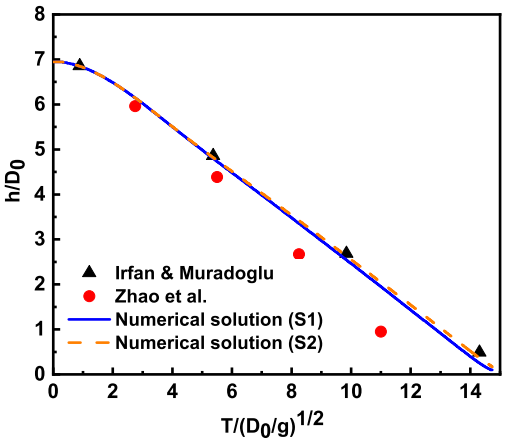


Fig. 25. Evolution of bottom position of droplet for test case of a $2D$ droplet evaporation under gravity. The blue solid line and the yellow dashed line are the numerical solutions of $S1$ and $S2$, respectively. The black triangles and the red dots are the numerical results in the literature. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $Sc = 1.0$, $Pr = 0.15$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 16D_0$, discretized using 128×512 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3.4.2. Falling of a 3D evaporating droplet below saturation temperature

Finally, we validate our method in the fully coupled system by testing the interaction of two falling droplets with phase change in a $3D$ domain. The computational domain size is $L_x \times L_y \times L_z = 6D_0 \times 16D_0 \times 4D_0$, discretized using $N_x \times N_y \times N_z = 192 \times 512 \times 128$ grid points. The two droplets are released simultaneously at $(x_1, y_1, z_1) = (2D_0, 14.4D_0, 2D_0)$ and $(x_2, y_2, z_2) = (4D_0, 14.4D_0, 2D_0)$, respectively. Other parameters, including the physical properties η, initial condition and boundary condition remain the same as that for the test case in Sec. 3.3.2. The successive snapshots of the temperature θ and vapor mass fraction Y are shown in Figs. 28 and 29, respectively. It is seen that the deformation of the droplets is less significant than that in the $2D$ case, though the fluid properties remain completely the same. As a consequence, the structures of temperature and vapor fields in the wake of the droplet are different from the results in the $2D$ case. This is similar to the numerical results of a $2D$ axis-symmetric droplet by Zhao et al. [37], which is closer to the $3D$ result than the $2D$ Cartesian framework. It is also found that the distance between two droplets increases as they fall down. This is

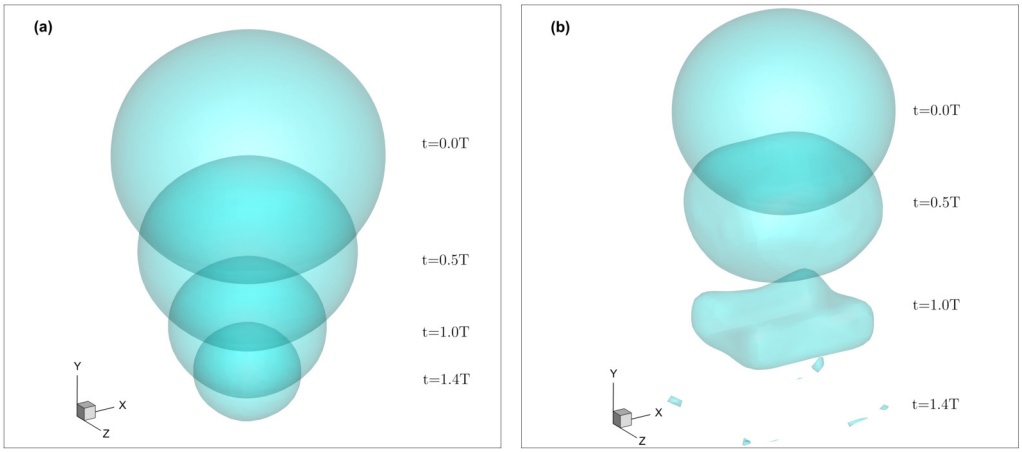


Fig. 26. Interface geometries of a falling three-dimensional evaporating droplet at different time instants obtained from using (a) $S1$ and (b) $S2$. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 8D_0 \times 4D_0$, discretized using $128 \times 256 \times 128$ grid points.

