International Journal of Heat and Mass Transfer 116 (2018) 30–49

Contents lists available at ScienceDirect

International Journal of Heat and Mass Transfer

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

Discretization and implementation of a sharp interface model for interfacial heat and mass transfer during bubble growth

Isaac Perez-Raya [a], Satish G. Kandlikar [a,b,*]

[a] Department of Microsystems Engineering, Rochester Institute of Technology, Rochester 14623-5604, NY, USA
[b] Department of Mechanical Engineering, Rochester Institute of Technology, Rochester 14623-5604, NY, USA

ARTICLE INFO

Article history:
Received 25 May 2017
Received in revised form 25 August 2017
Accepted 28 August 2017
Available online 9 September 2017

Keywords:
Mass transfer
Sharp interface
Phase change
Bubble growth
Interfacial gradient

ABSTRACT

Simulations of phase change rely on methods to compute heat and mass transfer at the interface. Current methods estimate the mass transfer with interpolation functions or by assuming a local temperature difference between saturation and interface. This work reports a method that uses a single cell around the interface to find the interfacial temperature gradient, and a linear interpolation normal to the interface to find the temperature of the mixture cell (cell with an interface). The one-cell algorithm for sharp interface and mass transfer (OCASIMAT) simplifies the estimation of the mass transfer and mixture cell temperature and improves accuracy. The proposed approach leads to a more realistic representation of the heat and mass transfer at the interface with a sharp discontinuity on the thermal properties at the interface and with mass transfer only at the cells with an interface. Simulations of planar interface evaporation and spherical bubble growth demonstrate the application of the proposed approach. Results indicate that OCASIMAT accurately predicts the temperature distributions near the interface and the interface displacements.

1. Introduction

Applications of evaporation and boiling are found in heat sinks for electronics cooling, nuclear and fossil fuel powered steam generators, distillation columns, concentrated solar power systems, glass melting furnaces, desalination chambers, and heat and mass exchangers. In order to increase the performance and safety margins of these applications, there is a need to develop tools that predict the localized interfacial dynamics during the phase change process. The analysis of bubble growth has been addressed by computer simulations, which employ methods for mass transfer and for the dynamic interfacial interaction. Accurate methods available in the technical literature to simulate bubble growth require advanced algorithms that present significant challenges in the analysis of phase change at the interface level. Other approaches require simplifying assumptions at the expense of accuracy. A new approach is presented in this paper to simplify the interfacial mass transfer computations and to include a sharp interface representation while preserving accuracy.

Simulations of boiling and evaporation have computed the mass transfer with the local difference between the saturation and an interface temperature [1–9]. This approach requires an iterative procedure to match the new interface position with the corresponding interface temperature [10]. Similarly, Strotos et al. adopted [11] an evaporation model that considered a difference between the saturation and bulk pressures in the simulation of droplet evaporation. Maki and Kumar [12] simulated droplet evaporation by assuming a temperature difference between the interface and a saturation temperature in addition to a pressure difference. These models avoid the identification of the interface location since they use local differences instead of gradients at the interface. However, the models depend on an empirically derived accommodation coefficient, and therefore provide only an indirect estimation of the mass transfer.

Other phase change simulations estimate the mass transfer with the temperature gradients at the center or faces of the surrounding cells rather than at the interface [13–16]. Haelssig et al. [13] computed the temperature gradient at the center of the mixture cell (a cell with an interface). The temperature gradient considered the temperature of the neighboring cells (cells around the mixture cell) and the normal vector. Sun et al. [14] computed the temperature gradient at the faces of the mixture cell. These approaches ignore the interface location for the purpose of easy calculation, but they sacrifice accuracy by computing the temperature gradient at a certain distance from the interface. Moreover, interpolation functions are needed to find proper temperatures of

[*] Corresponding author at: Department of Mechanical Engineering, Rochester Institute of Technology, Rochester 14623-5604, NY, USA.
E-mail address: skgme@rit.edu (S.G. Kandlikar).

http://dx.doi.org/10.1016/j.ijheatmasstransfer.2017.08.106
0017-9310/© 2017 Elsevier Ltd. All rights reserved.

Nomenclature

A_i	interface surface area (m^2)
D	distance interface to cell center (m)
C	large coefficient
C	specific heat (J/kg K)
F	volume-fraction
h_{fg}	latent heat (J/kg)
K	thermal conductivity (W/m K)
K	curvature (1/m)
m''	mass flux (kg/s m^2)
$\hat{n}$	unit normal vector
p	pressure (N/m^2)
r	radial coordinate (r)
R	bubble radius (m)
S_M	momentum source term (N/m^3)
S_T	energy source term (K/s)
sd	distance interface to cell center (m)
T	temperature (K)
t	time (s)

V_c	cell Volume (m^3)
v	velocity (m/s)
X	interface position (m)
x	x-position (m)
α	thermal diffusivity (m^2/s)
β	bubble growth constant
μ	dynamic viscosity (Pa-s)
ξ	growth rate constant
ρ	density (kg/m^3)
σ	surface tension (N/m)
ϕ	density ratio

Subscripts

l	liquid
sat	saturation
v	vapor

the mixture cells or the neighboring cells that lie on the opposite side of the interface.

Another approach is to compute the mass transfer with temperature gradients at the interface [17–21]. Akthar and Kleis [17,18] defined the temperature gradient with the distance from the neighboring cell center to the interface along the Cartesian directions. Schlottke and Weigand [21] adopted a similar approach to simulate droplet evaporation. Their mass transfer model considered vapor mass fractions at the interface and neighboring cells rather than temperatures. When the interface does not intersect with the centerlines of the mixture cells, the method is no longer applicable, and the gradient could be found with an interpolated mixture cell temperature [15]. Kunkelmann and Stephan [19] computed the mass transfer with an interface temperature, the saturation temperature, and the distance from the interface to the mixture cell. The main difficulty with this method is the estimation of the interface temperature, which requires an iterative procedure.

The present work provides a one-cell algorithm for sharp interface and mass transfer (OCASIMAT) that simplifies the estimation of mass transfer and improves accuracy. The proposed approach finds the temperature gradient at the interface with the temperature of only one computational cell without interpolation functions. To account for a sharp interface, a mathematical expression directly estimates the temperature of the mixture cells based on the temperature gradient at the interface. Two fundamental problems involving interfacial heat and mass transfer are used to evaluate and validate this approach; these problems include planar interface evaporation and spherical bubble growth.

2. Numerical model

This section is divided into two subsections. Section 2.1 shows the governing equations for the interface tracking and the mass transfer. It also shows the equations for momentum and energy transport. Section 2.2 focuses on the temperature gradients at the interface and on the temperature of the cells with an interface.

2.1. Governing equations

The volume-of fluid (VOF) equation Eq. (1) for the vapor volume-fraction tracks the interface.

$$\frac{\partial}{\partial t}(F_v) + \nabla\cdot(F_v\overrightarrow{v_v}) = \frac{m''}{\rho_v}\frac{A_i}{V_c} \tag{1}$$

where F_v is the vapor volume fraction on each cell, ρ_v is the vapor density, $\overrightarrow{v_v}$ is the vapor velocity, m'' is the mass flux, A_i is the interface surface area, and V_c is the cell volume. Eq. (2) is a constraint for the liquid volume fraction, F_l, at each cell.

$$F_l + F_v = 1 \tag{2}$$

The piecewise-linear scheme reconstructs the interface, which assumes the interface shape in each cell as a straight line. The interface inclination and the volume-fraction on each cell find the interface surface area A_i. Eq. (3) estimates the mass flux at each mixture cell:

$$m'' = \frac{k_l\nabla T_i}{h_{fg}} \tag{3}$$

where k_l is the liquid thermal conductivity, h_{fg} is the latent heat of evaporation, and ∇T_i is the temperature gradient at the interface. The main challenge in Eq. (3) is the estimation of the temperature gradient at the interface. The interfacial temperature gradient requires a temperature reconstruction algorithm. The present work proposes a method to find the temperature gradient at the interface with a single liquid cell. The proposed approach (explained in Section 2.4) is easy to implement and improves the accuracy of the mass transfer estimation.

The momentum conservation equation Eq. (4) finds the pressure and the velocity on the liquid and vapor phases:

$$\rho\left[\frac{\partial\vec{v}}{\partial t} + (\nabla\cdot\vec{v})\vec{v}\right] = -\nabla p + \nabla\cdot\mu\nabla\vec{v} + \nabla\cdot\mu\nabla\vec{v}^T + S_M \tag{4}$$

where p is the pressure, μ is the viscosity, and S_M accounts for the surface tension effects. The continuous surface force (CSF) model proposed by Brackbill et al. [22] accounts for surface tension effects. Eq. (5) estimates the surface tension volume force:

$$S_M = \sigma\frac{\rho k\nabla F_v}{0.5(\rho_v + \rho_l)} \tag{5}$$

where σ is the surface tension. The interface curvature, k, is estimated as:

$$k = \nabla\cdot\hat{n} \tag{6}$$

Eq. (7) gives the unit vector normal to the interface $\hat{n}$:

$$\hat{n} = \frac{\nabla F_v}{|\nabla F_v|} \tag{7}$$

The energy conservation equation Eq. (8) provides the temperature for both liquid and vapor phases:

$$\frac{\partial T}{\partial t} + \nabla \cdot \vec{v} T = \nabla \cdot \alpha \nabla T + S_T \tag{8}$$

where α is the thermal diffusivity, and S_T is a source term. The source term fixes the temperature of the mixture cells to properly account for the sharp interface; this procedure is described in Section 2.5. Similar to other simulations with a sharp interface [15,16], no source term was added in the mixture cells to account for latent heat of evaporation. This is because the simulation considers the interface as a boundary condition. In addition, the simulation solves the energy equation on liquid and vapor cells and interpolates the temperature of the mixture cells. Latent heat effects should be considered only at the interface and not on the liquid or vapor phases. In the simulation, the temperature of the surrounding liquid changes in accordance to the interface temperature and position.

In Eqs. (4), (5) and (8), the density, viscosity, and thermal diffusivity are determined with the vapor volume fraction:

$$\vartheta = F_v \vartheta_v + (1 - F_v)\vartheta_l \tag{9}$$

where ϑ represents the density, viscosity, or thermal diffusivity. It is important to highlight that in the present work, the energy equation considers a sharp interface. The sharp interface is considered by fixing the temperature of the mixture cells. Fixing the temperature at the mixture cells means that a value is imposed at the mixture cells during the simulation. Ghost cell method fixes the temperature of the mixture cells based on the temperature of the surrounding liquid (for more details, see Section 2.3). Therefore, the properties used to estimate the temperature of the mixture cells are the properties of the liquid, and not a combination of liquid and vapor properties. By fixing the temperature of the mixture cells, errors that arise with the use of average properties on the mixture cells are eliminated. This technique increases the accuracy of the prediction of the temperature field near the interface, which is translated into more precise interface displacements.

2.2. Previous methods to estimate mass transfer

Two of the main difficulties in the simulation of multiphase flows with heat and mass transfer are: (i) accurate estimation of the mass transfer, (ii) proper characterization of the of the mixture cells. In bubble growth the mass transfer depends on the temperature gradient at the interface, see Eq. (3). In a computer simulation, the temperature gradient at the interface is difficult to determine because its direction (perpendicular to the interface) rarely intersects a cell center (the location at which the temperature is known in the simulation). Moreover, the mixture cells have undefined properties because they have two embedded phases.

The present section provides a brief description of previous methods used for the estimation of the temperature gradient at the interface, and the characterization of the mixture cells. Fig. 1 shows different approaches to find the temperature gradients used for the mass transfer computation.

Fig. 1(a) shows the method proposed by Ling et al. [16]. This approach injects one probe to define the closest distance between the interface and the mixture cell center. The temperature gradient is given by $\nabla T = (T_{i,j} - T_{sat})/d_1$, where $T_{i,j}$ is the temperature of the mixture cell and d_1 is the length of probe-1. To find $T_{i,j}$ the authors assumed that the temperature gradient at the interface was similar to the temperature gradient at the mixture cell center given by $\nabla T = (T_{i+1,j} - T_{i,j})/\Delta x \cdot n_x + (T_{i,j-1} - T_{i,j})/\Delta y \cdot n_y$, where n_x and n_y are the x and y components of the vector normal to the interface, respectively. The method requires the temperature of two neighboring cells and of the mixture cell. Kunkelman and Stephan [19] used a similar mass transfer model, but the authors used an interface temperature rather than the temperature of the mixture cell center. The main issue with this approach is that errors may appear when the interface is far from the mixture cell center.

Udaykumar and Shyy [23] used a biquadratic polynomial function of the form $T(x,y) = ax^2 + bxy + cy^2 + dx + ey + f$ to define the temperature field near the interface, see Fig. 1(b). The method injects two probes normal to the interface: probe-1 defines the closest distance between the interface and the mixture cell center, and probe-2 defines a temperature at a certain distance from the interface. The polynomial function finds the temperature at the tip of probe-2 x_{ref}, y_{ref}. The temperature gradient is given by $\nabla T = (T_{ref} - T_{sat})/d_2$, where d_2 is the length of probe-2. The method requires the temperature of six cells around the interface to define the coefficients of the polynomial. This approach has been adopted by a few other numerical works [20,24].

The method proposed by Sato and Niceno [15] finds the temperature gradient at the neighboring cell center, see Fig. 1(c). The method injects two probes to find the distance between the interface and the neighboring cell center along the x and y directions. The temperature gradient is given by $\nabla T = T_{x,(i,j)}\mathrm{n}_x + T_{y,(i,j)}\mathrm{n}_y$, where n_x and n_y are the x and y components of the unit vector normal to the interface. The temperature gradients in the x and y direction adopt a second order approximation given by $T_x = (-d_1^2 T_{sat} + (d_1^2 - \Delta x^2)T_{i,j} + \Delta x^2 T_{i+1,j})/(\Delta x d_1(\Delta x + d_1))$ (a similar approximation was used for T_y). When the interface does not intersect the line that connects the cell centers of the mixture and neighboring cells, the temperature gradient uses an interpolated mixture cell temperature. The method requires the temperature of two or three cells around the interface.

2.3. Previous methods to account for interface temperature

In the simulation of bubble growth, the interface is an internal boundary moving inside the mixture cells. The internal moving boundary influences the temperature of the neighboring cells (the cells that compute the mass flux). Therefore, the accuracy of the simulation primarily depends on the correct treatment of the moving interface. The technical literature distinguishes various methods to include the interface effects in the solution of the governing equation. Fig. 1(a) and (c), and Fig. 2 show the main approaches.

One approach considers effective conductivities at the faces of the mixture cell. Patankar [25] used the face between two cells as the interface between two different materials and estimated the heat flux with an effective conductivity. The effective conductivity assumes equal and continuous heat fluxes on both sides of the interface. In other words, the heat flux arriving to the interface was equal to the heat flux leaving the interface. Son and Dhir [26,27] and Son [28] applied a similar concept to find an effective conductivity at the interface, see Fig. 2(a). The interface conductivity was imposed at the faces of the mixture cell. This is a valid assumption as long as the distance between the interface and the face is short. In addition, special considerations are required for inclined interfaces. Recent studies have used a similar approach to account for the interface temperature [29–31].

