



A volume-of-fluid method for interface-resolved simulations of phase-changing two-fluid flows

Nicolò Scapin^a, Pedro Costa^{a,*,1}, Luca Brandt^{a,b}

^a Linné FLOW Centre and SeRC (Swedish e-Science Research Centre), KTH Mechanics, SE-100 44 Stockholm, Sweden

^b Department of Energy and Process Engineering, Norwegian University of Science and Technology (NTNU), Trondheim, Norway

ARTICLE INFO

Article history:

Received 1 July 2019

Received in revised form 7 January 2020

Accepted 9 January 2020

Available online 13 January 2020

Keywords:

Interface-resolved direct numerical simulations

Volume-of-fluid method

Phase change

ABSTRACT

We present a numerical method for interface-resolved simulations of evaporating two-fluid flows based on the volume-of-fluid (VoF) method. The method has been implemented in an efficient FFT-based two-fluid Navier-Stokes solver, using an algebraic VoF method for the interface representation, and extended with the transport equations of thermal energy and vaporized liquid mass for the single-component evaporating liquid in an inert gas. The conservation of vaporizing liquid and computation of the interfacial mass flux are performed with the aid of a reconstructed signed-distance field, which enables the use of well-established methods for phase change solvers based on level-set methods. The interface velocity is computed with a novel approach that ensures accurate mass conservation, by constructing a divergence-free extension of the liquid velocity field onto the entire domain. The resulting approach does not depend on the type of interface reconstruction (i.e. can be employed in both algebraic and geometrical VoF methods). We extensively verified and validated the overall method against several benchmark cases, and demonstrated its excellent mass conservation and good overall performance for simulating evaporating two-fluid flows in two and three dimensions.

© 2020 Elsevier Inc. All rights reserved.

1. Introduction

Multiphase flows undergoing phase change are found in many environmental and industrial contexts, such as cloud formation and rainfall, cooling towers, wet scrubbers and spray combustion. These systems are inherently complex, with the different phases exchanging mass, momentum and energy through an interface that moves and deforms with the flow. The thickness of this interface is typically several orders of magnitude smaller than any other relevant flow scale. From a continuum mechanics modeling perspective, the interface is often modeled as infinitesimally thin, and its physics simplified to appropriate interphase coupling conditions, derived from interfacial transport balances and thermodynamic considerations [1]. Even considering these modeling assumptions, first-principles, interface-resolved numerical simulations of these flows remain a challenge although simulations can provide valuable insights for e.g. upscale models aiming to improve predictive engineering tools [2] in a context where experiments are particularly difficult.

* Corresponding author.

E-mail addresses: nicolos@mech.kth.se (N. Scapin), p.simoescosta@gmail.com (P. Costa), luca@mech.kth.se (L. Brandt).

¹ Present address: Faculty of Industrial Engineering, Mechanical Engineering and Computer Science, University of Iceland, Hjarðarhagi 2-6, 107 Reykjavik, Iceland.

Interface-resolved simulations of two-fluid flows are often divided into two categories according to the interface representation [3]: (1) *Interface-tracking* (or *front-tracking*) methods explicitly define a mesh with Lagrangian markers, attached to and moving with the interface; (2) *interface-capturing* methods define the interface in a higher dimension by solving a transport equation for an auxiliary scalar field. Front-tracking methods [4] allow in general for a more accurate interface representation, at the cost of more complex implementations, specially when it comes to handling surface topology changes. For the same grid resolution and computational cost, interface-capturing methods are in general less accurate in the calculation of the interface properties, such as curvature and normal vector, but simpler to implement, and naturally handle topology changes thanks to the implicit interface representation. Methods that fall in this category are the level-set, [5], the volume-of-fluid [6], constrained-interpolation [7] and diffuse-interface [8] methods. Given the pros and cons of the different approaches, most of them have been used to simulate phase-changing two-fluid flows in different configurations. Also, most methods assume that one of these two dominant mechanisms drives phase change: (1) large temperatures, when phase change is triggered by a prescribed interface saturation temperature – *boiling*, and (2) species concentration gradients near the interface, with phase change induced by a prescribed non-uniform interface concentration – *evaporation*.

Front-tracking (FT) methods have been used to study boiling flows, with application to film boiling; see e.g. [9], [10] and [11]. Recent studies have extended this framework to evaporating two-fluid flows in two dimensions [12], also in presence of chemical reactions [13]. Despite the successes of FT methods for phase change, highly scalable parallel implementations are challenging and remain scarce [14]. Such a feature is crucial for simulating e.g. turbulent gas-liquid flows, which may require massive simulations with $\mathcal{O}(10^8 - 10^9)$ Eulerian grid cells [15].