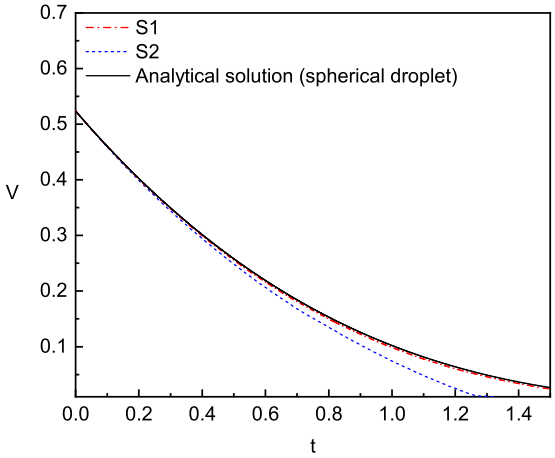


Fig. 27. Evolution of the droplet volume obtained from using $S1$ and $S2$. The analytical solution of a stationary sphere droplet with the $\dot{M} = 1.05$ is superimposed for comparison. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 8D_0 \times 4D_0$, discretized using $128 \times 256 \times 128$ grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

evident from the trajectories of the two droplets depicted in Fig. 30(a). The separation of two droplets is also observed in the $2D$ work of Irfan and Muradoglu [18]. We also depict the evolution of the mass of liquid m in Fig. 30(b), it is seen that originally the mass of liquid is M_0 and it decreases drastically at the early stage of the simulation, and after $t = 1T$ the mass is lost at a constant rate approximately.

4. Conclusion

In this paper, we have proposed a new phase-change model for the numerical simulation of evaporation within the CLSVOF framework. The basic idea is to construct a smooth interfacial velocity $\mathbf{u}_\Gamma$ by decoupling the effect of phase change from the total velocity $\mathbf{u}$. However, unlike in the geometric VOF framework, the interface capturing scheme (CLSVOF method) adopted in our solver is more sensitive to the numerical error. Unsatisfactory numerical results are observed in the case of the evaporation of a stationary droplet if we simply implement the method of Malan et al. [35]. To address the issue, the velocity in the proposed method is decomposed into a potential part induced solely by the mass source corresponding to the phase change and the residual rotational part. The potential velocity $\tilde{\mathbf{u}}$ is computed by solving a Poisson equation. The rotational-part velocity $\bar{\mathbf{u}}$ is governed by the modified momentum equation and solved using the NS-equation solver CAS-Tank developed in our previous study. Besides, by shifting the source points to the vicinity of the interface, a continuous interfacial velocity $\mathbf{u}_\Gamma$ is constructed at the cells with liquid-gas interfaces. Thus, the velocity jump at the interface is avoided and the interface propagates accurately with mass conserved.

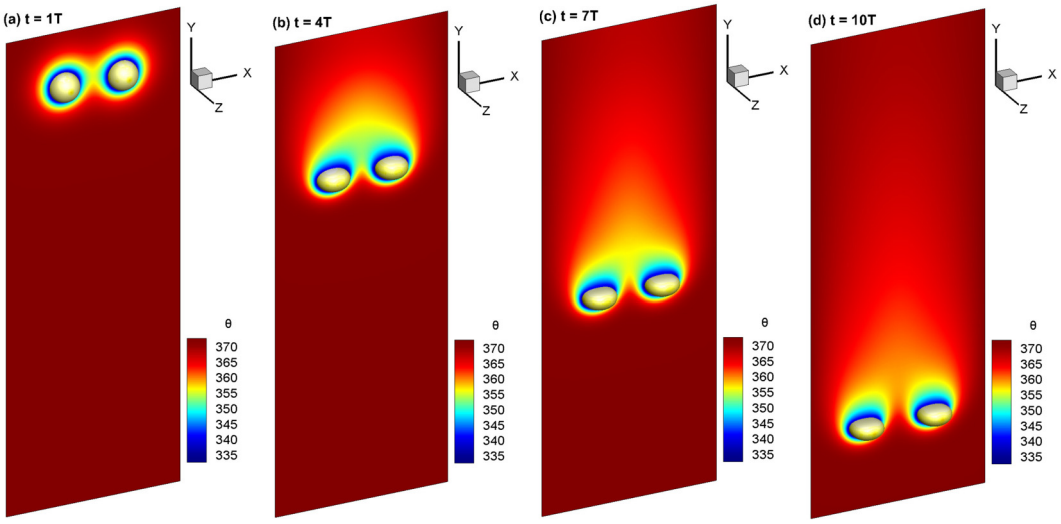


Fig. 28. Contours of temperature in the middle slice and the geometry of the droplet for the $3D$ droplets evaporation case at (a) $t = 1T$, (b) $t = 4T$, (c) $t = 7T$, (d) $t = 10T$. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $Sc = 1.0$, $Pr = 0.15$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 16D_0$, discretized using 128×512 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