An alternative approach imposes ghost values at the mixture cell, see Fig. 2(b). This approach interpolates the temperature of the mixture cells with the data on the phases and the interface temperature. Each phase has a corresponding interpolated or ghost

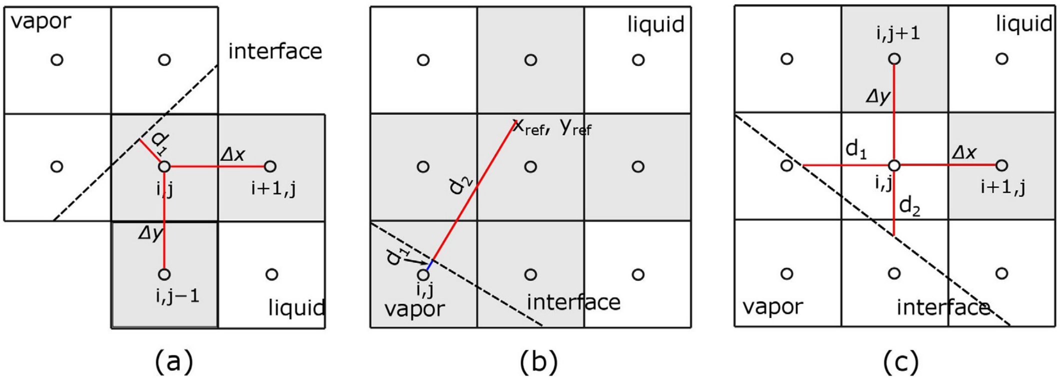

Fig. 1. Methods to compute the temperature gradient: (a) Ling et al. [16], (b) Udaykumar and Shyy [23] (c) Sato and Niceno [15]. Gray color highlights the cells needed for the estimation of the gradient.

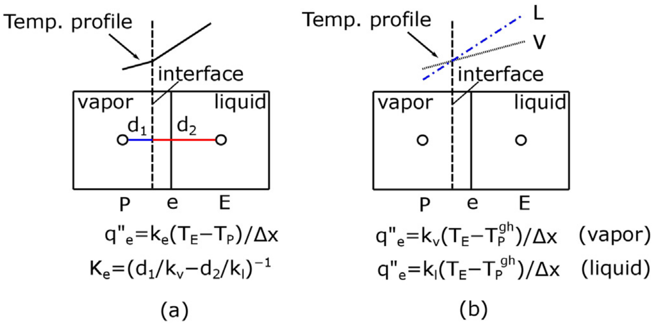
Temp. profile
interface
vapor
liquid
d_1
d_2
P e E
$q''_e=k_e(T_E-T_P)/\Delta x$
$K_e=(d_1/k_v-d_2/k_l)^{-1}$
(a)
Temp. profile
L
V
interface
vapor
liquid
P e E
$q''_e=k_v(T_E-T_P^{gh})/\Delta x$ (vapor)
$q''_e=k_l(T_E-T_P^{gh})/\Delta x$ (liquid)
(b)

Fig. 2. Methods to include interface in the fluxes of the mixture and neighboring cells. (a) Effective conductivity at the face, (b) ghost temperatures.

temperature at the mixture cell. The method ensures that the neighboring cells see proper temperatures on the mixture cell, which allows accurate gradients at the faces of the mixture cell. Gibou et al. [32] constructed a first order interpolation function of the form $T = ax + b$ to define ghost temperatures on the mixture cell. The temperature of the interface and of a neighboring cell determined the coefficients of the interpolation function. The method requires the distance between the interface and the neighboring cells. Ling et al. [16] found ghost values at the mixture cells with a mathematical expression. The derivation of the expression assumed that the temperature gradient at the interface was equal to the temperature gradient at the mixture cell center (see Fig. 1(a)). This assumption is valid as long as the interface is near the mixture cell center. Another alternative is to use a partial differential equation approach to define ghost values at mixture cells [33]. This approach considers a hyperbolic partial-differential-equation (PDE) that performs the extrapolation by considering the vector normal to the interface and the derivatives of the extrapolated variable. Tanguy et al. [34] considered a linear extrapolation with the PDE approach to find ghost values for liquid and gas temperatures and vapor mass fraction. The method requires the solution of an extra PDE and the communication of the new PDE with the governing equations.

Another method that includes the effects of the interface modifies the discretization schemes at the neighboring cells with the interface position and temperature. This method is independent of the mixture cell temperature. Sato and Niceno [15] found the distance between the neighboring cell centers and the interface along Cartesian directions, see Fig. 1(c). Second order discretization schemes determined the temperature gradient at the neighboring cell center based on the interface location. In the past, we implemented a similar approach in the simulation of planar interface evaporation, Perez-Raya and Kandlikar [35,36]. In that work, we used the interface as a face and adopted first order backward differences to estimate the temperature gradients at the interface. The method requires the distances between the interface and the neighboring cell center. In the case of using a simulation package, the method requires the elimination of the software procedures at the neighboring cells, and the integration of the modified discretization schemes, which is a complex task in 2D and 3D simulations.

2.4. Proposed method for mass transfer and mixture cell temperature

This work presents a method termed as OCASIMAT for modeling interfacial heat and mass transfer during bubble growth. The method determines the temperature gradient at the interface, which is needed for proper estimation of the mass transfer at the interface. Moreover, the method determines a fixed temperature value for the mixture cells, which is needed for proper representation of the interface.

The OCASIMAT finds the mass transfer and the fixed temperature of the mixture cells with the temperature gradient at the interface. Fig. 3 illustrates the procedures to determine the temperature gradient at the interface. Three probes that are perpendicular (or normal) to the interface are injected in the liquid phase. The objective of probe-1 is to locate the closest point on the interface from the mixture cell center, the objective of probe-2 is to identify the cell that is closest to the interface in the normal direction (the G-cell, where G comes from gradient). Probe-1 and probe-2 support the construction of probe-3. Probe-3 is special due to its orientation and its length; this probe is normal to the interface, and has a length such that the tip of the probe intersects a cell center on the liquid. After the injection of the three probes, mathematical expressions find the mass transfer through the interface and the fixed temperature at the mixture cells.

It is important to emphasize that most of the previous methods in the technical literature use probe-1 to identify the interface. Udaykumar and Shyy [23] used probe-2 to estimate the temperature gradient at a reference point, see Fig. 1(b). The method of Sato and Niceno [15] is independent of probe-1 and probe-2, but it requires horizontal or vertical probes to identify two points at the interface, see Fig. 1(c). Therefore, the proposed method adds only one extra probe (probe-3), which is defined based on probe-2 and on the vector normal to the interface.

The steps to inject the probes are described in the following lines:

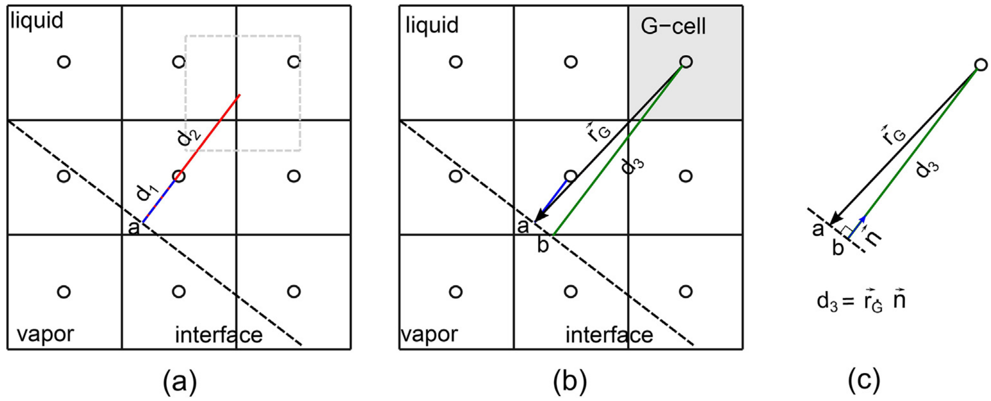

Fig. 3. Proposed once-cell algorithm to evaluate the temperature gradient at the interface. (a) Steps (1) to (4), (b) Steps (5) to (7), (c) alignment of G-cell center with normal vector.

Probe-1	(1) Identify a mixture cell for which the mass transfer and the fixed temperature value should be determined. (2) Inject probe-1. The origin of this probe is the mixture cell center, the orientation is normal to the interface, and the length is such that the tip intersects the interface. The point that intersects the probe with the interface is called “point a”.
Probe-2	(3) Inject probe-2. The origin of this probe is “point a”, the orientation is normal to the interface, and the length is d_2. (4) Explore a region around the tip of probe-2 to find the center of the cell that is nearest to the interface in the normal direction. The name of the nearest cell is “G-cell”
Probe-3	(5) Inject probe-3. The origin of this probe is the cell center of the G-cell, the orientation is normal to the interface, and the length is determined with steps (6) and (7). (6) Create a vector $\vec{r_G}$ connecting the center of the G-cell with the “point a” at the interface. (7) Determine the length of probe-3 by the projection of the vector $\vec{r_G}$ on the normal vector $\hat{n}$. The name of the intersection between probe-3 and the interface is “point b”.

In the procedure of injecting probes, the length of probe-2, d_2, in step 3 is such that the tip of probe-2 lies outside of the mixture cell. A proper length of probe-2 could be determined based on the size of the mixture cell. The tip of the probe lies outside the mixture cell if the length is larger than the maximum distance between two points on the periphery of the mixture cell. This distance is given by $\sqrt{\Delta x^2 + \Delta y^2}$, where Δx and Δy are the length of the cell in the x and y directions, respectively. Therefore, d_2 should be greater than $\sqrt{\Delta x^2 + \Delta y^2}$.

Another important factor is the region explored to find the nearest cell center in step 4. This step should ensure that only the nearest cell center is inside the perimeter of exploration. To perform this task, the center of the explored region should be the tip of the probe-2, and the length of the explored region could be Δx in the x-direction and Δy in the y-direction.

In step 7, Eq. (10) determines the length of probe-3:

$$d_3 = \vec{r_G} \cdot \hat{n} \tag{10}$$

where $\hat{n}$ is a unit vector that is normal to the interface. It is important to observe that the proposed method aligns the center of the G-cell with the normal vector. The projection of the vector $\vec{r_G}$ on the normal vector gives the distance (d_3) from the center of the G-cell to the interface along the normal direction, see Fig. 3(c). Therefore, it is valid to use d_3 to determine the temperature gradient. The only drawback is that the gradient ignores the center of the interface (point “a” in Fig. 3(b)). Instead, the gradient considers a point near the center (point “b” in Fig. 3(b)). In other words, the method translates the normal vector along the interface to align the gradient with the center of the G-cell. The translation assumes that the interface is a straight line. Such a translation might introduce inaccuracies, which could become significant with highly deformed interfaces. However, results reveal insignificant errors with uniform interfaces in bubble growth.

Another important fact is that the proposed approach requires structured meshes near the interface. This is a requirement to identify the G-cell. The simulation in the present work uses an interface reconstruction algorithm to define the interface. In addition, the present simulation considers the interface location to fix the temperature of the mixture cells. Under these conditions, the use of structure meshes is acceptable. Previous works have used unstructured grids with moving meshes to increase numerical accuracy by augmenting grid resolution near the interface [37,38]. In the case of simulations with unstructured meshes, the proposed method should be modified to account for the nonorthogonal cell shape.

After performing the proper injection of the three probes, the temperature at the extremes of the probes and the length of the probes determine the temperature gradients at the interface. These gradients form the basis of the proposed expressions to estimate the mass transfer and the fixed temperature value at the mixture cell.

Particularly, the approach requires the temperature gradients at points “a” and “b” (see Fig. 2). The gradient at points “a” and “b” are the difference of the temperatures at extremes of the probes divided by the length of the probes. Eq. (11) gives the temperature gradient at point-a, whereas Eq. (12) gives the temperature gradient at point-b:

$$\nabla T|_a = \frac{T_M - T_{sat}}{d_1} \tag{11}$$

$$\nabla T|_b = \frac{T_G - T_{sat}}{d_3} \tag{12}$$

where T_M is the fixed temperature value at the mixture cell center, which lies at the tip of the probe-1. T_G is the temperature at the center of the G-cell, which lies at the tip of the probe-3.d_1 and d_3 are the length of the probes 1 and 3, respectively. T_{sat} is the temperature of the interface, which depends on the fluid properties.

As it was previously stated, the mass transfer through the interface is proportional to the temperature gradient at the interface, see Eq. (3). The main difficulty in Eq. (3) was the estimation of the temperature gradient at the interface. The proposed method determines the temperature gradient at points “a” and “b” at the interface. Under this scenario, the mass transfer can be determined as:

$$m'' = \frac{k\nabla T|_b}{h_{fg}} = \frac{k}{h_{fg}}\frac{T_G - T_{sat}}{d_3} \tag{13}$$

In Eq. (13), probe-3 determines the temperature gradient. For the estimation of the mass transfer, probe-3 is more convenient than probe-1 because the tip of probe-3 lies outside the mixture cell. It is important to emphasize that the OCASIMAT estimates the mass transfer without temperatures of mixture cells, which allows a direct estimation of the mass transfer without interpolation functions.

The aspect that is missing is the estimation of the fixed temperature value at the mixture cell. The information on probe-1 and probe-3 provides the fixed temperature value at the mixture cell. The expression to estimate the temperature of the mixture cell assumes that the temperature gradients at points “a” and “b” are approximately equal. This assumption is valid if the distance between points “a” and “b” is small. Under this assumption, Eq. (14) determines the fixed temperature value at the mixture cell T_M.

$$T_M = T_{sat} + d_1\frac{(T_G - T_{sat})}{d_3} \tag{14}$$

The proposed method finds the fixed value at the mixture cell, T_M, and the mass transfer through the interface with the temperature gradient at the interface. Eq. (14) linearly interpolates T_M by using gradients of temperature along the normal direction.

The proposed approach assumes a linear temperature profile at the interface for the estimation of the interfacial temperature gradient and the mixture cell temperature. Such assumption is valid as long as the distance from the neighboring cell to the interface is small (of the order of the grid cell size). Numerical simulations of bubble growth usually assume a first order approximation (linear profile) to estimate the temperature gradient [13,14,16,20,21,23], whereas a few works have considered a second order backward approximation (quadratic profile) [15,24]. Esmaeeli and Tryggvason [20] investigated the effect of the order of approximation of the temperature gradient at the interface; the authors pointed out insignificant differences between first and second order. Shin and Choi [24] reported that linear and quadratic profiles give similar results as long as the cell center is at a certain distance (of the order of the grid cell size) from the interface. Therefore, the assumption of a linear temperature profile is acceptable.

To summarize the proposed method, two of the main difficulties in the simulation of multiphase flows with heat and mass transfer are: (1) the mass transfer through the interface, and (2) the characterization of the mixture cell. To address these difficulties, the proposed method injects three probes. The probes, which are perpendicular to the interface, connect the interface with the center of the mixture cell and with the center of the G-cell. The length of the probes and the temperatures at extremes of the probes determine temperature gradients at the interface. These gradients are the basis of the expressions that find the mass transfer through the interface Eq. (13) and the fixed temperature of the mixture cell Eq. (14).