In regard to interface-capturing methods, the first study of interface-resolved simulations of phase changing two-fluid flows used a level-set method, applied to film boiling [16]. Several studies have followed, aiming to incorporate interphase-coupling jump conditions with the so-called ghost-fluid method, which provides a sharp representation of the jump at the discrete level. These methods have been employed for boiling [17–19], evaporation [20] and the combination of the two [21]. Despite the proven successes of the level-set method for phase change problems, these are not mass-preserving by construction. Mass conservation properties are desirable for numerical simulations of several systems, e.g. in multiphase turbulent flows, where the flow statistics should be collected over periods of time long enough that mass loss may become significant. Recent studies have dealt with this problem of level-set methods (in the absence of phase change) by introducing mass correction steps, where lost mass is redistributed near the interface depending on e.g. the local interface curvature [22,23].

Another widely-used interface-capturing approach for multiphase flow simulations is the volume-of-fluid (VoF) method, which has the major strength of ensuring mass conservation by construction. These methods have been extended to simulations of boiling flows [24–26] (in two dimensions), and evaporation [27,28]. Mass-preserving methods for boiling flows have also been devised based on coupled level-set and VoF methods [29], or the constrained-interpolation method [30]. Most of these methods use the flow velocity to transport the liquid/gas volume fraction field and include a source term accounting for the phase change. This can however cause numerical issues, as the velocity in the presence of phase change has a jump across the interface, and its divergence is non-zero. Moreover, these approaches cannot be easily and directly adapted to VoF methods where a smooth (and often divergence-free) velocity field is necessary to transport the VoF function with preserved interface thickness. Interface smearing due to the transport of a VoF function with a non-divergence-free velocity field is often avoided by resorting to ad-hoc interface-compression schemes developed for certain classes of algebraic VoF methods [31,32]. Other approaches using VoF for evaporating flows rely on the geometrical reconstruction of the interface for computing interface mass fluxes and estimating and re-distributing divergence errors in the grid cells around the interface in order to improve mass conservation; see [27]. Finally, we should note that, in the context of numerical studies for evaporating flows, thorough verification/validation studies demonstrating the grid convergence of the methods for different benchmarks remain scarce.

Here, we present a numerical model for three-dimensional direct numerical simulations of evaporating flows using a volume-of-fluid method. The transport of vaporized liquid mass is solved with a Dirichlet boundary condition at the interface, obtained assuming thermodynamic equilibrium through the Clausius-Clapeyron relation, and solved with the aid of a reconstructed level-set field. This allows to easily compute the interfacial mass flux in a band around the interface [20]. We propose here to transport the VoF function with a smooth interface velocity consisting of two terms: a divergence-free extension of the liquid velocity field, and an irrotational term due to phase change. Accordingly, the standard directional-splitting method used for the VoF advection is extended with a volume deflation step. This results in a novel approach for transporting the VoF function that shows excellent mass conservation properties, and can be easily applied to other geometrical or algebraic VoF methods for incompressible two-fluid flows. Unlike previous approaches, this allows to use a whole-domain formulation for the momentum equations, without introducing interface-sharpening terms in the VoF transport equation. The method is implemented in an efficient, FFT-based two-fluid finite-difference Navier-Stokes solver, extended with the MTHINC (algebraic) VoF method for the interface representation. We verify the proposed method against several benchmark cases of droplet evaporation, and demonstrate the solution grid convergence. Moreover, we validate the overall numerical method against psychrometric data, and prove its ability to simulate evaporating flows in the presence of large droplet deformations near solid boundaries, in two and three dimensions.

This paper is organized as follows. The governing equations are presented within the so-called one-fluid formulation in section 2. Then section 3, describes the numerical approach used to solve the system of equations, together with the

interface representation and construction of the interface velocity. The overall method is verified and validated against several benchmarks in section 4. Finally, conclusions are drawn in section 5.

2. Governing equations

We shall consider a system with two immiscible and incompressible Newtonian fluids: a single component liquid (phase 1) and an ideal mixture of an inert gas and vaporized liquid (phase 2). The two phases are bounded by an infinitesimally small interface, through which energy, momentum and mass can be transferred. Evaporation (i.e. mass transfer due to phase change) can occur, and is driven by the partial pressure of the inert gas in phase 2.