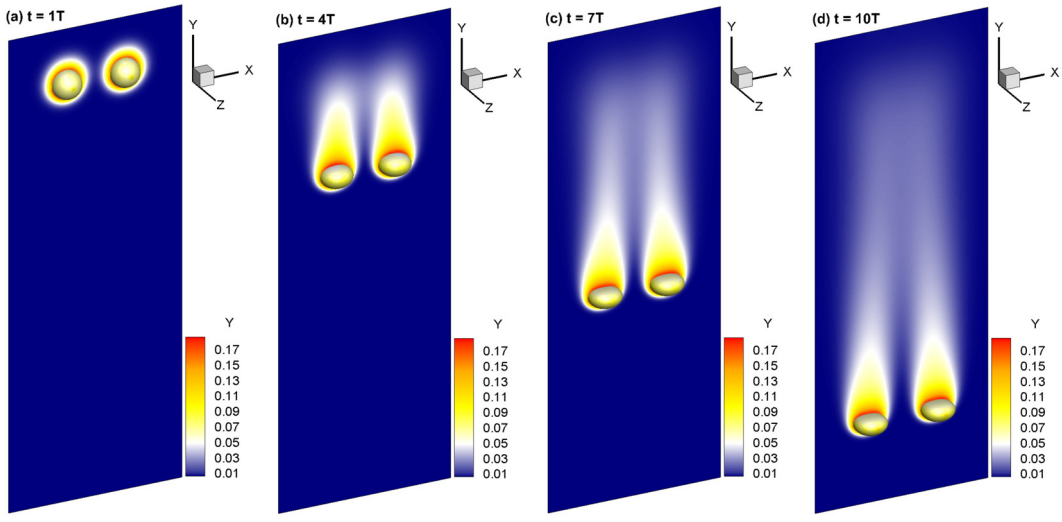
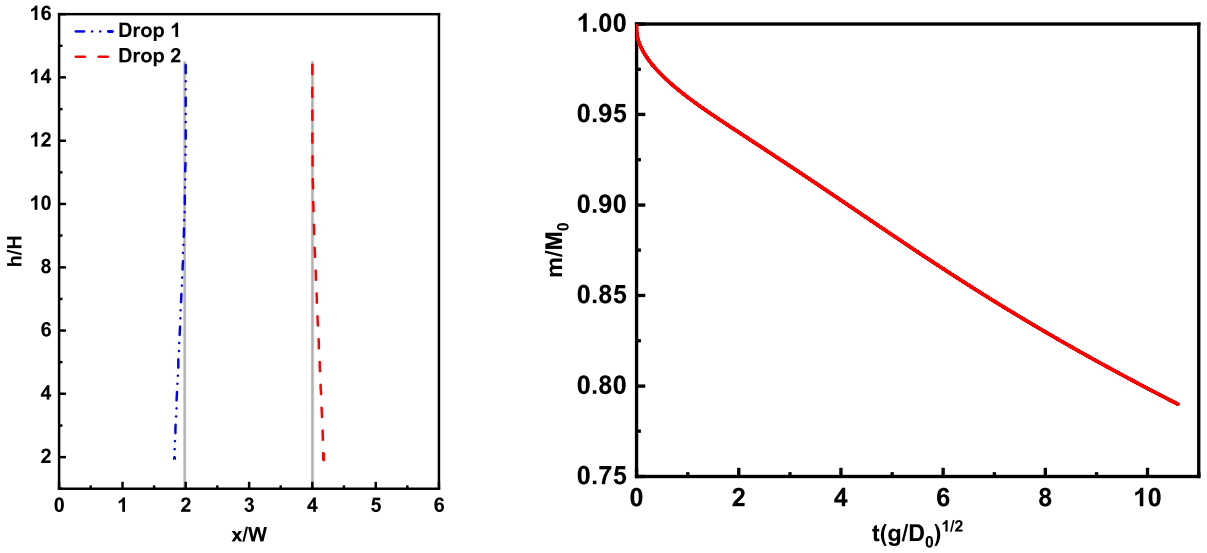