2.5. ANSYS-Fluent customization

The numerical simulation of interfacial phase change was performed in ANSYS-Fluent, which solves the governing equations for mass, momentum, and energy conservation. In addition, the simulation software solves the volume-of-fluid (VOF) tracking interface equation. At the present, the numerical solver is in need of a mass transfer model that considers the heat flux at the interface. Moreover, the numerical software assumes average properties at the mixture cells to solve the energy transport equation, which lead to inaccurate interface displacements [36]. In the present work, the software was customized to ensure an accurate heat and mass transfer at the interface. User-Defined-Functions (UDFs) identified the mixture and G-cells, fixed the temperature of the mixture cells, and imposed the mass transfer.

Fig. 4 shows a flow diagram of the procedures in the simulation; the simulation starts with initial conditions for VOF and temperature. UDFs fix the temperature of the mixture cells before the software solves the governing equations for momentum and energy conservation. The software iterates until the convergence criteria is satisfied (the convergence criteria is 10^{-8} for all the equations). After convergence is satisfied, UDFs estimate the mass source term with the temperature gradient at the interface. After UDFs declare the mass source, the software advances the interface by solving the VOF equation. In the next time step, the process repeats with the new interface position.

In the simulation, the UDF Define-Init sets the initial conditions of volume-fraction and temperature field. The UDF Define-Adjust was used to identify the mixture and G-cells, to estimate the mass transfer, and to fix the temperature of the mixture cells. The UDF Define-Source declared source terms in the energy equation. The UDF Define-Diffusivity defined the thermal diffusivity with Eq. (9). The UDF Define-Mass-Transfer declared the mass transfer in the VOF equation. The UDF Execute-at-the-End cleaned the variables, including the User-Defined-Memories that passed information between UDFs. Perez-Raya and Kandlikar [36] provided a detailed explanation on writing UDFs to account for heat and mass transfer at the interface.

The simulation uses the sharp interface VOF model provided by ANSYS-Fluent to track the interface. The interface is not physically present in the simulation, but the VOF values on the interface cells give information on the normal vector and interface location. The software macros C_VOF_G and Get_surface_distance retrieved the normal vectors and the distance from the interface to the mixture cell center, respectively. For details on the solution of the VOF equation and the use of the macros, the reader is directed to ANSYS-Fluent documentation.

The numerical software defines an average thermal diffusivity at the cells with an interface. However, UDFs eliminate the effect of average properties at the mixture cells by interpolating the temperature on these cells. At each iteration, the UDF Define-Adjust calculates the mixture cell temperature with Eq. (10), and the UDF Define-Source declares a source term to fix the temperature of the mixture cell.

In the simulation, the large coefficients method fixes the temperature of the mixture cells. Patankar [25] proposed this method to fix the temperature of internal cells in a computational domain. Sun et al. [14] used the large coefficients method to fix a saturation temperature at the mixture cells in ANSYS-Fluent in the simulation of film boiling. In the present work, the large coefficients method imposes a value of T_M at the mixture cells. The source term in the energy equation is defined as:

$$\begin{aligned} &S_T = CT_M - CT_{i,j} \textit{ At the mixture cell } i,j \\ &S_T = 0 \textit{ At other cells} \end{aligned} \tag{15}$$

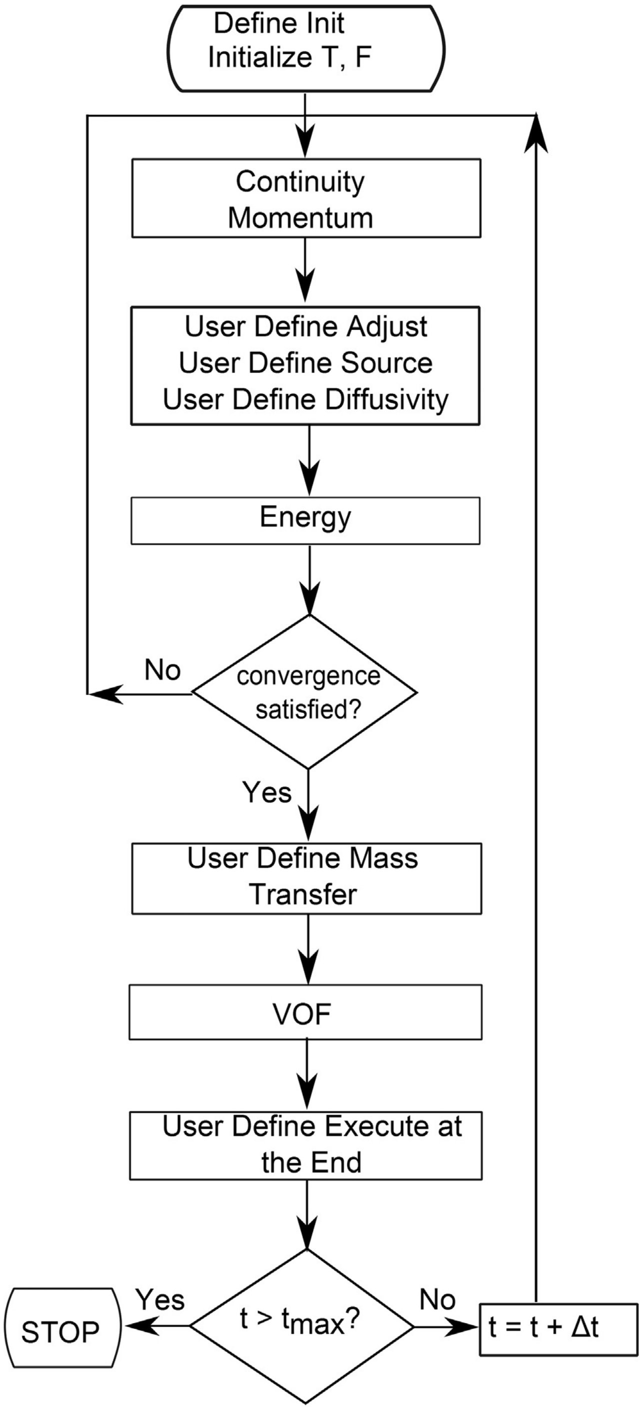

Fig. 4. ANSYS-Fluent procedures and customization.

where C is a large coefficient 10^{30}, and $T_{i,j}$ is the temperature of the mixture cell provided by the numerical software at each iteration. The main advantage of the large coefficients method is that it is independent of the software procedures. This is important because in ANSYS-Fluent the source codes that solve the governing equations are unavailable. Therefore, the correction of the software procedures at the neighboring cells is not trivial.

It is worth to observe that the large coefficient method imposes an internal boundary condition without damaging energy conservation. To explain this method, consider the discretization of the energy equation in 1 D in a cell: $a_{i,j}T_{i,j} = a_{i-1,j}T_{i-1,j} + a_{i+1,j}T_{i+1,j} + S_T$, where a_i are coefficients and T_i are temperatures of the discretized cell and surrounding cells [39]. The substitution of Eq. (15) in the discretization equation gives $T_{i,j} = T_M$ if the coefficients C are big enough. Therefore, the units of Eq. (15) might not necessarily match the units of Eq. (8) since the C coefficients cancel all the terms in Eq. (8) to impose a temperature on the mixture cells.

It is important to note that the simulation fixes the temperature of the interface cells in every iteration. The simulation performs multiple iterations at each time step until it reaches a convergence criterion. We defined a convergence limit of 10^{-8}, which is small enough to ensure energy conservation. At each iteration, Eq. (10) determines the temperature of the mixture cells based on the temperature of the interface and of the G-cell, and UDFs impose this temperature on the mixture cells. In the new iteration, the solution of the energy equation considers the determined mixture cells temperature. The process repeats until the temperatures of the previous and new iteration are similar (convergence limit is satisfied). Usually, no more than five iterations reach the convergence criterion.

Another important fact is that the simulation ignores the energy equation used by ANSYS-Fluent. Instead, we adopted a User-Defined-Scalar (UDS) equation, which is a general transport equation. The main advantage of a UDS is that it provides access to the customization of the transient, the convective, and the diffusive terms.

2.6. Identification of mixture and G-cells

In the simulation of bubble growth, we observed that the declaration of mass source terms in the VOF requires special procedures to maintain the interface shape. Mass transfer in cells with a small interface could lead to interface deformations due to errors in the normal vector. These errors could damage the interface sharpness. To maintain a sharp interface, the present work proposes an algorithm to identify mixture cells. The algorithm ensures that the mixture cells are located next to cells filled with vapor.

Consider the interface displayed in Fig. 5(a). The small interface in cells D and E might result in wrong normal vectors on these cells. These errors could damage the interface shape. The proposed identification algorithm declares mass transfer only in cells A, B, and C. To identify the cells with mass transfer, the proposed algorithm makes a loop through the faces of the cells in the computational domain, and uses connectivity macros to retrieve the volume-fraction of two adjacent cells (e.g. cells $c0$ and $c1$ in Fig. 5b)). The algorithm declares mass transfer if a cell has liquid volume-fraction greater than zero ($F_l > 0$) and if the adjacent cell has vapor ($F_l = 0$). In other words, if $(F_{l,c1} - F_{l,c0}) > 0$ and $F_{l,c0} = 0$, then $c1$ is a mixture cell. This method prevents the identification

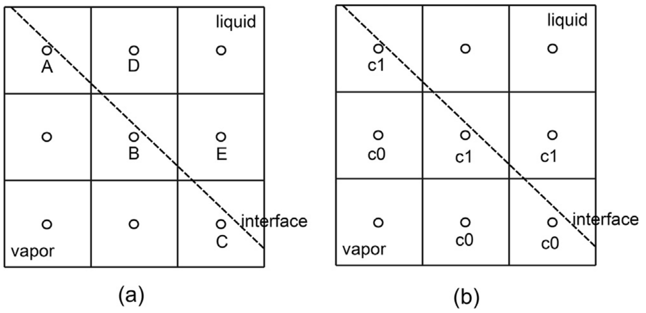

Fig. 5. Identification of mixture cells in simulation to maintain a sharp interface. (a) Cells A, B and C are mixture cells; cells D and E are neighboring cells, (b) two adjacent cells c0 and c1 in the computational domain.

of cells D and C as mixture cells. To avoid the loss of mass, the simulation in the present work adopts a more sophisticated algorithm that adds the small length of the interface in cells D and E to the length of the interface in cells A and B, respectively.

Similarly, the G-cells were identified with connectivity macros and the mixture cell. A cell is a neighboring cell if it lies next to a mixture cell and has liquid volume-fraction greater than zero. A loop over the four faces of the neighboring cell identifies the G-cell based on the position of probe-2 and the coordinates of the cells around the neighboring cell.

3. Theoretical models for validation

In the simulation of interfacial heat and mass transfer, the validation asserts that the mass transfer and the interface effects are properly captured by the numerical model. Techniques to validate a phase change simulation include theoretical models for temperature and interface displacement, and experimental bubble shape or heat transfer data. Both approaches are generally accepted, but the theoretical models account for most of the physics in the phase change process and at the same time provide a unique solution. Therefore, theoretical models are commonly preferred for the validation of boiling simulations.

The theoretical models for validation are Stefan problems for a planar interface and spherical bubble growth, see Fig. 6. Stefan problems test the accuracy of the mass transfer computation and the inclusion of the interface effects in a simplified environment such as one-dimensional heat flow. Spherical bubble growth tests the accuracy of the phase change phenomena in an environment with curved interfaces. For a literature review on Stefan problems, please refer to Perez-Raya and Kandlikar [36]. The following subsections describe the theoretical framework of these problems.

3.1. Planar interface evaporation

Various works have used Stefan problems to validate simulations of phase change [1,10,14,20,40–42]. Son and Dhir [26] numerically analyzed the transition of bubble patterns in film boiling near critical pressures. Errors in interface displacement of Stefan problems with a saturated liquid and a superheated vapor were significant at the beginning of the simulation (from 0 to 0.1 s), but they decreased to 2% at longer times. Kunkelmann and Stephan [2] simulated pool boiling of HFE-7100. Minimal differences were observed between theory and the simulation of saturated vapor and superheated liquid. Perez-Raya and Kandlikar [35] simulated Stefan problems with heat transfer in both phases and with large density ratios. Simulations and theory showed excellent agreement in terms of interface displacement and temperature distribution for all periods of time.

A schematic diagram of planar interface evaporation is shown in Fig. 6(a). In this problem heat flows only in the x-direction. A wall and the interface constrain the vapor, and the liquid is unconstrained. The mass transfer is from liquid to vapor. During phase change, the vapor expands because the vapor has a lower density relative to the liquid. The vapor expansion induces motion in the liquid, which results in a convective transport. The boundary conditions are $T = T_A$ at $x = 0$, $T = T_B$ at $x = \infty$, and a moving interface with $T = T_{sat}$. The 1D heat diffusion equation governs the heat transfer in the vapor, whereas the 1D advection–diffusion equation governs the heat transfer in the liquid.

During the phase change process, the interface thermally interacts with the two phases. Initially, the slab has only liquid with temperature $T(x,0) = T_B$. Suddenly, at 0 s the interface appears at $x = 0$ with a temperature $T = T_{sat}$. The thermal interaction between the liquid and the interface creates a thermal film on the liquid. The interfacial heat flux generates a mass flux, and therefore the interface moves. In addition, $T = T_A$ at $x = 0$ could be different from the interface temperature T_{sat}; in such a case, both phases inject a heat flux to the interface, and the mass transfer depends on the net heat flux. Perez-Raya and Kandlikar [36] defined eight different cases: Sup-V/Sat-L, Sat-V/Sup-L, and Sup-V/Sub-L, Sup-V/Sup-L, where each of them neglects or considers density effects. In these cases, Sup, Sub, and Sat stand for superheated ($T > T_{sat}$), subcooled ($T < T_{sat}$), and saturated ($T = T_{sat}$), respectively. In addition, V and L stand for vapor and liquid, respectively. These cases represent evaporation under different scenarios.