Before introducing the governing equations, it is convenient to define a phase indicator function H distinguishing the two phases at position \mathbf{x} and time t :

$$H(\mathbf{x}, t) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_1, \\ 0 & \text{if } \mathbf{x} \in \Omega_2, \end{cases} \quad (1)$$

where Ω_1 and Ω_2 are the domains pertaining to phases 1 and 2. We can use H to define the thermophysical properties in the whole domain $\Omega = \Omega_1 \cup \Omega_2$ as follows:

$$\xi(\mathbf{x}, t) = \xi_1 H(\mathbf{x}, t) + \xi_2 (1 - H(\mathbf{x}, t)), \quad (2)$$

where ξ_i ($i = 1, 2$ for phases 1 and 2) can be the mass density ρ_i , the dynamic viscosity μ_i , the thermal conductivity k_i or the inverse of the heat capacity at constant pressure $(\rho_i c_{p,i})^{-1}$. The reasons behind the use of the harmonic mean for ρc_p instead of the arithmetic one, as for the other thermophysical properties, will be detailed in section 3.5. Hereafter, unless otherwise stated, thermophysical quantities not specifically referring to one of the phases are defined from eq. (2).

The equations governing the mass, energy and momentum transport for phase 1 and 2 are coupled through appropriate interfacial conditions [1]. Below we present the governing equations in the so-called one-fluid or whole-domain formulation, where each transport equation is defined in Ω [33,18].

Navier-Stokes equations

The interfacial mass flux \dot{m} due to phase change makes the velocity field \mathbf{u} discontinuous. This can be expressed by the following Rankine-Hugoniot condition [6]:

$$\rho_1(\mathbf{u}_1 - \mathbf{u}_\Gamma) \cdot \mathbf{n} = \rho_2(\mathbf{u}_2 - \mathbf{u}_\Gamma) \cdot \mathbf{n} = \dot{m}, \quad (3)$$

where \mathbf{n} is interface normal vector (pointing to Ω_2), $\mathbf{u}_{i=1,2}$ is the fluid velocity in each subdomain, and \mathbf{u}_Γ the interface velocity ($\Gamma = \Omega_1 \cap \Omega_2$). The continuity equation accounting for this condition reads:

$$\nabla \cdot \mathbf{u} = \dot{m} \left(\frac{1}{\rho_2} - \frac{1}{\rho_1} \right) \delta_\Gamma, \quad (4)$$

where $\delta_\Gamma \equiv \delta(\mathbf{x} - \mathbf{x}_\Gamma)$ is a three-dimensional Dirac delta function, non-zero at the interface position \mathbf{x}_Γ .

The momentum equation can be written as follows [6]:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot \left(\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right) + \rho \mathbf{g} + \sigma \kappa \delta_\Gamma \mathbf{n}, \quad (5)$$

where p is the pressure field and \mathbf{g} the gravitational acceleration; the right-most term accounts for the jump in stress due to surface tension, with σ being the surface tension coefficient and κ local interface curvature. Note that equation (5) is coupled to the equation for mass conservation, eq. (4), and therefore the velocity field must satisfy the constrain on the velocity divergence, which is directly related to the mass flux at the interface.

Vaporized liquid mass transport

The transport of vapor mass is only defined in Ω_2 , and driven by a standard convection-diffusion equation:

$$\frac{\partial Y_2^l}{\partial t} + \mathbf{u} \cdot \nabla Y_2^l = D_{lg} \nabla^2 Y_2^l, \quad (6)$$

where Y_2^l denotes the mass fraction of vapor in the ideal mixture (i.e. vaporized liquid in Ω_2) and D_{lg} is the diffusion coefficient of vapor in the gas. Note that, since we consider a single component liquid, the analogous equation for the liquid phase is trivial, i.e. $Y_1^l = 1$ in Ω_1 . The interface boundary condition for $Y_{2,\Gamma}^l$ related to the saturation vapor pressure $p_{2,\Gamma}^{l,sat}$ is as follows:

$$Y_{2,\Gamma}^l = \frac{p_{2,\Gamma}^{l,sat} M_l}{(p_t - p_{2,\Gamma}^{l,sat}) M_g + p_{2,\Gamma}^{l,sat} M_l}, \quad (7)$$

where p_t is the total pressure of the mixture, and M_l and M_g denote the molar mass of the liquid and inert gas. By assuming the thermodynamic equilibrium at the interface, it is possible to relate $p_{2,\Gamma}^{l,sat}$ to the local interface temperature T_Γ , through the Clausius-Clapeyron relation [34]:

$$p_{2,\Gamma}^{l,sat} = p_t \exp \left[-\frac{h_{lv} M_l}{R} \left(\frac{1}{T_\Gamma} - \frac{1}{T^{sat}} \right) \right], \quad (8)$$

where T^{sat} is the liquid saturation temperature at ambient pressure p_t , R is the universal molar gas constant, and h_{lv} the latent heat of phase change. Finally, since in the current work we limit ourselves to a single-component liquid and neglect the gas dissolution in the liquid phase, the mass balance across the interface results in the following condition for \dot{m} [34]:

$$\dot{m}(1 - Y_{2,\Gamma}^l) = -\rho_2 D_{lg} \nabla_\Gamma Y_2^l \cdot \mathbf{n}, \quad (9)$$

where ∇_Γ denotes the gradient at $\mathbf{x} = \mathbf{x}_\Gamma$.