Fig. 29. Contours of vapor in the middle slice and the geometry of the droplet for the $3D$ droplets evaporation case at (a) $t = 1T$, (b) $t = 4T$, (c) $t = 7T$, (d) $t = 10T$. The non-dimensional parameters are $Re = 28.82$, $We = 2.5$, $Fr = 1.0$, $Sc = 1.0$, $Pr = 0.15$, $\gamma_\rho = 5$ and $\gamma_\mu = 20$. The computational domain size is $4D_0 \times 16D_0$, discretized using 128×512 grid points. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

The proposed method is first tested by simulating a $2D$ droplet with constant evaporation rate. This test decouples the effect of the transport of temperature and vapor mass fraction on the interface evolution. The numerical results show that the evolution of droplet diameter converges to the analytical solution as the grid is refined. It is also found that the accuracy of the proposed method is close to the second order. After this test, a 1D Stefan problem and a 1D sucking problem are simulated to verify the coupling of the transport equations of momentum and temperature. The numerical results converge to the analytical solution as the grid is refined in both test cases. A further test of the droplet evaporation at saturation temperature θ_{sat} is conducted in a $2D$ domain. We compare the numerical results with the analytical solution for different St numbers and found that the numerical results are in agreement with the analytical solutions. The verification is finished by simulating the evaporation of a $2D$ droplet below the saturation temperature. In this test case, the transport equations of momentum, temperature and vapor mass fraction are fully coupled. The wet bulb temperature θ_{wb} when the simulation reaches a steady state agrees with the data read from the psychometric chart under different conditions. Finally, we validate the proposed method by simulating the falling of evaporating droplets in both $2D$ and $3D$ domains. The evolution of the $2D$ droplet position is in general consistent with the results in the literature. The capability of the proposed method to perform



(a) (b)

Fig. 30. Numerical results for the test case of 3D evaporating droplets under the gravity. (a) The trajectories of the centroid of the droplets at XOY plane. The blue dash dot line represents the trajectory of the left droplet and the red dash line is the trajectory of the right one. The two gray vertical straight lines are employed to demarcate the distance between the two droplets and its original position in x-direction. (b) Evolution of the total mass of liquid. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

3D simulations is validated by simulating the falling of two evaporating droplets. It is observed that the two droplets separate from each other gradually as they fall down, which is also consistent with the observation in a 2D work in the literature.

In summary, the present study proposes a new method to implement the phase-change model into the CLSVOF framework. Specifically, there are two key treatments for ensuring numerical accuracy. The first treatment is to decompose the total velocity into a potential part and a rotational part, and solve them separately to eliminate the nonlinear interaction between the errors in these two parts. This treatment reduces the spurious velocity in rotational part velocity, which is crucial for accurate interface propagation in the CLSVOF framework. The second treatment is to reallocate the source into the liquid phase to calculate the interface regression velocity. This treatment ensures the mass conservation of the present phase-change model in the CLSVOF framework with the source term being sharply distributed in the interface cells.

CRediT authorship contribution statement

Min Lu: Formal analysis, Investigation, Methodology, Software, Validation, Writing – original draft. **Zixuan Yang:** Conceptualization, Project administration, Supervision, Writing – review & editing. **Guowei He:** Conceptualization, Supervision.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

This research is supported by the National Natural Science Foundation of China (NSFC) Basic Science Center Program for 'Multiscale Problems in Nonlinear Mechanics' (No. 11988102), NSFC project (No. 11972038), and Strategic Priority Research Program (Grant No. XDB22040104).

References

[1] Z. Yang, M. Lu, S. Wang, A robust solver for incompressible high-Reynolds-number two-fluid flows with high density contrast, J. Comput. Phys. 441 (2021) 110474.