The theoretical solutions for the interface position $X(t)$ and the temperature distribution $T(x,t)$ on the phases are given by Eqs. (16)–(19).

$$X(t) = \xi\sqrt{t} \tag{16}$$

$$T(x,t) = T_A - (T_A - T_{sat})\frac{\operatorname{erf}\left(\frac{x}{2\sqrt{\alpha_v t}}\right)}{\operatorname{erf}\left(\frac{\xi}{2\sqrt{\alpha_v}}\right)} \quad 0 \leqslant x \leqslant X(t) \tag{17}$$

$$T(x,t) = T_B - (T_B - T_{sat})\frac{\operatorname{erfc}\left(\frac{x}{2\sqrt{\alpha_l t}} - \frac{\xi}{2\sqrt{\alpha_l}}(1-\phi)\right)}{\operatorname{erfc}\left(\frac{\xi}{2\sqrt{\alpha_l}}\phi\right)} \quad X(t) \leqslant x \leqslant \infty \tag{18}$$

$$\frac{\xi\sqrt{\pi}}{2}h_{fg} = \frac{\sqrt{\alpha_v}c_v(T_A - T_{sat})}{\exp\left(\frac{\xi^2}{4\alpha_v}\right)\operatorname{erf}\left(\frac{\xi}{2\sqrt{\alpha_v}}\right)} + \frac{\sqrt{\alpha_l}c_l(T_B - T_{sat})}{\phi exp\left(\frac{\xi^2}{4\alpha_l}\phi^2\right)\operatorname{erfc}\left(\frac{\xi}{2\sqrt{\alpha_l}}\phi\right)} \tag{19}$$

In Eqs. (17)–(19), ϕ is the density ratio ρ_v/ρ_l, c is the specific heat, α is the thermal diffusivity, and ξ is a constant determined with

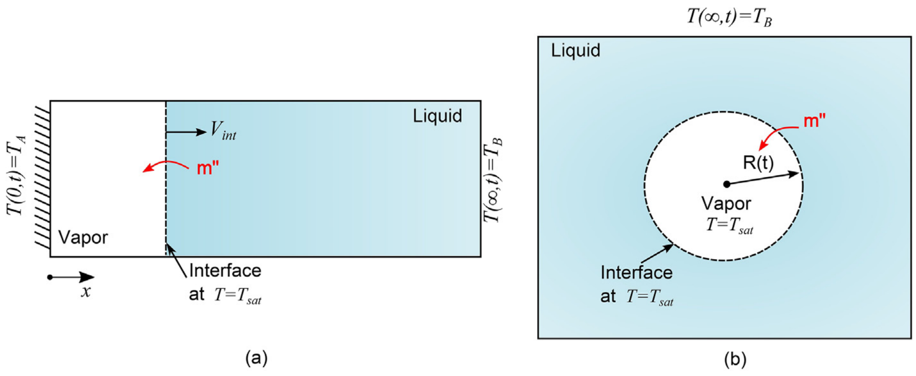

Fig. 6. Fundamental problems of phase change: (a) Planar interface evaporation Stefan problem, (b) spherical bubble growth.

Eq. (19). The subscripts v and l stand for liquid and vapor phases, respectively.

Eq. (17) provides the vapor temperature distribution, whereas Eq. (18) provides the liquid temperature distribution. It is important to emphasize that previous numerical works on planar interface evaporation have used semi-analytical solutions in cases with a superheated liquid [2,15,43]. However, Eq. (16)–(19) represent exact solutions that apply for all possible cases including superheated liquid [36,44].

3.2. Spherical bubble growth

Spherical bubble growth tests the numerical parameters related to the interface curvature. These parameters include surface tension, normal vector, interface area, and mass transfer. A schematic diagram spherical bubble growth is shown in Fig. 6(b).

The bubble growth process in uniformly superheated liquid has been well stablished in the technical literature [45–47]. Plesset and Zwick [45] analyzed spherical bubble growth in a uniformly superheated liquid. The authors distinguished that initially bubble growth occurs due to inertia effects; in this region, the pressure difference at the interface is significant and the saturation temperature decreases. After a certain time, the cooling effect at the interface governs the bubble growth rate; in this region, the pressure difference is negligible and saturation temperature slightly changes. In the formulation, the momentum equation related the pressure difference at the interface with the interface velocity. An approximated equation found the interface temperature as a function of the bubble radius and interfacial heat flux. Temperature and pressure in the systems were related by a linear function. A source term added heat into the liquid to initiate bubble growth. The authors derived and expression to estimate the bubble radius and liquid temperature in the diffusion controlled region by assuming a limit in the interface temperature.

Scriven [47] analyzed spherical bubble growth in pure liquids and in binary solutions. The governing equations were energy and species for temperature and concentration distribution, respectively. Mass transfer model included normal gradients of concentration and temperature at the interface. The boundary conditions assumed a constant interface concentration and temperature. The initial conditions had a small bubble immersed in an infinite mass of liquid with uniform temperature and composition. Initially, the liquid temperature was higher than the saturation temperature. The author pointed out that inertia and viscous are significant only during the early expansion of the bubble nucleus, and assumed similar pressures on vapor and liquid phases. The author derived exact solutions for bubble radius, temperature, and fluid composition in the diffusion-controlled region.

Mikic et al. [46] theoretically analyzed spherical bubble growth in the inertia-controlled and the diffusion-controlled regions. Mechanical energy conservation related vapor and liquid pressures with the interface velocity. The authors used an approximated equation to relate the saturation temperature to the bubble radius. Clausius-Clapeyron equation related the pressure difference with the temperature difference at the interface. The final solution predicted a different dependence of the bubble radius with time in each region: a linear dependence in the inertia-controlled region, and a square-root dependence in the diffusion-controlled region.

Computer simulations of bubble growth usually neglect inertia effects. Computer simulations distinguish two cases of spherical bubble growth. The first case is an adiabatic bubble where the mass flux is prescribed independently of the liquid temperature. The simulation of adiabatic bubble evaluates errors on mass conservation and momentum transport. The second case is a non-adiabatic bubble where the mass flux is prescribed with the temperature gradient at the interface. Non-adiabatic bubble growth quantifies errors on the computation of the mass transfer and on the solution of the energy equation with a moving curved interface.

The simulation of adiabatic bubble growth ascertains the proper computation of the interface curvature and surface tension. Tanguy et al. [34] simulated the evaporation of a two-dimensional drop with a prescribed mass flux. Jump conditions included the interface effects in the Navier-Stokes equations. A projection method extended the liquid velocities into the gaseous phase and vice versa. The velocity fields were extended with a ghost pressure. Results showed that the computed temporal mass variation of the vaporizing drop and the exact solution were the same in a 1 ms interval. In a different work, Tanguy et al. [48] simulated adiabatic spherical bubble growth with a constant mass flux to compare whole domain and jump continuous formulations. The whole domain formulation smeared the interface across two or three computational cells, and the jump continuous formulation adopted ghost cells for the velocity. The simulation had a 2D axisymmetric domain with an initial bubble radius of 1 mm. Relative to the exact solution of the bubble radius, the whole domain formulation had 25% of error, and a continuous jump had less than 1% of error. The authors concluded that average velocities at the interface in the whole domain formulation and spurious currents lead to wrong interface displacements.

Non-adiabatic spherical bubble quantifies errors on the mass transfer and on the interface effects. Son [28] simulated spherical bubble growth with the level-set method. A volume correction step ensured mass conservation. The computed bubble maintained a spherical shape and the growth rate was equivalent to Scriven's type solution [47]. Tanguy et al. [48] simulated spherical bubble growth, the simulation accurately predicted growth rates relative to Scriven's theory with both a simple and a divergence-free velocity extrapolation. Relative to the linear extrapolation, a quadratic extrapolation of the thermal field improved the accuracy. Sato and Niceno [15] simulated a spherical bubble and compared their results against Scriven's theory [47]; the accuracy of the simulation improved with smaller cells (grid cell sizes of 1 μm or less). There are other works that have simulated non-adiabatic spherical bubble growth [2,3].

In bubble growth, the mass of liquid evaporated in a time interval is given by the mass flux times the surface area $m = m'' A_s dt$. At the same time, the mass of vapor generated is the interface displacement times the surface area and the vapor density $m = \rho_v A_s dr$. Therefore, with a constant mass flux (adiabatic bubble growth), the theoretical expression for the radius as a function of time is given by Eq. (20).

$$R(t) = R_0 + \frac{m''}{\rho_v} t \tag{20}$$

where R_0 is the initial bubble radius, and m'' is the interfacial mass flux.

In non-adiabatic bubble growth, the mass flux depends on the temperature gradient at the interface. The temperature gradient at the interface changes with time due to the thermal interaction between the interface and the liquid. Scriven [47] theoretically analyzed spherical bubble growth in uniformly superheated liquid. The theory is based on the solution of the advection–diffusion equation in spherical coordinates. The solution uses a similarity variable. One important assumption is that the saturation temperature remains constant during the bubble growth.

Scriven's theory [47] considers an initial small spherical bubble immersed on liquid. Initially, the vapor and the interface have a temperature $T = T_{sat}$ and the liquid has a temperature $T(r, 0) = T_B$. As time progresses, the thermal interaction between the liquid and the interface creates a thermal film in the liquid around the interface. The thickness of the thermal film increases with time due to conduction heat transfer in the liquid. Scriven's solution assumes that the bubble radius is given by Eq. (21) [47].

$$R(t) = 2\beta\sqrt{\alpha_l t} \tag{21}$$

Eq. (22) gives the temperature distribution in the liquid side.

$$T(r,t) = T_B - (T_B - T_{sat})\frac{\int_s^\infty \lambda^{-2} exp(-\lambda^2 - 2\varepsilon\beta^3\lambda^{-1})d\lambda}{\int_\beta^\infty \lambda^{-2} exp(-\lambda^2 - 2\varepsilon\beta^3\lambda^{-1})d\lambda} \tag{22}$$

The vapor phase remains at T_{sat} during the bubble growth. In Eq. (22), T_B is the liquid temperature at $r = \infty$, $\varepsilon = 1 - \rho_g/\rho_l$, λ is a symbolic variable, and s is the similarity variable defined as:

$$s = \frac{r}{2\sqrt{\alpha_l t}} \tag{23}$$

β is a growth rate constant obtained by solving Eq. (24).

$$2\beta^3\left[h_{fg} + (T_B - T_{sat})(c_l - c_g)\right]\int_\beta^\infty \lambda^{-2} exp(-\lambda^2 - 2\varepsilon\beta^3\lambda^{-1})d\lambda = \frac{c_l(T_B - T_{sat})\rho_l/\rho_g}{exp(\lambda^2 + 2\varepsilon\beta^2)}. \tag{24}$$

It is important to highlight that Eqs. (21)–(24) are equivalent to Scriven's solution. The only difference is that Eqs. (21)–(24) ignore dimensionless parameters and where rearranged to get more discernible expressions.

4. Results and discussion

This section contains two different subsections. Section 4.1 discusses the accuracy of the proposed methods to estimate the mass transfer and the mixture cell temperatures, and Section 4.2 discusses the application of the proposed approach for the simulation of planar interface evaporation and spherical bubble growth.

4.1. Accuracy of the proposed method

This section compares the accuracy of the proposed one-cell algorithm for sharp interface and mass transfer (OCASIMAT) against other approaches in the technical literature. In addition, an analysis is presented on the accuracy of the estimation of the mass transfer and mixture cell temperatures along a circular interface.

Section 4.1.1 compares the proposed method to determine the temperature gradient against different models in the technical literature. The comparison considers three different interface cells located at θ equal to 5, 20, and 40°, where θ is the angle of the normal vector. Section 4.1.2 shows the performance of the proposed method for all the interface cells located in the range $0^\circ \leqslant \theta \leqslant 90^\circ$. The results in Section 4.1.2.1 evaluate the accuracy of the proposed method to determine mass transfer or temperature gradient at the interface. In the proposed method, Eq. (13) gives the interfacial temperature gradient. The results in Section 4.1.2.2 evaluate the accuracy of the proposed method to estimate the ghost cell temperature or temperature of the mixture cell. In the proposed method, Eq. (14) gives the ghost cell temperature.

4.1.1. Comparison with previous mass transfer models

OCASIMAT improves the accuracy on the computation of the interfacial mass transfer in bubble growth. The method applies for interfaces with any orientation and position. The method is free of interpolation functions since it ignores the temperature of the mixture cell. To provide evidence of these facts, this section compares the proposed model for mass transfer Eq. (13) against the methods proposed by Ling et al. [16], Udaykumar and Shyy [23], and Sato and Niceno [15], see Fig. 7. The comparison evaluates the accuracy of each method on determining the temperature gradient needed for the mass transfer computation.

To perform this task, ideal scenarios were recreated at three different locations along the interface of a spherical bubble. The radius of the spherical bubble is $R_0 = 0.1$ mm, and the scenarios lie at angles of inclination 5, 20 and 40° along the interface. The components of the scenario are the interface and nine cells. The

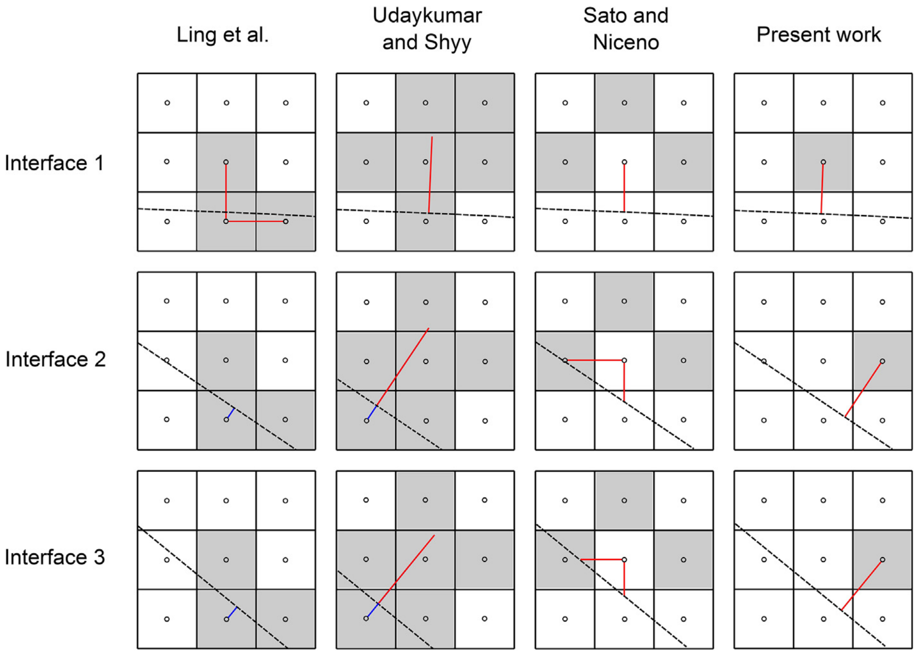

Fig. 7. Diagrams for the estimation of ∇T of three different interfaces. Methods adopted for the evaluation: Ling et al. [16], Udaykumar and Shyy [23], Sato and Niceno [15], present work.

nine cells represent the computational cells that would appear in the estimation of the temperature gradient in the simulation. The temperature of the cells is the theoretical temperature field, which was determined with Eq. (22) and with the radial position of the cell centers. The temperature field corresponds to a growing spherical bubble with a superheat level of 5 K and with fluid properties of water at 1 atm. The size of the cells is 1 μm.

Fig. 7 shows the three different recreated scenarios for each method. The diagrams use gray color and lines to highlight the cells and probes needed for the estimation of the temperature gradient. Section 2.1 provided an explanation of the particular procedures on each method. It is important to observe that the methods of Ling et al. [16] and Udaykumar and Shyy [23] use the temperature of the mixture cell, which represents a disadvantage since the temperature of this cell requires an extra interpolation procedure. Sato and Niceno [15] proposed a method that estimates the temperature gradient by taking into account the interface temperature and position, but the temperature gradient is estimated at the neighboring cell center rather than at the interface. Moreover, in some scenarios the method of Sato and Niceno should use interpolation procedures to find the temperature of the adjacent cells that have an interface (see for instance the scenario interface-2). The proposed method ignores the mixture cell and estimates the temperature gradient at the interface. In addition, the proposed method requires the control of fewer cells around the interface, which reduces the complexity of computing the mass transfer.

Table 1 shows the results of the estimation of the temperature gradients with the various approaches. The theoretical temperature gradient is 1.58×10^6 K/m, which corresponds to the derivative of Eq. (22) evaluated at $r = R_0$.