Energy transport

The conservation of thermal energy can be written in the one-fluid formulation as follows:

$$\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = \nabla \cdot (k \nabla T) - \dot{m} [h_{lv} + (c_{p,1} - c_{p,2})(T^{sat} - T_\Gamma)] \delta_\Gamma, \quad (10)$$

where viscous dissipation has been neglected. Here T is the temperature field, c_p the specific heat at constant pressure and k the thermal conductivity. The last term in eq. (10) quantifies the jump in enthalpy due to phase change, mostly due to the latent heat h_{lv} , but also due to the differences in specific heat between the two phases. Note that the jump in heat flux at the interface can be easily derived by integrating eq. (10) across Γ .

Governing parameters

Eqs. (4), (5), (6), (8), (7) and (10) can be written in non-dimensional form by introducing proper scaling parameters, namely a reference velocity, length and temperature scales, u_{ref} , l_{ref} and T_{ref} , together with reference thermophysical properties, here taken as those of the gas phase 2. The non-dimensional governing parameters read:

$$\begin{aligned} \text{Re} &= \frac{\rho_2 u_{ref} l_{ref}}{\mu_2}, & \text{We} &= \frac{\rho_2 u_{ref}^2 l_{ref}}{\sigma}, & \text{Fr} &= \frac{u_{ref}^2}{l_{ref} |\mathbf{g}|}, & \text{Pr} &= \frac{\mu_2 c_{p,2}}{k_2}, \\ \text{Sc} &= \frac{\mu_2}{D_{lg} \rho_2}, & \text{Ste} &= \frac{c_{p,2} T_{ref}}{h_{lv}}, & \text{Ste}^m &= \frac{R T^{sat}}{M_l h_{lv}}, & \lambda_M &= \frac{M_l}{M_g}, \end{aligned}$$

where Re , We , Fr , Pr , Sc , Ste , Ste^m and λ_M are the Reynolds, Weber, Froude, Prandtl, Schmidt, Stefan number, a modified Stefan number and the molar mass ratio; in addition to these, one needs to consider the ratios of the different thermophysical properties, $\lambda_\xi \equiv \xi_1/\xi_2$, where ξ can be ρ , μ , c_p or k .

3. Numerical method

The governing equations are solved on a fixed regular Cartesian grid (i.e. with spacing $\Delta x = \Delta y = \Delta z$), with a marker-and-cell arrangement of velocity and pressure points, using a finite-difference method. All scalar fields are defined at the cell centers. Hereafter we describe the phase-change two-fluid solver, starting with the interface-capturing method.

3.1. Interface representation

To capture the interface, we use the MTHINC volume-of-fluid method [35]. For the sake of clarity, we start by briefly describing the original method and then explain our approach for modifying the advection scheme to account for a *smooth but non-divergence-free* interface velocity, \mathbf{u}_Γ . The construction of the interface velocity is described later in section 3.6.

We start by considering the cell-averaged volume fraction field, or volume-of-fluid function C , governed by the following equation:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma H^{ht}) = C \nabla \cdot \mathbf{u}_\Gamma, \quad (11)$$

where H^{ht} is a hyperbolic tangent function, approximating the phase indicator function H . The numerical fluxes are determined from the semi-analytical procedure described in [35], without requiring an explicit interface reconstruction. The governing equation for C , eq. (11), is integrated in time with a directional-splitting method [36,37], with an additional correction accounting for the non-zero divergence of the interface velocity. As described in [35], the following implicit equations are solved sequentially from time step n to $n + 1$:

$$\begin{aligned} C_{i,j,k}^* &= C_{i,j,k}^n - \frac{1}{\Delta x} \left(f_{i+1/2,j,k}^n - f_{i-1/2,j,k}^n \right) + \frac{\Delta t}{\Delta x} C_{i,j,k}^* \left(u_{\Gamma i+1/2,j,k}^n - u_{\Gamma i-1/2,j,k}^n \right), \\ C_{i,j,k}^{**} &= C_{i,j,k}^* - \frac{1}{\Delta y} \left(g_{i,j+1/2,k}^* - g_{i,j-1/2,k}^* \right) + \frac{\Delta t}{\Delta y} C_{i,j,k}^{**} \left(v_{\Gamma i,j+1/2,k}^n - v_{\Gamma i,j-1/2,k}^n \right), \\ C_{i,j,k}^{***} &= C_{i,j,k}^{**} - \frac{1}{\Delta z} \left(h_{i,j,k+1/2}^{**} - h_{i,j,k-1/2}^{**} \right) + \frac{\Delta t}{\Delta z} C_{i,j,k}^{***} \left(w_{\Gamma i,j,k+1/2}^n - w_{\Gamma i,j,k-1/2}^n \right), \end{aligned} \quad (12)$$