[2] S. Subramaniam, Lagrangian–Eulerian methods for multiphase flows, Prog. Energy Combust. Sci. 39 (2013) 215–245.
[3] I. Senocak, W. Shyy, A pressure-based method for turbulent cavitating flow computations, J. Comput. Phys. 176 (2002) 363–383.
[4] T. Liu, B. Khoo, W. Xie, Isentropic one-fluid modelling of unsteady cavitating flow, J. Comput. Phys. 201 (2004) 80–108.
[5] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100 (1992) 25–37.
[6] J. Glimm, J.W. Grove, X.L. Li, K.-M. Shyue, Y. Zeng, Q. Zhang, Three-dimensional front tracking, SIAM J. Sci. Comput. 19 (1998) 703–727.
[7] G. Tryggvason, A. Esmaeeli, N. Al-Rawahi, Direct numerical simulations of flows with phase change, Comput. Struct. 83 (2005) 445–453.
[8] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flow, Annu. Rev. Fluid Mech. 31 (1999) 567–603.
[9] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions to incompressible two-phase flow, J. Comput. Phys. 114 (1994) 146–159.
[10] Y. Zeng, A. Xuan, J. Blaschke, L. Shen, A parallel cell-centered adaptive level set framework for efficient simulation of two-phase flows with subcycling and non-subcycling, J. Comput. Phys. 448 (2022) 110740.
[11] Y. Zeng, Numerical Simulations of the Two-phase flow and Fluid-Structure Interaction Problems with Adaptive Mesh Refinement, Ph.D. thesis, University of Minnesota, 2022.
[12] M. Theillard, F. Gibou, D. Saintillan, Sharp numerical simulation of incompressible two-phase flows, J. Comput. Phys. 391 (2019) 91–118.
[13] C.R. Kharangate, I. Mudawar, Review of computational studies on boiling and condensation, Int. J. Heat Mass Transf. 108 (2017) 1164–1196.
[14] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169 (2001) 708–759.
[15] D. Juric, G. Tryggvason, Computations of boiling flows, Int. J. Multiph. Flow 24 (1998) 387–410.
[16] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part I: numerical method, Int. J. Heat Mass Transf. 47 (2004) 5451–5461.
[17] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part II: multi-mode film boiling, Int. J. Heat Mass Transf. 47 (2004) 5463–5476.
[18] M. Irfan, M. Muradoglu, A front tracking method for direct numerical simulation of evaporation process in a multiphase system, J. Comput. Phys. 337 (2017) 132–153.
[19] G. Son, V.K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, J. Heat Transf. 120 (1998) 183–192.
[20] S.W.J. Welch, J. Wilson, A volume of fluid based method for fluid flows with phase change, J. Comput. Phys. 160 (2000) 662–682.
[21] J. Schlottke, B. Weigand, Direct numerical simulation of evaporating droplets, J. Comput. Phys. 227 (2008) 5215–5237.
[22] Y. Sato, B. Niceno, Nucleate pool boiling simulations using the interface tracking method: boiling regime from discrete bubble to vapor mushroom region, Int. J. Heat Mass Transf. 105 (2017) 505–524.
[23] D.Q. Nguyen, R.P. Fedkiw, M. Kang, A boundary condition capturing method for incompressible flame discontinuities, J. Comput. Phys. 172 (2001) 71–98.
[24] R.P. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457–492.
[25] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, J. Comput. Phys. 222 (2007) 536–555.
[26] F. Gibou, R. Fedkiw, R. Caflisch, S. Osher, A level set approach for the numerical simulation of dendritic growth, J. Sci. Comput. 19 (2003) 183–199.
[27] S. Tanguy, T. Ménard, A. Berlemont, A level set method for vaporizing two-phase flows, J. Comput. Phys. 221 (2007) 837–853.
[28] L.R. Villegas, R. Alis, M. Lepilliez, S. Tanguy, A ghost fluid/level set method for boiling flows and liquid evaporation: application to the leidenfrost effect, J. Comput. Phys. 316 (2016) 789–813.
[29] G. Sahut, G. Ghigliotti, G. Balarac, M. Bernard, V. Moureau, P. Marty, Numerical simulation of boiling on unstructured grids, J. Comput. Phys. 432 (2021) 110161.
[30] J. Palmore Jr, O. Desjardins, A volume of fluid framework for interface-resolved simulations of vaporizing liquid-gas flows, J. Comput. Phys. 399 (2019) 108954.