The results in Table 1 indicate that the method of Udaykumar and Shyy [23] has a relative error in the range of 5.3% to 5.8%. We observed that a similar error was obtained if the temperature at the tip of the probe was estimated with the theoretical equation rather than the polynomial function. By using theoretical equations, we observed that the error was reduced to 1.2% if a second order backward approximation was adopted. Therefore, the 5% error is mainly due to the adoption of a first order backward approximation for the estimation of the temperature gradient. The method of Udaykumar and Shyy [23] can determine the interfacial temperature gradient with a second order approximation. However, this task requires the identification of more cells around the interface. Another important fact in Table 1 is that the method of Udaykumar and Shyy provides equivalent temperature gradients with interface 1, 2 and 3. This is because the method determines the gradient with a probe of equal length on each interface analyzed.

The 38% error with interface-1 indicates that the method of Ling et al. [16] is inappropriate for interfaces that have a small angle of inclination. This is because the method considers the temperature of cells in x and y directions, while the interface is almost horizontal. The authors proposed alternative procedures to reduce the error on these interfaces. The results also show that the method proposed by Ling et al. is minimal (1.98%) with interface-2, but it increases to 13.39% with interface-3. The reason for this trend is that the method estimates the temperature gradient at the mixture cell center. Therefore, the minimal error is because interface-2 is closer to the mixture cell center relative to interface-3. The results indicate that the assumption of the temperature gradient at the mixture cell center may lead to significant errors when the interface is far from the cell center.

The method proposed by Sato and Niceno [15] shows errors of the order of 20% with interface-1 and interface-2, and the error is minimal and equal to 1.26% with interface-3. The error is significant with interface-1 and interface-2 because these interfaces are far from the neighboring cell center, which is the location at which the temperature gradient is estimated. The small error with interface-3 is due to the short distance between the interface and the neighboring cell center. Also, the estimation assumes a second order approximation, which reduces the magnitude of the error.

The one-cell method proposed by the present authors has an accuracy that is better than the more advanced method of Udaykumar and Shyy [23]. The method of Udaykumar and Shyy has a relative error in the range of 5.8–5.3% whereas the proposed OCASIMAT has a relative error is in the range of 1.79–4.17%. In addition, different to the methods of Ling et al. [16] and Sato and Niceno [15], the accuracy is maintained for all the analyzed interfaces with the proposed method. The error is only 1.79% with interface-1, and the error increases to 4.17% with interface-3. The variation is due to the change in length of the probe used for the estimation of the gradient. In OCASIMAT, the length of the probe that finds the gradient is defined by the distance from the G-cell center to the interface, which is not always the same in all the mixture cells. However, it is important to emphasize that the simulation of bubble growth indicates that these small variations have a null influence on the interface shape.

Table 2 presents a brief description of the main features of the various analyzed approaches. The information in the table shows that the proposed method is more accurate than the actual methods in the technical literature. Only OCASIMAT and the method of Udaykumar and Shyy [23] estimate the temperature gradient at the interface, which helps to improve the accuracy of the mass transfer computation. Other methods assume a temperature gradient at a cell center. Another important fact is that the proposed method requires the identification of only one computational cell in the phase, which reduces the complexity of computing the mass transfer. Besides, the proposed method is free of interpolations functions since it ignores the mixture cell temperature. Moreover, the analysis showed that the proposed method applies for interfaces of different orientation and position. These characteristics provide strong evidence of the importance of the proposed method in simulations with interfacial mass transfer.

Table 1
Computation of temperature gradients with various methods. The units are MK/m. The values in parenthesis indicate the relative error.

Method	Interface-1	Interface-2	Interface-3
Ling et al. [16]	0.921 (38%)	1.478 (1.98%)	1.306 (13.39%)
Udaykumar and Shyy [23]	1.422 (5.8%)	1.428 (5.3%)	1.428 (5.3%)
Sato and Niceno [15]	1.144 (24.1%)	1.129 (25.1%)	1.489 (1.26%)
Present work	1.481 (1.79%)	1.470 (2.5%)	1.445 (4.17%)

Theoretical temperature gradient 1.580 MK/m.

4.1.2. Accuracy of OCASIMAT along the interface

4.1.2.1. Estimation of mass transfer along a spherical interface. This section shows the accuracy of the proposed method to estimate the mass transfer along an evaporative interface. The main objective is to analyze the performance of the method with interfaces of different orientations and with domains of various grid cell sizes.

To perform this task, we evaluated the temperature gradients along the periphery of a 2D-axisymmetric bubble. The computation domain had a bubble of a specific radius. The bubble patch had reconstructed interfaces. With the initial patch, an algorithm identified the G-cells and retrieved the corresponding coordinates. These coordinates and Eq. (22) provided the G-cell temperature, which determined the numerical temperature gradient ∇T_{num} with Eq. (12).

The analysis considered two bubble radii of 0.1 mm and 1 mm. For a 0.1 mm, the length of the computational domain was 0.15 mm and the analyzed grid cell sizes were 1, 0.6, and 0.2 μm.

Table 2
General terms of the various approaches for the computation of mass transfer.

Method	Relative error (%)[a]	Location of ∇T estimation	Number of cells involved	Requires interpolation functions
Ling et al. [16]	2.0–38	Mixture cell center	3	Yes
Udaykumar and Shyy [23]	5.3–5.8	Interface	6	Yes
Sato and Niceno [15]	1.3–24.1	Neighboring cell center	3	Yes
Present work	1.7–4.2	Interface	1	No

[a] Determined with a spherical bubble of radius 0.1 mm. Δs = 1 μm, water properties 1 atm.

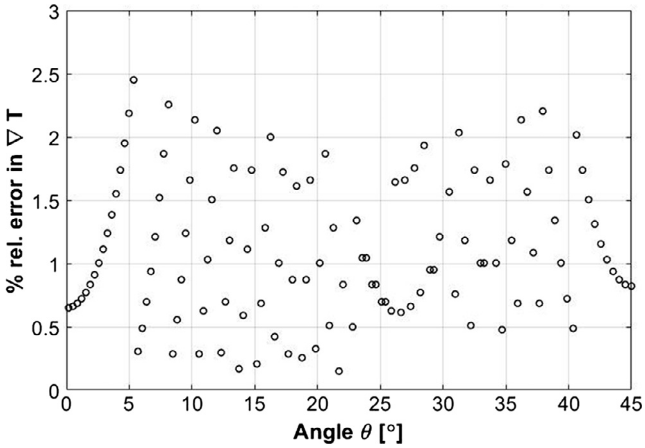

Fig. 8. Accuracy on the estimation of interfacial ∇T with the proposed method. Bubble radius 0.1 mm, grid cell size 0.6 μm.

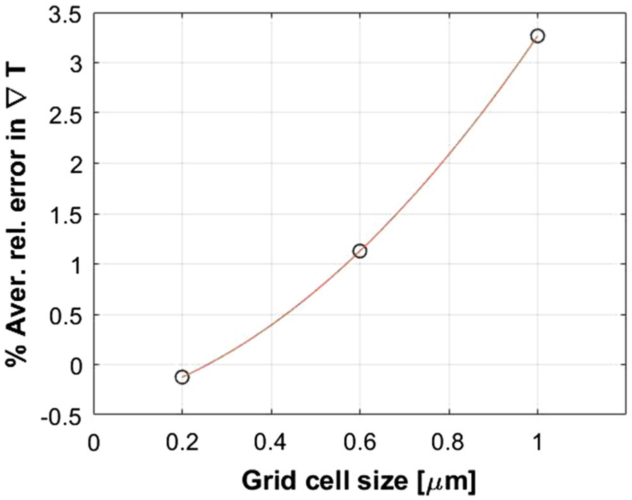

Fig. 9. Accuracy of the proposed method with various grid cell sizes. Bubble of 0.1 mm bubble radius.

For a radius of 1 mm, the length of the computational domain was 1.5 mm and the analyzed grid cell sizes were 10, 6, and 2 μm.

Fig. 8 shows the relative error of the temperature gradient on each G-cell along the interface. In the plot, θ is the angle measured from the vertical symmetrical axis along the clockwise direction. The plot shows results in a range from 0 to 45°, and similar results were observed in the range 45 to 90° with a symmetric line at $\theta = 45^\circ$. The results correspond to a bubble radius of 0.1 mm and to a grid cell size of 0.6 μm. The relative error is equal to $(\nabla T_{th} - \nabla T_{num})/\nabla T_{th} \times 100$, where the theoretical temperature gradient is given by the derivative of Eq. (22) evaluated at the radial location of the interface.

Fig. 8 shows that the relative error follows an oscillatory trend along θ. The range of the relative error is 0.2 to 2.5%. From 0 to 5° the relative error increases and the distance between the G-cell center and the interface (length of probe-3 d_3) is proportional to θ. This is because the radial positon of the G-cells remains constant, while the interface bends towards central axis. Around 5°, the relative error changes from 2.5% to 0.3% since the G-cell moved one cell below. A similar behavior occurs from 5 to 8°, the radial position of the G-cell remains constant, and the relative error augments because d_3 increases. These results show that the relative error is proportional to the length of probe-3. In the proposed approach, the maximum length of probe-3 is $d_3 = \sqrt{(1.5\Delta s)^2 + (1.5\Delta s)^2}$, which occurs when the interface is oriented at 45° and lies near the corner of the mixture cell. The results indicate that with a grid cell size of 0.6 μm, the maximum relative error is 2.5%. This error is small and might insignificantly influence the bubble growth rate.

Fig. 9 shows the average relative errors in the estimation of the interfacial temperature gradient. The analysis considered a bubble with a radius of 0.1 mm and with grid cell sizes of 1, 0.6, and 0.2 μm. The average relative error is $\sum_0^n \% rel.error_i/n$, where n is the total number of mixture cells.

Fig. 9 shows a reduction in the average relative error as the grid is refined. The accuracy follows a quadratic trend despite the fact that the temperature gradient assumes a first order approximation. The deviations from the linear trend occur because the estimation of temperature gradient is along the normal direction and not along vertical or horizontal directions. As a result, the change in the grid cell size is not equivalent to the change in distance in the estimation of the temperature gradient. The non-linear dependence is stronger as the interface inclination gets closer to 45°. These results show that the proposed method takes more advantage of a reduction of the cell size relative to other approaches with gradients along Cartesian directions.

Another important result in Fig. 9 is that small grid cell sizes lead to negative average relative errors. The negative value indicates that in some cells the numerical temperature gradient is greater than the theoretical temperature gradient. The relative error becomes negative when the distance between the G-cell and the interface d_3 is shorter than 0.25 μm. This occurs because the temperature gradient increases as the length d_3 becomes smaller. In fact, the limit of the temperature gradient as d_3 approaches zero is infinite. However, with the proposed method, the shortest distance is $\Delta s/2$ since the method locates the G-cell in the neighboring cells (cells around the mixture cell).

Table 3 shows the results obtained for a bubble radius of 1 mm with grid cell sizes of 10, 6, and 2 μm. The small relative errors show that the proposed method works with bigger bubbles and with larger computational cells. The maximum relative error is

Table 3
Accuracy of the proposed method to estimate mass transfer with various grid cell sizes. Bubble radius of 1 mm.

Grid cell size (μm)	Min. % rel. error	Max. % rel. error	Aver. % rel. error
10	0.75	6.33	3.27
6	0.21	2.45	1.13
2	−0.33	0.11	−0.12

small and the magnitude decreases with smaller grid cell sizes. The results indicate that bubble radii of 0.1 mm and 1 mm have equivalent relative errors. This occurs because in spherical bubble growth, the thickness of the thermal film is proportional to the size of the bubble. For example, the thickness of the thermal film is 12 μm with a bubble radius $R = 0.1$ mm, whereas the thickness of the thermal film is 120 μm with a bubble radius $R = 1$ mm.

4.1.2.2. Estimation of mixture cell temperature along a spherical interface. This section evaluates the accuracy of OCASIMAT to estimate the temperature of the mixture cells. The proposed method uses a mathematical expression to determine the temperature of the mixture cells. The derivation of this expression assumes a linear temperature profile near the interface. The linear temperature profile is a valid assumption as long as the points used to estimate the temperature gradient remain close to the interface. Under this assumption, the temperature gradients obtained with two different points are equivalent if these points are located along a direction perpendicular to the interface.

To analyze the accuracy of the proposed approach on determining the temperature of the mixture cell, we considered a spherical bubble of radius $R_0 = 0.1$ mm with the initial temperature profile. The numerical temperature of the mixture cells was determined with Eq. (14). To compare the numerical temperatures, Scriven's solution Eq. (22) determined the theoretical mixture cell temperatures. Scriven's solution is valid only for the liquid side, but Eq. (22) has a continuous temperature variation through the interface. Therefore, the theoretical solution provides mixture temperatures that preserve a uniform temperature gradient through the interface. The percentage of relative error is given by $(T_{M,th} - T_{M,num})/T_{M,th} \times 100$.

Fig. 10(a) shows the relative error of the temperature on each mixture cell along the interface with a grid cell size is 0.6 μm. The results show a relative error in the range −0.1 to 0.17%. In addition, the relative error changes from positive to negative values as the interface moves through the mixture cell. For example, in the range 0 to 2.5° the relative error decreases from 0.05 to 0%, and in the range 2.5 to 5°, the relative error becomes negative. At 5° the error varies from −0.1 to 0.12%, which is due to a change in the radial location of the G-cell; this jump increases the distance between the G-cell and the interface d_3. These results show that the relative error decreases as the interface gets closer to the mixture cell center, and the relative error is zero when the interface is in contact with the mixture cell center.

Fig. 10(b) shows the temperature of the mixture cells along the interface. The results indicate that the temperature of the mixture cells oscillates around the saturation temperature $T_{sat} = 373.15$ K. The magnitude of the mixture cell temperature depends on the position of the interface: $T_M < T_{sat}$ when the interface lies above the mixture cell center, and $T_M > T_{sat}$ when the interface lies below the mixture cell center. These results show that the estimated value for the mixture cell temperature preserves the temperature profile near the interface, which helps to compute accurate temperature gradients at the faces of the mixture cells.

Table 4 shows the relative errors of the estimated mixture cells temperatures for grid cell sizes of 1, 0.6, and 0.2 μm. Results show maximum relative errors of 0.25%; this error can be reduced to 0.07% by augmenting the grid cell resolution. The average relative error is smaller than 0.08%, which shows that the proposed method accurately determines the temperature of the mixture cells. The dependence between the grid cell size and the average relative error follows a linear trend. This trend occurs because the method performs a linear interpolation along the direction normal to the interface to find the temperature of the mixture cell.

4.2. Application of OCASIMAT to phase change problems

This section applies the proposed methods for mass transfer and interface cell temperature to problems of phase change with a planar interface and with a spherical bubble. The purpose is to probe the application of the proposed methods to solve problems with moving interfaces. In addition, the comparison with theoretical solutions for interface displacement validates the numerical methods adopted that conduct the simulation.