where Δt is the time step, and f^n , g^* and h^{**} are the numerical fluxes computed as in [35], and u_Γ , v_Γ and w_Γ denote the three components of the interface velocity vector. Since each of the advection steps in eq. (12) corresponds to a non-divergence-free one-dimensional velocity field [38], the original method performs a further correction to ensure that the divergence-free condition is satisfied:

$$C_{i,j,k}^{n+1} = C_{i,j,k}^{***} - \Delta t F_{i,j,k}^n, \quad (13)$$

where $F_{i,j,k}^n$ is given by:

$$\begin{aligned} F_{i,j,k}^n &= C_{i,j,k}^* \frac{u_{\Gamma i+1/2,j,k}^n - u_{\Gamma i-1/2,j,k}^n}{\Delta x} + \\ &C_{i,j,k}^{**} \frac{v_{\Gamma i,j+1/2,k}^n - v_{\Gamma i,j-1/2,k}^n}{\Delta y} + \\ &C_{i,j,k}^{***} \frac{w_{\Gamma i,j,k+1/2}^n - w_{\Gamma i,j,k-1/2}^n}{\Delta z}. \end{aligned} \quad (14)$$

The second term on the right-and-side of eq. (13) corresponds to the correction used in the conventional directional-splitting method for a divergence-free advection velocity. In the present work, we extend the directional-splitting advection method in eq. (13) to ensure that the corresponding non-zero interface velocity divergence is accurately prescribed:

$$C_{i,j,k}^{n+1} = C_{i,j,k}^{***} - \Delta t F_{i,j,k}^n + \Delta t C_{i,j,k}^{n+1} (\nabla \cdot \mathbf{u}_\Gamma)_{i,j,k}^n. \quad (15)$$

The last term in eq. (15) can be seen as an implicit volume deflation step, and ensures that the correct value of the velocity divergence is used to update C to time level $n + 1$; $(\nabla \cdot \mathbf{u}_\Gamma)_{i,j,k}^n$ is the discrete divergence of \mathbf{u}_Γ at time level n .

Once C^{n+1} is determined, the thermophysical properties are updated using eq. (2), where the volume fraction field C is used as a smoothed approximation of the phase indicator function H . This makes all terms involving thermophysical properties numerically differentiable.

3.2. Flow solver

The two-fluid Navier-Stokes solver uses a projection method [39] with the pressure-splitting technique described in [40], reducing the pressure-correction step to a constant-coefficients Poisson equation. The underlying idea is to split the variable-coefficients pressure Poisson equation into two parts: a variable-coefficients term that is treated explicitly by extrapolating the pressure field into the current time level, and a constant-coefficients term. This allows for using efficient FFT-based direct solvers for the second-order finite-difference Poisson equation with constant coefficients as shown in [41], which are about one order of magnitude faster than a standard iterative solver. We should note however that the overall method can be easily applied to standard two-fluid solvers that do not use this pressure-splitting technique. The overall solution procedure uses an Adams-Bashforth method to advance the solution from time step n to $n + 1$, and is summarized below in semi-discrete form:

$$\hat{p} = \left(1 + \frac{\Delta t^{n+1}}{\Delta t^n} \right) p^n - \left(\frac{\Delta t^{n+1}}{\Delta t^n} \right) p^{n-1}, \quad (16)$$

$$\begin{aligned} \rho^{n+1} \left(\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t^{n+1}} \right) &= \left(1 + \frac{1}{2} \frac{\Delta t^{n+1}}{\Delta t^n} \right) \mathbf{RHS}^n - \left(\frac{1}{2} \frac{\Delta t^{n+1}}{\Delta t^n} \right) \mathbf{RHS}^{n-1}, \\ \frac{\nabla^2 p^{n+1}}{\rho_0} &= \nabla \cdot \left[\left(\frac{1}{\rho_0} - \frac{1}{\rho^{n+1}} \right) \nabla \hat{p} \right] + \end{aligned} \quad (17)$$

$$\frac{1}{\Delta t^{n+1}} \left[\nabla \cdot \mathbf{u}^* - \dot{m}^{n+1} \left(\frac{1}{\rho_2} - \frac{1}{\rho_1} \right) \delta_\Gamma^C \right], \quad (18)$$

$$\mathbf{u}^{n+1} = \mathbf{u}^* - \Delta t^{n+1} \left[\frac{1}{\rho_0} \nabla p^{n+1} + \left(\frac{1}{\rho^{n+1}} - \frac{1}{\rho_0} \right) \nabla \hat{p} \right], \quad (19)$$

where \mathbf{u}^* is the predicted velocity, Δt^{n+1} and Δt^n represent the time-step computed at time step $n+1$ and n to fulfill the temporal stability requirements, $\rho_0 = \min(\rho_1, \rho_2)$, **RHS** denotes the discretized advection, diffusion, gravity and surface tension terms in eq. (5); $\delta_\Gamma^C = |\nabla C^{n+1}|$ is a regularized Dirac delta function approximating δ_Γ in eqs. (4),(5) and (10), and is discretized using the Youngs method [42]. The continuum surface force model (CSF) proposed by Brackbill [43] is used for discretizing the surface tension term.