[31] E. Bayat, R. Egan, D. Bochkov, A. Sauret, F. Gibou, A sharp numerical method for the simulation of Stefan problems with convective effects, J. Comput. Phys. (2022) 111627.
[32] A. Guittet, M. Lepilliez, S. Tanguy, F. Gibou, Solving elliptic problems with discontinuities on irregular domains–the Voronoi interface method, J. Comput. Phys. 298 (2015) 747–765.
[33] D. Bochkov, F. Gibou, Solving elliptic interface problems with jump conditions on cartesian grids, J. Comput. Phys. 407 (2020) 109269.
[34] R. Egan, F. Gibou, xgfm: recovering convergence of fluxes in the ghost fluid method, J. Comput. Phys. 409 (2020) 109351.
[35] L.C. Malan, A.G. Malan, S. Zaleski, P.G. Rousseau, A geometric VOF method for interface resolved phase change and conservative thermal energy advection, J. Comput. Phys. 426 (2021) 109920.
[36] N. Scapin, P. Costa, L. Brandt, A volume-of-fluid method for interface-resolved simulations of phase-changing two-fluid flows, J. Comput. Phys. 407 (2020) 109251.
[37] S. Zhao, J. Zhang, M.-J. Ni, Boiling and evaporation model for liquid-gas flows: a sharp and conservative method based on the geometrical VOF approach, J. Comput. Phys. 452 (2022) 110908.
[38] L.G. Martinez, B. Duret, J. Reveillon, F. Demoulin, A new DNS formalism dedicated to turbulent two-phase flows with phase change, Int. J. Multiph. Flow 143 (2021) 103762.
[39] P. Ferrant, L. Gentaz, B. Alessandrini, D. Le Touzé, E.C. de Nantes, A potential/RANSE approach for regular water wave diffraction about 2-d structures, Ship Technol. Res. 50 (2003) 165–171.
[40] K. Kim, A.I. Sirviente, R.F. Beck, The complementary RANS equations for the simulation of viscous flows, Int. J. Numer. Methods Fluids 48 (2005) 199–229.
[41] Z. Li, B. Bouscasse, G. Ducrozet, L. Gentaz, D. Le Touzé, P. Ferrant, Spectral wave explicit Navier-Stokes equations for wave-structure interactions using two-phase computational fluid dynamics solvers, Ocean Eng. 221 (2021) 108513.
[42] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (2003) 110–136.
[43] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (2004) 349–355.
[44] D. Bochkov, F. Gibou, Pde-based multidimensional extrapolation of scalar fields over interfaces with kinks and high curvatures, SIAM J. Sci. Comput. 42 (2020) A2344–A2359.
[45] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, S. Zaleski, Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423–456.
[46] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (2000) 228–237.
[47] G.Y. Soh, G.H. Yeoh, V. Timchenko, An algorithm to calculate interfacial area for multiphase mass transfer through the volume-of-fluid method, Int. J. Heat Mass Transf. 100 (2016) 573–581.
[48] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct Numerical Simulations of Gas–Liquid Multiphase Flows, Cambridge University Press, 2011.
[49] Y. Zeng, A.P.S. Bhalla, L. Shen, A subcycling/non-subcycling time advancement scheme-based DLM immersed boundary method framework for solving single and multiphase fluid–structure interaction problems on dynamically adaptive grids, Comput. Fluids 238 (2022) 105358.

[50] J.K. Patel, G. Natarajan, A generic framework for design of interface capturing schemes for multi-fluid flows, Comput. Fluids 106 (2015) 108–118.
[51] M. Lu, Z. Yang, G. He, A robust scheme for numerical simulation of heat transfer in two-fluid flows with high volumetric heat capacity contrasts, Int. J. Numer. Methods Heat Fluid Flow (2022), https://doi.org/10.1108/HFF-05-2022-0296, in press.
[52] G.H. Shortley, R. Weller, The numerical solution of Laplace's equation, J. Appl. Phys. 9 (1938) 334–348.
[53] F. Gibou, R. Fedkiw, A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem, J. Comput. Phys. 202 (2005) 577–601.
[54] S. Dong, G.E. Karniadakis, C. Chryssostomidis, A robust and accurate outflow boundary condition for incompressible flow simulations on severely-truncated unbounded domains, J. Comput. Phys. 261 (2014) 83–105.