In the simulation, the continuity and momentum equation were solved with the SIMPLE (Semi-Implicit Method for Pressure Linked Equations) algorithm. The PRESTO (PREssure STaggering Option) scheme interpolated the pressure values at the faces in the momentum equation. The second order upwind scheme discretized the convective terms in the momentum equation. The Power-Law scheme discretized the convective terms in the energy equation. For the diffusive terms in the momentum and energy equation, a central difference second order discretization scheme was adopted. The VOF equation was solved with an explicit scheme with a sharp-type interface modeling without interfacial anti-diffusion. The convergence criteria for the continuity, momentum, and energy equation was 1×10^{-8}. For more details on the discretization schemes, the reader is directed to ANSYS-Fluent documentation.

The fluid properties correspond to water at 1 atm. The vapor remained at a saturated state and the liquid was superheated. The operating conditions had a superheat level $T_B - T_{sat} = 5$ K.

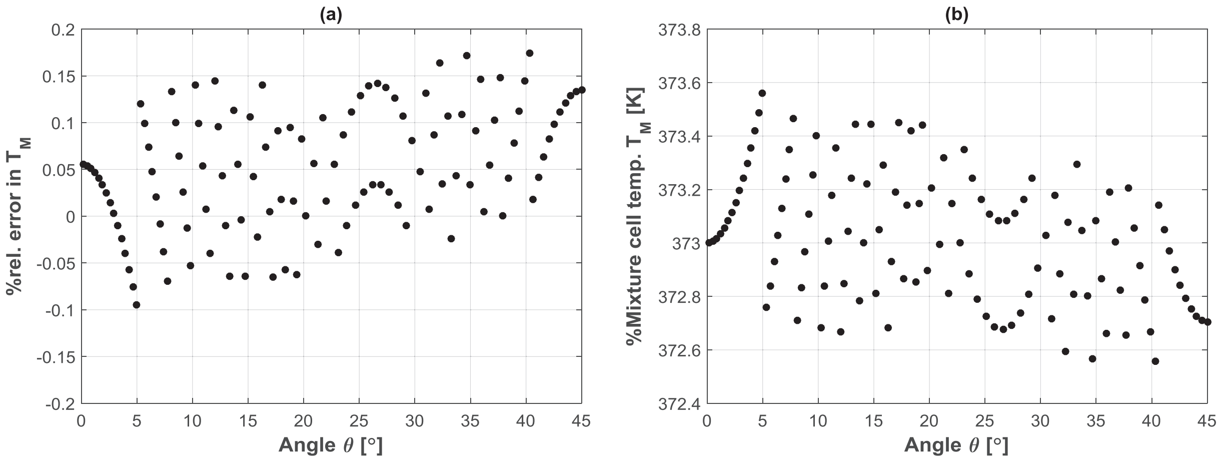

Fig. 10. Accuracy on the estimation of the mixture cell temperature along the interface. (a) Percentage of relative error (b) magnitude of mixture cell temperature.

Table 4
Accuracy of the proposed method to estimate mixture cell temperature with various grid cell sizes. Bubble radius of 0.1 mm.

Grid cell size (μm)	Min. % rel. error	Max. % rel. error	Aver. % rel. error
1	−0.14	0.25	0.077
0.6	−0.09	0.17	0.051
0.2	−0.01	0.07	0.028

4.2.1. Planar interface evaporation

Planar interface evaporation is a relatively simple problem since the interface has a constant orientation. Still, the mass transfer requires the distance between the interface and the neighboring cell center to estimate the interfacial temperature gradient. In addition, the simulation requires a fixed temperature on the mixture cell to ensure a correct numerical prediction of the temperature of the neighboring cell.

In the simulation, the initial interface position was $X_0 = 2.1$ mm that corresponds to a time of $t_0 = 0.1$ ms. The initialization at t_0 creates a thick thermal film, which can be captured with large grid cell sizes. The thickness of the thermal film at t_0 is $\delta_{th} = 0.8$ mm. Previous numerical works adopted a similar initialization [2,15,43]. Eq. (18) determined the initial temperature $T(x, 0)$. The computational domain was one cell thick, which facilitates the access to the mixture and neighboring cells. Grid cell sizes of $\Delta x =$ 80, 40 and 10 μm were adopted. The time step was 5×10^{-6} s, with a Courant number smaller than 0.1.

Fig. 11 shows the change in the interface position as a function of time. Eq. (16) gives the theoretical interface displacement. The results in Fig. 11 indicate that a reduction in the grid cell size improves the accuracy of the simulation. At 1 s, the relative error in interface displacement is 10, 3.5, and 0.04% with 80, 40 and 10 μm, respectively. The good agreement between the theory and the simulation validates the proposed numerical practices that compute the mass flux and the mixture cell temperature on the evaporation of a planar interface.

Fig. 12 shows the temperature distribution on the vapor and liquid phases at 0.2 s in the evaporation of a planar interface. At 0.2 s the interface location is X = 3.1 mm and the thickness of the thermal film is δ_{th} = 1.12 mm. The results in Fig. 12 indicate that the temperature profile gets closer to the theoretical data as the grid cell size Δs is reduced. At 0.2 s the number of cells that capture the thermal film is 14 with a cell size of 80 μm and 112 with a cell size of 10 μm. The results indicate that the increase in the grid resolution improves the accuracy of temperature gradient at the interface, which is translated into more precise interface displacements. In addition, the results show evidence of a limitation on the number of computational cell needed in the thermal film to generate accurate mass transfers. For the analyzed case at the initial time, 10 cells in the thermal film are insufficient to provide accurate interfacial temperature gradients, and 80 computational cells provide a good resolution.

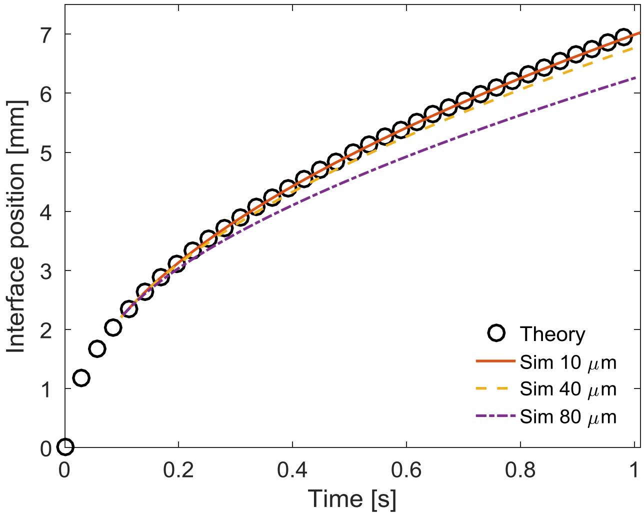

Fig. 11. Comparison simulation versus theory of the interface displacement in evaporation with a planar interface.

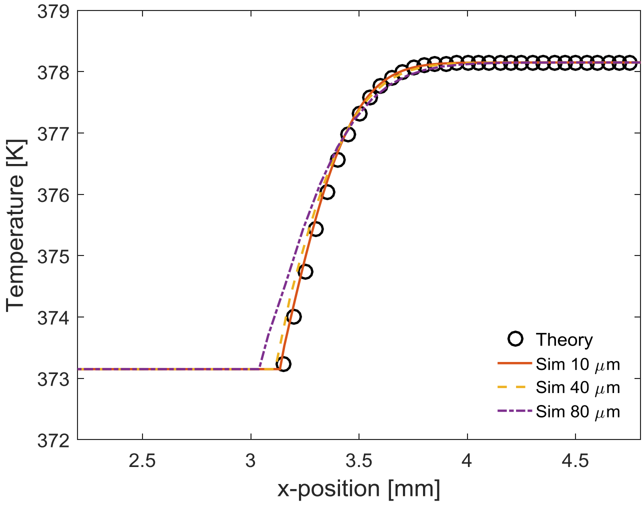

Fig. 12. Accuracy of the numerical simulation on the prediction of the temperature field in the vapor and liquid phases at 0.2 s. Planar interface evaporation.

4.2.2. Spherical bubble growth

Spherical bubble growth is a more complex simulation that involves geometrical calculations to account for the interface effects. The analysis considers adiabatic and non-adiabatic spherical bubble growth. Adiabatic bubble growth has a constant mass flux at the interface. A comparison with theory ascertains a proper solution of the momentum and volume-of-fluid equations. Non-adiabatic bubble growth has a mass flux that depends on the temperature gradient at the interface. A comparison with theory ascertains a proper solution of the energy equation, and the proper estimation of the mass transfer based on the interface position and orientation.

The simulations with a spherical bubble consider an initial bubble radius R_0 = 0.1 mm. The initial bubble patch had the reconstructed interface, meaning that the initial volume-fractions of the mixture cells correspond to the actual interface location and orientation. The computational domain was a square of length L = 0.5 mm. The boundary conditions were axisymmetric at the bottom boundary, symmetric at the vertical boundary in contact with the bubble, and pressure-outlet at the other boundaries. Only one quarter of the domain is simulated with axisymmetric and symmetric boundary conditions. Simulations of spherical bubble growth had a Courant number of less than 0.1, which led to a time step of 1×10^{-6} s for grid cells sizes of 1 μm. Simulation with a grid cell size of 0.2 μm had a time step of 1×10^{-7} s.

4.2.2.1. Adiabatic bubble growth. The simulation of adiabatic bubble growth had an initial bubble of radius 0.1 mm. The simulation imposed a prescribed mass transfer in the mixture cells. At each time step, UDFs identify the mixture cells and determine the mass source term in Eq. (1) based on the mass flux. In the mass source term, the interface inclination and the volume-fraction give the interface surface area. It is important to emphasize that in the present work, the interface was not smeared and the mass source terms were not distributed into two to three cells as previously done in several works. The distribution of the mass source terms is usually done to preserve the stability of the simulation. The present simulation declared mass source terms only on the cells that had an interface, and the mass transfer assumed a constant mass flux. The imposed mass flux is the average mass flux in a non-adiabatic bubble in a time interval. The time interval corresponds to 1 ms with an initial bubble of radius 0.1 mm. The magnitude of the imposed mass flux is $m'' = 0.18$ kg/s m^2. The simulation considered two different grids cell sizes of 10 and 1 μm.

Fig. 13 compares the numerical and the theoretical bubble radius. A comparison between numerical and theoretical bubble radius points out issues in mass conservation and interface shape during bubble growth. The results show excellent agreement between the numerical and the theoretical bubble radius as a function of time. In addition, both grid cell sizes provide a similar level of mass conservation. The good agreement is because by nature the VOF method is mass conservative since it moves the interface with changes in volume. Moreover, in this problem the generation of vapor drives the movement of the interface. The advection effect dictated by the vapor velocity in the VOF equation is negligible since the vapor remains static during the phase change process. The good agreement between theory and simulation shows evidence of an accurate interface tracking simulation when there is mass transfer between two phases divided by a sharp interface.

Fig. 14 shows the liquid volume-fractions F_l at 0.5 ms for grid cell sizes of 10 and 1 μm. The numbers around the contour give the coordinates in millimeters. The vapor and liquid phases have F_l equal to 0 and 1, respectively. The mixture cells (cells with an interface) have liquid volume-fractions in the range of 0 to 1. The initial bubble radius is 0.1 mm and at 0.5 ms the bubble radius is 0.25 mm. The contours of volume-fraction indicate that the bubble maintains its shape during the growth process. In addition, the results in Fig. 14(a) show that interface lies within one cell since only one cell has a volume-fraction between 0 and 1 along the interface. A comparison between the contours in Fig. 14 (a) and (b) shows that a decrease in the grid cell size improves the resolution around the interface; a grid cell size of 1 μm creates an almost unperceived division between the liquid and the vapor phases. These results indicate that a peak in the mass source terms at the mixture cells provides a stable bubble growth simulation; this is true for any grid cell size.

Fig. 15 shows the velocity vectors on both phases at a simulation time of 0.5 ms for a bubble that grows with a constant mass flux. The data generated by the numerical solver was exported to a graphics software to provide a better visualization of the flow behavior. The numbers around the contour give coordinates in millimeters. The vapor velocity is small because the bubble remains static during phase change. The liquid velocity is highest at the interface and it decreases as the fluid gets far from the interface due to the increase in the fluid area. At the interface, the liquid velocity is high due to the expansion of the vapor during phase change. For a certain volume of liquid consumed V_l, the equivalent volume of vapor generated V_v is almost 1600 times higher ($\rho_l V_l = \rho_v V_v$). Therefore, during the phase change process, the vapor expansion increases the momentum of the surrounding liquid. In addition, it is worth to notice the influence of the grid cell size on the fluid velocity around the interface. The vapor velocity near the interface is high with a grid cell size of 10 μm, and relatively small with a grid cell size of 1 μm. This behavior is due to assumed average properties on the mixture cells in the solution of the momentum equation. The results show that this approximation generates unphysical velocities on the neighboring cells on the vapor, which become more notorious with larger grid cell sizes. However, as the grid cell size is refined, the effect of the average properties on the mixture cells is less prominent, and the simulation provides a better representation of jump in the liquid velocity at the interface.

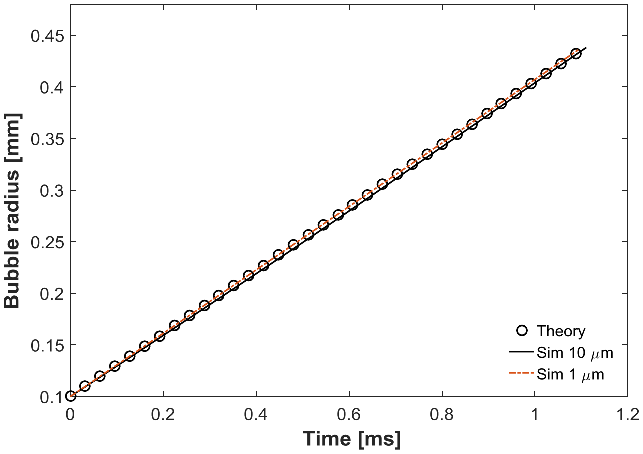

Fig. 13. Comparison theory versus simulation of adiabatic bubble growth with a constant mass flux.

Previous works have reported the presence of parasitic velocities that arise mainly due to errors in the estimation of the surface tension. Parasitic velocities create convective currents that might affect the momentum and energy transport near the interface. To determine the magnitude of the parasitic velocities in the simulation, the velocity field near the interface was analyzed. The study identified the mixture and G-cells after the first time-step. The velocity magnitude of the G-cells was retrieved and compared with the theoretical liquid velocities.