This set of equations is very close to that of [41], except for the last term of eq. (18). One can easily see that this term approximates the right velocity divergence at the interface due to phase change, i.e. eq. (4), regularized over the grid cells where $0 < C < 1$.

This CSF-like approach makes the final velocity field numerically differentiable. Still, the flow velocity \mathbf{u} may vary strongly across the interface, and therefore the use of second-order central schemes for the spatial derivatives of the convective terms in **RHS** is not desirable. Accordingly, a QUICK scheme [44] is employed for the convective term (discretized as $\nabla \cdot (\mathbf{u} \otimes \mathbf{u}) - \mathbf{u} \nabla \cdot \mathbf{u}$), while second-order central differences are used for the diffusion terms.

Our method uses the fast and versatile FFT-based DNS code *CaNS* [45] as base Navier-Stokes solver. This solver allows for several combinations of homogeneous pressure boundary conditions, which is particularly convenient for the computational setups used in the present work. Finally, validations of the interface-capturing procedure in absence of phase change have been reported in [46], and its suitability for simulating complex and turbulent flows demonstrated in [47,48].

It is worth remarking that the use of the pressure-splitting approach for solving the Poisson equation can be beneficial for interface resolved simulation of phase-changing problems, especially those involving high mass transfer rates between phases. In fact, as noted also in [30], as \dot{m} increases and as the resolution to properly resolve the governing equations is increased, the source term on the right-hand side of eq. (18) becomes more and more important, and might pose a stiffness problem. Hence, if a variable-coefficient pressure Poisson equation is solved using an iterative method, the convergence of the solver may be slow. This is not an issue in the present method, since our approach allows for a fast direct solver.

3.3. Vapor mass transport

Equation (6) is solved in Ω_2 with the Dirichlet boundary condition prescribed at the interface in eq. (7). This equation is discretized in time as follows:

$$\frac{Y_2^{l,n+1} - Y_2^{l,n}}{\Delta t} = -\mathbf{u}^n \cdot \nabla Y_2^{l,n} + D_{lg} \nabla^2 Y_2^{l,m}, \quad (20)$$

with $m = n+1$ or n , depending on whether the diffusion term is discretized implicitly or explicitly. The spatial discretization needs to be modified close to the interface to prescribe the boundary condition at $\mathbf{x} = \mathbf{x}_\Gamma$. To achieve this, the finite-difference stencil is modified in grid cells close to interface by constructing a signed distance (i.e. level-set) field ϕ from C , with the method proposed in [49,50]. Here $\phi > 0$ corresponds to Ω_1 , and $\phi < 0$ to Ω_2 .

The advection term is discretized using an upwind scheme; see e.g. [30]. Taking the discretization in x as example, it reads:

$$\mathbf{u} \cdot \nabla Y_2^l = \frac{(u_c + |u_c|)}{2} \frac{\partial Y_2^l}{\partial x} \Big|_- + \frac{(u_c - |u_c|)}{2} \frac{\partial Y_2^l}{\partial x} \Big|_+, \quad (21)$$

where u_c is the x -velocity component interpolated into the cell center (i.e. $u_c = (u_{i+1/2,j,k}^c + u_{i-1/2,j,k}^c)/2$). When the interface crosses a grid cell, the discretized form of the gradients of Y_2^l should be modified to conform to the interface boundary condition. This is achieved by considering a higher-order one-sided difference on an irregular stencil. For instance, the gradient of Y_2^l at $i - 1/2$ is computed as follows (the procedure for $i + 1/2$ is analogous):

$$\frac{\partial Y_2^l}{\partial x} \Big|_- = \begin{cases} \beta_0 Y_{2,\Gamma}^l(T_{\Gamma,x}^-) + \sum_{p=0}^3 \beta_{p+1} Y_{2,i+p}^l & \text{if } \phi_{i-1} \phi_i < 0, \\ \frac{Y_{2,i}^l - Y_{2,i-1}^l}{\Delta x} & \text{otherwise,} \end{cases} \quad (22)$$

where $Y_{2,\Gamma}^l$ is computed from eqs. (7) and (8), with the interface temperature $T_{\Gamma,x}^-$ estimated from the neighboring values of ϕ [51,20]:

$$T_{\Gamma,x}^- = \frac{T_{i-1,j,k} |\phi_{i,j,k}| + T_{i,j,k} |\phi_{i-1,j,k}|}{|\phi_{i-1,j,k}| + |\phi_{i,j,k}|}. \quad (23)$$