Fig. 16 compares the magnitude of the numerical and theoretical velocities on both phases. Results in Fig. 16(a) show the liquid velocity along the θ direction from $0^\circ \leqslant \theta \leqslant 90^\circ$ near $r = 0.1$ mm (on the G-cells). Results in Fig. 16(a) show variations between the numerical and theoretical velocities, but at $\theta = 0^\circ$ there is good agreement. The results in Fig. 16(b) show the fluid velocity along the radial direction from $0 \leqslant r \leqslant 0.5$ mm at $\theta = 0^\circ$. Despite the differences observed in Fig. 16(a), the results in Fig. 16(b) show excellent agreement for the liquid velocity because the plot considers fluid velocities at $\theta = 0^\circ$. The results in Fig. 16(a) show appreciable variations between the numerical and theoretical velocities along θ direction. However, the numerical and theoretical velocities have the same order of magnitude. The maximum relative error is 16 %, and the average relative error is 0.75%. The deviations of liquid velocities oscillate around the theoretical values, which contributes to reduce the magnitude of the average relative error. Maximum deviations occur around 15° and 75°; around these angles, we observed the presences of a large number of mixture cells with small interfaces. Small interfaces are far from the mixture cell center. Therefore, the source of the deviation of the liquid velocities could be related to the estimation of the normal vector at the mixture cell center rather than at the interface. The errors in the estimation of the normal vector affect the surface tension effects at the interface, which creates deviations in the fluid velocities around the interface. The results in Fig. 15(b) show that the simulation accurately predicts the variation of the velocity along the radial direction. However, the simulation predicts a continuous change in the vapor velocity near the interface. The continuous change occurs because the mixture cells have average properties. To capture the jump in velocity at the interface, the simulation should consider ghost velocities at the mixture cells as done by Tanguy et al. [48]. However, it is important to mention that the

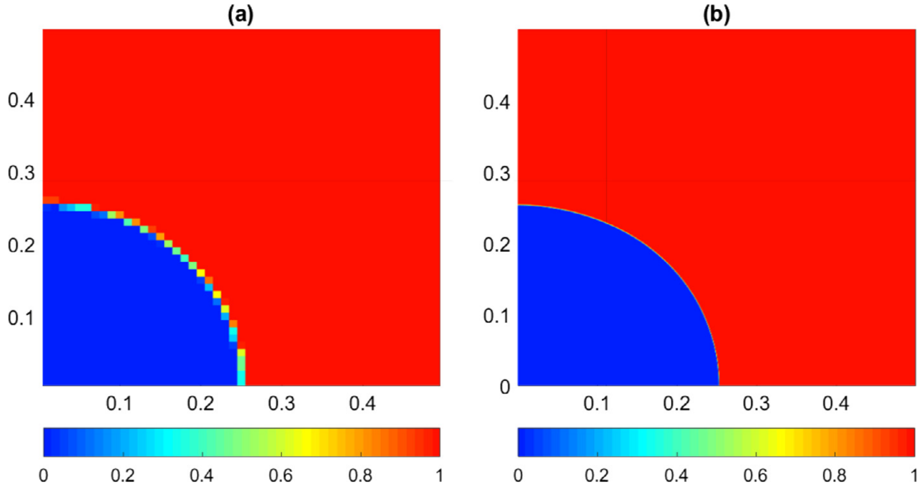

Fig. 14. Liquid volume-fractions at 0.5 ms in axisymmetric bubble growth. Two different grid cell sizes: (a) Δs = 10 μm, (b) Δs = 1 μm.

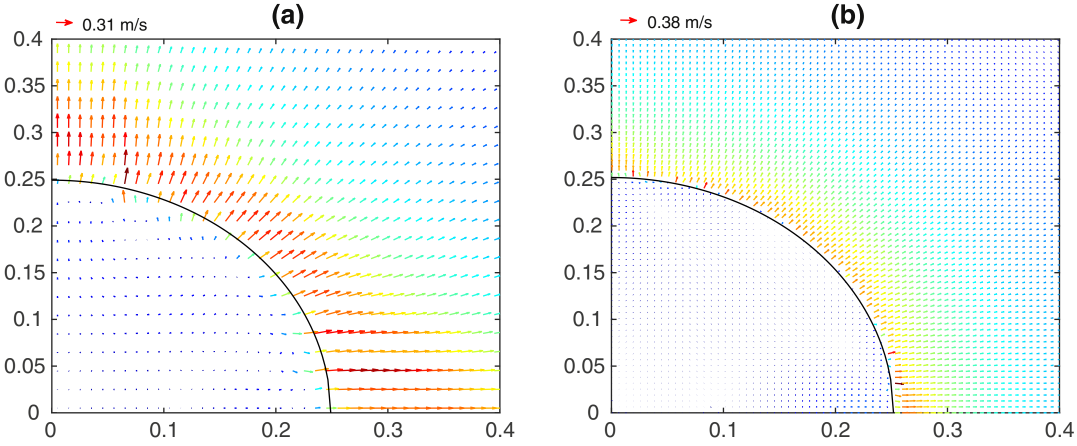

Fig. 15. Fluid velocity vectors in adiabatic bubble growth with two different grid cell sizes: (a) Δs = 10 μm, (b) Δs = 1 μm. Time 0.5 ms.

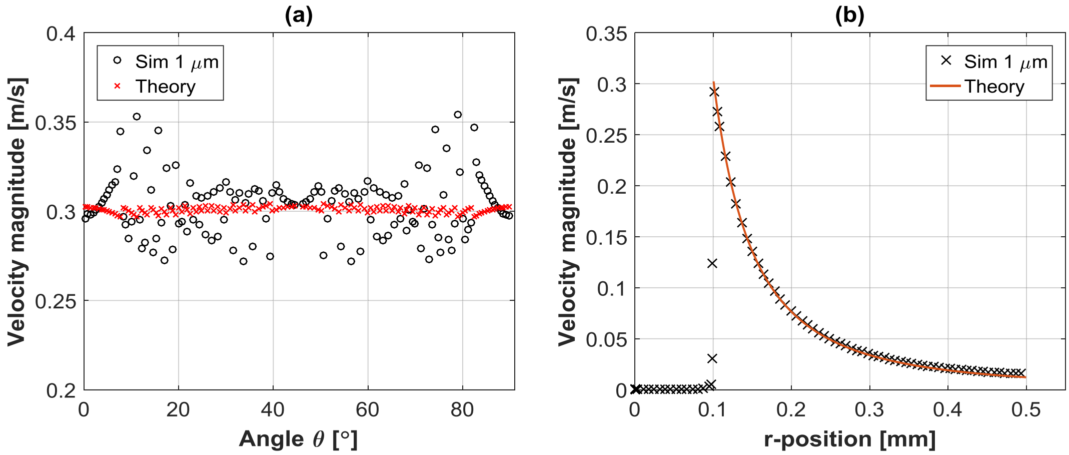

Fig. 16. Comparison of the numerical and theoretical velocities after one time step. (a) Near the interface on the G-cells, and (b) on the cells along the radial direction. Grid cell size Δs = 1 μm, results reported after first time step.

continuous jump has a minimal effect in the bubble growth rate since the mass transfer model is independent of the velocities near the interface. Therefore, the use of average properties at the mixture cells in the solution of the momentum equation might still provide a reasonable approximation of the bubble growth process.

4.2.2.2. Non-adiabatic bubble growth. In non-adiabatic bubble growth, parasitic velocities near the interface might affect the temperature distribution near the interface. Another challenge is the mass transfer computation, which depends on the temperature gradient at the interface. In addition, accurate temperature distributions near the interface require proper temperatures on the mixture cells.

The simulation of a non-adiabatic bubble was performed for a time of 0.1 ms. The initial bubble radius was R_0 = 0.1 mm. The wall superheat was 5 K and the properties correspond to water at 1 atm. Similar times of simulation and initial conditions were adopted in previous numerical works on bubble growth [2,15,24]. Scriven's solution Eq. (22) provide the initial temperature field. The objective is to evaluate if the developed methods correctly simulate the phase change process with a moving interface. These methods include: (i) estimation of the mass transfer, (ii) estimation of the interface cell temperatures, and (iii) numerical solver customization.

A comparison between the numerical and theoretical bubble radius evaluates the accuracy the interfacial heat and mass transfer. The numerical bubble radius assumes a spherical bubble. Scriven's solution Eq. (21) gives the theoretical bubble radius. Fig. 17 compares the computed and the theoretical bubble radius. The computational domain had grid cell sizes of 1, 0.6 and 0.2 μm. The simulations ran in parallel processing and the grid was refined near the interface to reduce the cost of the simulation.

The results in Fig. 17 indicate that the accuracy of the simulation improves with smaller grid cell sizes. The relative error in bubble radius at the end of the simulation is 6.2, 3.0, and 0.2% with grid cell sizes of 1, 0.6, and 0.2 μm, respectively. The improvement is due to the higher grid resolution in the thermal film obtained with smaller grid cells. The number of cells in the thermal film at the beginning of the simulation is 12, 20, and 80 for grid cell sizes of 1, 0.6, and 0.2 μm, respectively. Interestingly, similar to the simulation of planar interface evaporation, the simulation of a spherical bubble required at least 80 cells to capture the thermal film near the interface. Another important observation is that large grid cell sizes leads to smaller bubble growth rates. These results indicates that large grid cells size lead to a faster growth of the thermal film relative to the theory. The faster growth creates smaller mass fluxes because the magnitude of the interfacial temperature gradients decreases.

The variation of the temperature gradients along the interface (see Fig. 8) could create a non-uniform mass transfer. Moreover, the interface was not smeared, which account for the abrupt change in the thermal properties at the interface. These factors might introduce interface deformations or numerical instabilities as pointed out by previous works. To observe possible deformations of the interface, the bubble shape was analyzed at different times. Fig. 18 shows the liquid volume fraction at times of 0.11 and 0.16 ms for a grid cell size of 0.6 μm. The results distinguish interfaces without appreciable deformations. Moreover, the bubble conserved its shape throughout the simulation. In addition, the numerical solution was stable since the residuals for the governing equations reached the convergence criteria. These results indicate that the variations in the estimation of the temperature gradient generated by OCASIMAT are insignificant, and therefore they have a null influence on the shape of the interface. In addition, the results indicate that fixing the temperature of the mixture cells (to account for the sharp interface) leads to a stable simulation.

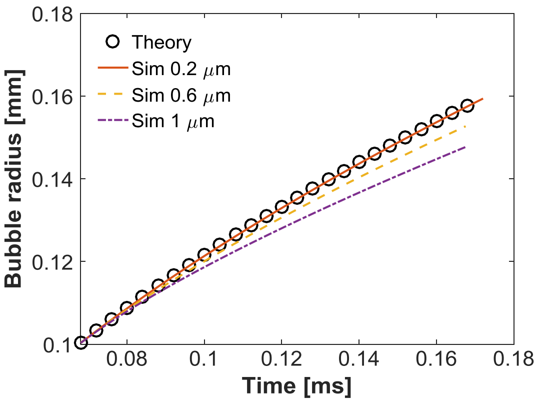

Fig. 17. Prediction of the bubble growth rate. Non-adiabatic bubble growth with temperature gradients at the interface.

Fig. 19 shows the velocity distribution on both phases at two different instances. Fig. 19(a) shows the velocity distribution at 0.11 ms, and Fig. 19(b) shows the velocity distribution at 0.16 ms. A comparison between the two figures shows that the liquid velocity decreases with time. This occurs because as the bubble grows, the mass flux becomes smaller due to the increase of the thermal film thickness. The results also indicate maximum liquid velocities of 0.8 m/s and 0.68 m/s at 0.11 ms and at 0.16 ms, respectively. These magnitudes are higher than the theoretical values of 0.56 m/s and 0.46 m/s at 0.11 ms and at 0.16 ms, respectively. As shown in Fig. 16(a), the liquid velocity distribution along the interface follows an oscillatory trend, and therefore higher numerical velocities relative to theoretical data is an expected trend. Nevertheless, the results indicate that simulation predicts velocities that are in the same order of magnitude as in the theory, and that the oscillations in the velocity magnitude occurs only on certain regions along the interface.

Deviations in the velocity of the liquid might affect the temperature distribution due to the convective transport. However, conduction also governs the heat transfer distribution near the interface. In the present case of analysis, the convective transport plays an important role since Peclet number $\left(Pe = \rho_l v_l / \left(\frac{k_l}{c_l}\frac{1}{\Delta s}\right)\right)$ changes from 2.6 to 1.7. This implies that initially, when the bubble is small, convection governs the heat transfer because the mass flux is high. As the bubble continues to grow, the convective effect becomes less significant and conduction begins to control the heat transfer near the interface. Fig. 20 shows the temperature distribution on both phases at two different times. The vapor phase remains at a saturation temperature of 373.15 K, whereas the liquid phase has a temperature that changes from 373.15 K to 378.15 K. In the contours, the mixture cells have a saturation temperature instead of a fixed temperature for clarity purposes. As mentioned earlier, the thermal film forms due to the thermal interaction between the interface and the liquid. The results show that the temperature distribution is uniform along the interface, which proves a correct fixing of the mixture cells to account for a sharp interface. Another important fact is that the thermal film remains thin throughout the bubble growth process. The thickness of the thermal film changes from 12 μm to 20 μm in the simulation. The thermal film is not consumed during bubble growth, but it moves along with the interface due to a convective transport. At the same time, the thickness of the thermal film increases with time due to the diffusive transport. Results in Fig. 20(b) indicate that as the bubble grows, slight deformations of the thermal film appear near the boundaries. The reason of these variances is an increase in the liquid velocity on these regions. Fig. 16(a), which plots the velocity distribution near the interface along θ direction, shows clear evidence of such increase in the liquid velocity on these regions. As observed in Fig. 16(a), the changes in the liquid velocity occur in adiabatic bubble growth; therefore, the errors are unrelated to the methods proposed to estimate the mass transfer or the temperature of the mixture cell. Such a trend could be due to secondary effects of errors in normal vectors around 15° and 75°. Different to other regions, the velocity near the boundaries does not oscillates around the theoretical velocity. Therefore, these variations in velocity enhance the convective heat transfer, which causes the deformation of the thermal film. However, results indicate that

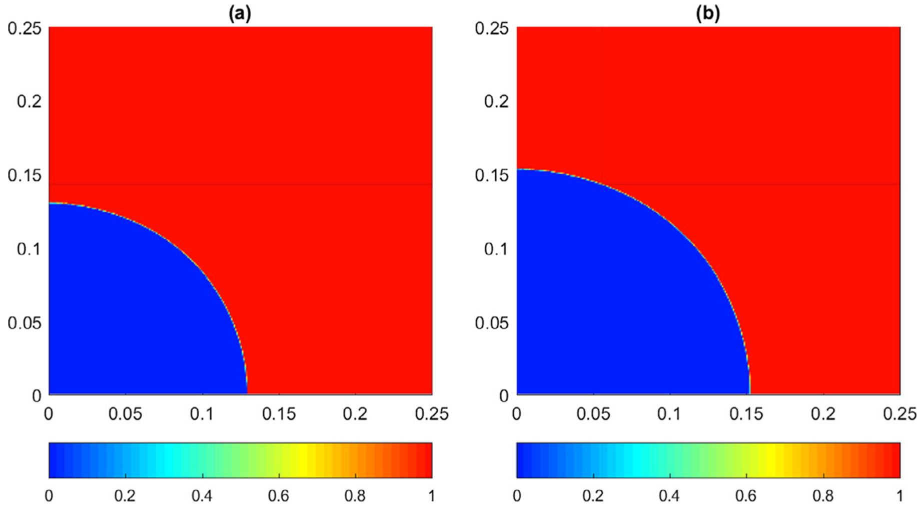

Fig. 18. Field of liquid volume-fraction in non-adiabatic bubble growth at two simulation times. Grid cell size 0.6 μm. (a) $t = 0.11$ ms (b) $t = 0.16$ ms.

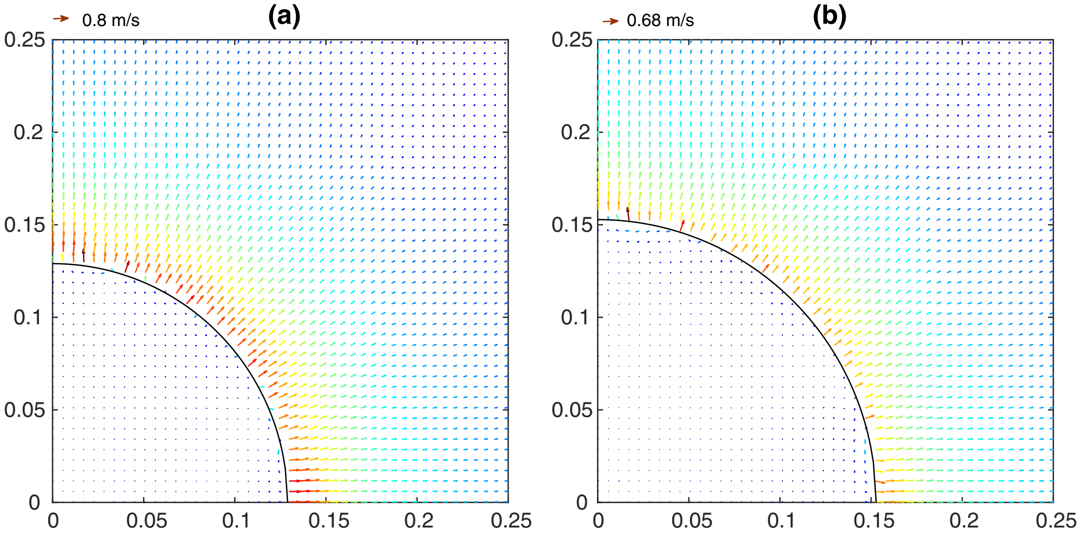

Fig. 19. Velocity vectors in non-adiabatic bubble growth. Axis dimension is meters. Grid cell size 0.6 μm. (a) $t = 0.11$ ms (b) $t = 0.16$ ms. Circular line gives interface location.

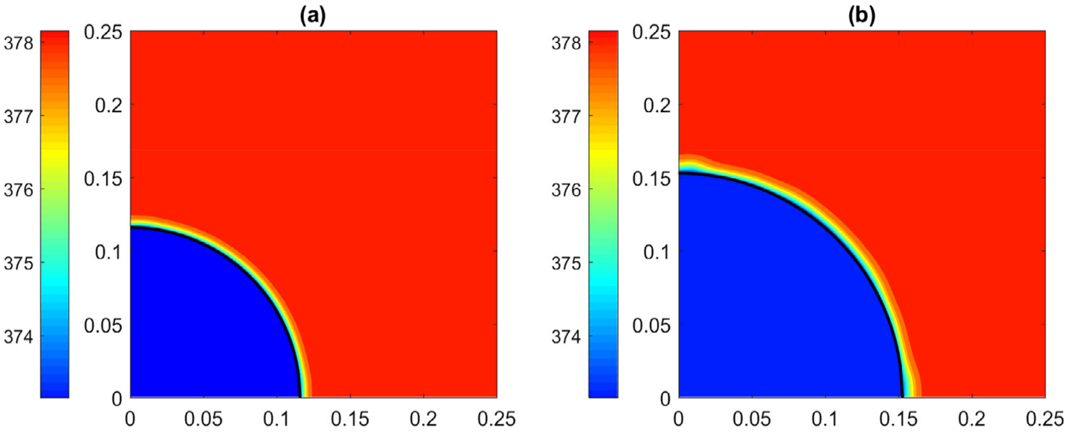

Fig. 20. Temperature distribution in non-adiabatic bubble growth at two simulation times. Grid cells size Δs = 0.6 μm. (a) $t = 0.11$ ms (b) $t = 0.16$ ms. Circular line gives an approximated interface location.

this deformation of the thermal film has a minimal influence in the mass transfer and bubble growth rate.

5. Conclusions

The present work described a method to simulate the heat and mass transfer on interfaces with phase change. The proposed one-cell algorithm for sharp interface and mass transfer (OCASIMAT) finds the mass transfer and the fixed temperature of the mixture cells with the temperature gradient at the interface. The advantages are better accuracy and simpler computation as compared to available methods in the literature. In addition, the work describes the implementation of the proposed method in a numerical software to simulate vapor–liquid problems with phase change.

OCASIMAT injects three probes at the interface. The length and the temperature at the extremes of the three probes define temperature gradient at the interface. This method allows the estimation of mass transfer with the temperature of the closest cell to the interface along the normal direction. In addition, the method allows the estimation of the mixture cell temperature by a mathematical expression. This expression relates the temperature gradients at the interface by assuming a linear temperature profile near the interface. The method requires the control of fewer cells around the interface and avoids the use of mixture cells in the mass transfer, which reduces the complexity of modeling the evaporative interface while improving accuracy.

Results indicate that the proposed method for mass transfer is more accurate relative to other approaches in the technical literature. For three different scenarios, the method for mass transfer had a maximum relative error of 4.17%, whereas previous common methods had maximum relative errors of the order of 5 to 20%. The estimation of mass transfer along the interface of a spherical bubble had a maximum average relative error of 3.5% for grids cell sizes in the range of 10 to 0.2 μm. For the same range of grid cell sizes, the estimation of the temperature of the mixture cell had maximum average relative errors of 0.077%. These results indicate that the accuracy proposed approach is superior to other methods, which makes it attractive to perform simulations of bubble growth.

OCASIMAT was applied on the simulation of planar interface evaporation and spherical bubble growth. The simplicity of the proposed approach allowed the simulation of these problems in a numerical software. To our knowledge, this is the first report that adopts a numerical software to predict accurate temperature distributions near the interface in spherical bubble growth. Results indicated that the proposed approach leads to accurate interface displacements since the mass flux is determined with the interfacial temperature gradient. The method to determine the temperature of the mixture cell allowed an accurate temperature distribution near the interface. In addition, the method provided a better representation of the heat and mass transfer at the interface by generating a stable simulation with a sharp interface and with mass transfer only on the mixture cells.

Funding

The work was performed in the Thermal Analysis, Microfluidics, and Fuel Cell Laboratory in the Mechanical Engineering Department at the Rochester Institute of Technology, NY. This work was supported by the National Science Foundation [grant number 1511314].

Conflict of interest

The authors declared that there is no conflict of interest.

References

[1] S. Hardt, F. Wondra, Evaporation model for interfacial flows based on a continuum-field representation of the source terms, J. Comput. Phys. 227 (11) (2008) 5871–5895.

[2] C. Kunkelmann, P. Stephan, CFD simulation of boiling flows using the volume-of-fluid method within OpenFOAM, Numer. Heat Transf. Part Appl. 56 (8) (2009) 631–646.

[3] M. Magnini, B. Pulvirenti, J.R. Thome, Numerical investigation of hydrodynamics and heat transfer of elongated bubbles during flow boiling in a microchannel, Int. J. Heat Mass Transf. 59 (2013) 451–471.

[4] Y. Yang, L. Pan, J. Xu, Effects of microgravity on marangoni convection and growth characteristic of a single bubble, Acta Astronaut. 100 (2014) 129–139.

[5] H.W. Jia, P. Zhang, X. Fu, S.C. Jiang, A numerical investigation of nucleate boiling at a constant surface temperature, Appl. Therm. Eng. 88 (2015) 248–257.

[6] M. Yazdani, T. Radcliff, M. Soteriou, A.A. Alahyari, A high-fidelity approach towards simulation of pool boiling, Phys. Fluids 28 (1) (2016) 012111.

[7] Q. Liu, B. Palm, Numerical study of bubbles rising and merging during convective boiling in micro-channels, Appl. Therm. Eng. 99 (2016) 1141–1151.

[8] Q. Liu, W. Wang, B. Palm, A numerical study of the transition from slug to annular flow in micro-channel convective boiling, Appl. Therm. Eng. 112 (2017) 73–81.

[9] Q. Liu, W. Wang, B. Palm, Numerical study of the interactions and merge of multiple bubbles during convective boiling in micro channels, Int. Commun. Heat Mass Transf. 80 (2017) 10–17.

[10] D. Juric, G. Tryggvason, Computations of boiling flows, Int. J. Multiph. Flow 24 (3) (1998) 387–410.

[11] G. Strotos, M. Gavaises, A. Theodorakakos, G. Bergeles, Numerical investigation on the evaporation of droplets depositing on heated surfaces at low weber numbers, Int. J. Heat Mass Transf. 51 (7–8) (2008) 1516–1529.

[12] K.L. Maki, S. Kumar, Fast evaporation of spreading droplets of colloidal suspensions, Langmuir 27 (18) (2011) 11347–11363.

[13] J.B. Haelssig, A.Y. Tremblay, J. Thibault, S.G. Etemad, Direct numerical simulation of interphase heat and mass transfer in multicomponent vapour–liquid flows, Int. J. Heat Mass Transf. 53 (19–20) (2010) 3947–3960.

[14] D. Sun, J. Xu, Q. Chen, Modeling of the evaporation and condensation phase-change problems with fluent, Numer. Heat Transf. Part B-Fundam. 66 (4) (2014) 326–342.

[15] Y. Sato, B. Ničeno, A sharp-interface phase change model for a mass-conservative interface tracking method, J. Comput. Phys. 249 (2013) 127–161.

[16] K. Ling, Z.-Y. Li, W.-Q. Tao, A direct numerical simulation for nucleate boiling by the VOSET method, Numer. Heat Transf. Part -Appl. 65 (10) (2014) 949–971.

[17] M.W. Akhtar, S.J. Kleis, N. Prabhu, N. Kobasko, S.W. Dean, A volume of fluid phase change model on adaptive octree grids, J. ASTM Int. 8 (3) (2011) 103348.

[18] M.W. Akhtar, S.J. Kleis, Boiling flow simulations on adaptive octree grids, Int. J. Multiph. Flow 53 (2013) 88–99.

[19] C. Kunkelmann, P. Stephan, Numerical simulation of the transient heat transfer during nucleate boiling of refrigerant HFE-7100, Int. J. Refrig.-Rev. Int. Froid 33 (7) (2010) 1221–1228.

[20] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part I: numerical method, Int. J. Heat Mass Transf. 47 (25) (2004) 5451–5461.

[21] J. Schlottke, B. Weigand, Direct numerical simulation of evaporating droplets, J. Comput. Phys. 227 (10) (2008) 5215–5237.

[22] J. Brackbill, D. Kothe, C. Zemach, A continuum method for modeling surface-tension, J. Comput. Phys. 100 (2) (1992) 335–354.

[23] H. Udaykumar, W. Shyy, Simulation of interfacial instabilities during solidification. 1. Conduction and capillarity effects, Int. J. Heat Mass Transf. 38 (11) (1995) 2057–2073.

[24] S. Shin, B. Choi, Numerical simulation of a rising bubble with phase change, Appl. Therm. Eng. 100 (2016) 256–266.

[25] S.V. Patankar, Chapter Seven-Finishing Touches, Numerical Heat Transfer and Fluid Flow, McGraw-Hill, New York, 1980.

[26] G. Son, V.K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, J. Heat Transf.-Trans. Asme 120 (1) (1998) 183–192.

[27] G. Son, V.K. Dhir, N. Ramanujapu, Dynamics and heat transfer associated with a single bubble during nucleate boiling on a horizontal surface, J. Heat Transf.-Trans. Asme 121 (3) (1999) 623–631.

[28] G.H. Son, A numerical method for bubble motion with phase change, Numer. Heat Transf. Part B-Fundam. 39 (5) (2001) 509–523.

[29] D. Li, V.K. Dhir, Numerical study of a single bubble sliding on a downward facing heated surface, J. Heat Transf.-Trans. Asme 129 (7) (2007) 877–883.

[30] G. Son, V.K. Dhir, Numerical simulation of nucleate boiling on a horizontal surface at high heat fluxes, Int. J. Heat Mass Transf. 51 (9–10) (2008) 2566–2582.

[31] G. Son, V.K. Dhir, Three-dimensional simulation of saturated film boiling on a horizontal cylinder, Int. J. Heat Mass Transf. 51 (5–6) (2008) 1156–1167.

[32] F. Gibou, R.P. Fedkiw, L.T. Cheng, M.J. Kang, A second-order-accurate symmetric discretization of the poisson equation on irregular domains, J. Comput. Phys. 176 (1) (2002) 205–227.

[33] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (1) (2004) 349–355.

[34] S. Tanguy, T. Ménard, A. Berlemont, A level set method for vaporizing two-phase flows, J. Comput. Phys. 221 (2) (2007) 837–853.

[35] I. Perez-Raya, S.G. Kandlikar, Numerical modeling of interfacial heat and mass transport phenomena during a phase change using ANSYS-Fluent, Numer. Heat Transf. Part B-Fundam. 70 (4) (2016) 322–339.

[36] I. Perez-Raya, S.G. Kandlikar, Chapter three – evaporation on a planar interface – numerical simulation and theoretical analysis of heat and mass transport processes, in: J.P.A. Ephraim, M. Sparrow John, M. Gorman, Thomas F. Irvine, James P. Hartnett (Eds.), Advances in Heat Transfer, Elsevier, 2016, pp. 125–190.

[37] J. Wu, V.K. Dhir, J. Qian, Numerical simulation of subcooled nucleate boiling by coupling level-set method with moving-mesh method, Numer. Heat Transf. Part B-Fundam. 51 (6) (2007) 535–563.

[38] T. Maric, H. Marschall, D. Bothe, lentFoam – a hybrid level set/front tracking method on unstructured meshes, Comput. Fluids 113 (2015) 20–31.

[39] H.K. Versteeg, W. Malalasekera, An Introduction to Computational Flud Dynamics, the Finite Volume Method, Pearson Education Limited, 2007.

[40] H. Thakur, K.M. Singh, P.K. Sahoo, Phase change problems using the mlpg method, Numer. Heat Transf. Part -Appl. 59 (6) (2011) 438–458.

[41] D.Z. Guo, D.L. Sun, Z.Y. Li, W.Q. Tao, Phase change heat transfer simulation for boiling bubbles arising from a vapor film by the voset method, Numer. Heat Transf. Part -Appl. 59 (11) (2011) 857–881.

[42] D.-L. Sun, J.-L. Xu, L. Wang, Development of a vapor-liquid phase change model for volume-of-fluid method in FLUENT, Int. Commun. Heat Mass Transf. 39 (8) (2012) 1101–1106.

[43] S.W.J. Welch, J. Wilson, A volume of fluid based method for fluid flows with phase change, J. Comput. Phys. 160 (2) (2000) 662–682.

[44] V. Alexiades, A.D. Solomon, Chapter two-problems with explicit solutions, Mathematical Modeling of Melting and Freezing Processes, Hemisphere Publishing Corporation, 1993.

[45] M.S. Plesset, S.A. Zwick, The growth of vapor bubbles in superheated liquids, J. Appl. Phys. 25 (4) (1954) 493–500.

[46] B. Mikic, W. Rohsenow, P. Griffith, On bubble growth rates, Int. J. Heat Mass Transf. 13 (4) (1970), 657-.

[47] L.E. Scriven, On the dynamics of phase growth, Chem. Eng. Sci. 50 (24) (1995) 3907–3917.

[48] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, C. Colin, Benchmarks and numerical methods for the simulation of boiling flows, J. Comput. Phys. 264 (2014) 1–22.