In eq. (22), the one-sided difference coefficients β are computed following the approach reported in [52], as in [53]. The resulting stencil is given by $x = x_i + \{-\theta_x^-, 0, 1, 2, 3\}\Delta x$. Using the level-set function, the coefficient $\theta_x^- = (x_i - x_\Gamma)/\Delta x$ is computed as proposed in [51]:

$$\theta_x^- = \frac{|\phi_{i,j,k}|}{|\phi_{i-1,j,k}| + |\phi_{i,j,k}|}. \quad (24)$$

In case of small values of $\theta_x^- < 0.25$, the point x_i is removed from the one-sided difference stencil to prevent errors as it approaches a singular value [53]. Finally, the above procedure is performed in a dimension-by-dimension manner for directions y and z .

Note that the aforementioned approach to discretize the vapor mass equation is not sufficiently robust when complex topological changes such coalescence and merging occur. Equation (22) requires a stencil of three grid cells in the vapor domain and, therefore, when two or multiple droplets start to coalesce and merge, numerical problems occur since there are not enough vapor grid cells for an accurate estimation of the gradient of Y_2^l . In these cases, we decide to reduce the stencil of equation (22) such that $x = x_i + \{-\theta_x^-, 0, 1\}\Delta x$ and compute β accordingly. Even though this approach reduces the accuracy of the calculation of the vapor mass gradient, we show in the Results Section (Case 4.5) that it still provides stable and grid convergent results.

If an explicit temporal discretization of eq. (20) is employed, the second derivatives in the diffusion term are discretized using the first derivatives, previously computed. Taking once more the term in x as an example, the second derivative reads [54]:

$$\frac{\partial^2 Y_2^l}{\partial x^2} = \frac{1}{\Delta x} \left(\frac{\partial Y_2^l}{\partial x} \Big|_+ - \frac{\partial Y_2^l}{\partial x} \Big|_- \right). \quad (25)$$

Conversely, if an implicit discretization is chosen, the procedure involves solving an Helmholtz equation where the boundary condition in eq. (7) is prescribed with the aid of the constructed level-set field [17,20]. The resulting symmetric definite positive linear system is solved with the parallel semicoarsening multigrid solver (PFMG) of the HYPRE library [55].

3.4. Interfacial vapor mass flux

The calculation of the interfacial vapor mass flux is accomplished using eq. (9), written in terms of \dot{m} :

$$\dot{m} = -\frac{\rho_2 D_{lg}}{1 - Y_{2,\Gamma}^l} \nabla_\Gamma Y_2^l \cdot \mathbf{n}. \quad (26)$$

Since Y_2^l is not defined in Ω_1 , standard finite differences cannot be directly used to approximate the gradient at the interface. Instead, we follow the approach firstly proposed in [56] and used in e.g. [18] and extend the Y_2^l into Ω_1 . Hence, before computing \dot{m} from eq. (26), Y_2^l is extrapolated into the liquid domain using a second-order pde-based extrapolation. This involves the successive solution to steady state of the three following equations:

$$\begin{aligned} \frac{\partial Y_{2,nn}^{l,e}}{\partial \tau} + H_\phi(\phi + a_2 \Delta l) \mathbf{n} \cdot \nabla Y_{2,nn}^{l,e} &= 0, \\ \frac{\partial Y_{2,n}^{l,e}}{\partial \tau} + H_\phi(\phi + a_1 \Delta l) (\mathbf{n} \cdot \nabla Y_{2,n}^{l,e} - Y_{2,nn}^{l,e}) &= 0, \\ \frac{\partial Y_2^{l,e}}{\partial \tau} + H_\phi(\phi + a_0 \Delta l) (\mathbf{n} \cdot \nabla Y_2^{l,e} - Y_{2,n}^{l,e}) &= 0, \end{aligned} \quad (27)$$

where $Y_{2,n}^{l,e}$ and $Y_{2,nn}^{l,e}$ are the first and the second derivative of Y_2^l along the Ω_1 -pointing interface normal, H_ϕ represents a cosine regularized Heaviside function [57,58], computed from the level-set function ϕ and τ is a pseudo time used only to advance to the steady state solution eqs. (27). The offsets $a_i \Delta l$ (with Δl being the grid spacing) ensure that only values in the vapor domain, i.e. at $\phi < 0$ are used to solve eqs. (27). When implicit diffusion is used to discretize eq. (20), the values suggested in the original reference suffice for an accurate extrapolation of the vapor field onto the liquid domain. However, if this equation is discretized explicitly, care should be taken when the phase in a certain grid cell changes (e.g. from liquid to gas). If the solution is not allowed to gradually adapt itself (within a few time steps) to the sudden change in phase, numerical errors may occur. For this reason, in this case, the solution is extrapolated from a position slightly inward of the interface. The corresponding shifted offset is now given by $(a_i + 1)\Delta l$ (i.e. one grid size inward of the interface).

It is worth noting that modifications in the computation of derivatives of mass fraction in the extrapolation procedure are needed e.g. when two or more droplets are too close to each other. When the finite-difference stencil for the computation of a derivative at two liquid grid cells separated a thin gas layer, the spatial derivatives of the mass fraction are approximated as follows:

$$\frac{\partial Y_{2,n}^l}{\partial x} = \begin{cases} \frac{Y_{2,\Gamma}^l(T_\Gamma^+) - Y_{2,\Gamma}^l(T_\Gamma^-)}{|x_\Gamma^+ - x_\Gamma^-|} & \text{if } \varepsilon \Delta x < |x_\Gamma^+ - x_\Gamma^-| < \Delta x, \\ 0 & \text{if } |x_\Gamma^+ - x_\Gamma^-| < \varepsilon \Delta x, \end{cases} \quad (28)$$

where we set $\varepsilon = 0.25$; T_Γ^+ and T_Γ^- are computed using equation (23), and x_Γ^+ and x_Γ^- represent the two interface locations computed using $\phi_{i+1,j}$ and $\phi_{i-1,j}$. This is to say that if two liquid interfaces separated by a thin layer of gas with thickness smaller than $0.25\Delta x$, we consider the gradient to be under-resolved and set it to zero.

Eqs. (27) are integrated in time using about 15 forward Euler pseudo time steps, and a first-order upwind scheme [18] to advect the scalar fields. The resulting extended vapor species field $Y_{2,\Gamma}^{l,e}$ is then used to compute the gradients in eq. (26) using a central difference scheme. Moreover, in the calculation of the interfacial mass flux we use the unit normal vector computed from the VoF field directly from its definition, i.e., $\mathbf{n} = \nabla C / |\nabla C|$. As done for the regularized Dirac delta function in eqs. (4), (5) and (10), the gradients of the color function are evaluated using finite differences, following the Youngs approach [42]. Accordingly, \mathbf{n} can be directly employed in eq. (26) and in the calculation of the interface velocity (see section 3.6).

The interfacial values required for computing the mass flux in eq. (26) are not extended in a band around the interface. Instead, we take the simple approach of approximating these quantities by the local extrapolated field quantities, i.e. $Y_{2,\Gamma}^l \approx Y_2^{l,e}$, and $\nabla_\Gamma Y_2^l \cdot \mathbf{n} \approx \nabla Y_2^{l,e} \cdot \mathbf{n}$ [20,21]. Potential improvements of this approach are a constant extrapolation of the interfacial mass flux to a band around the interface [18], or the approach in [59] that extrapolates the interface temperature to a band around the interface, and computes $Y_{2,\Gamma}^l$ from this extrapolated temperature field. We have tested these approaches and observed a marginal improvement in the accuracy of mass conservation for the cases presented in this manuscript.

Overall, the approach described in this section allows us to define \dot{m} in a band of about $6\Delta l$ around the interface, which covers the region where $0 < C < 1$.

3.5. Energy equation

All the terms in the energy equation (10) are discretized explicitly using an Adams-Bashforth scheme whose coefficients are computed for a variable time step:

$$\begin{aligned} \frac{T^{n+1} - T^n}{\Delta t^{n+1}} &= \left(1 + \frac{1}{2} \frac{\Delta t^{n+1}}{\Delta t^n}\right) \left(-\mathbf{u} \cdot \nabla T + \frac{1}{\rho c_p} \text{RHS}_T\right)^n \\ &\quad - \left(\frac{1}{2} \frac{\Delta t^{n+1}}{\Delta t^n}\right) \left(-\mathbf{u} \cdot \nabla T + \frac{1}{\rho c_p} \text{RHS}_T\right)^{n-1}, \end{aligned} \quad (29)$$

where RHS_T denotes the discretized form of the diffusive and source terms in eq. (10), and the temperature field around the interface is used as an approximation of T_Γ in the source term. Note again that, as specified after eq. (2), we employ the harmonic average for the term ρc_p to perform a consistent discretization of the diffusion and jump terms, as eq. (29) suggests. In fact, we find that the use of the arithmetic mean may lead to unphysical values in the temperature field. Our spatial discretization uses the 5th-order WENO scheme described in [60], while the diffusion term is treated using standard central differences. The Dirac delta function in the singular term is regularized as explained above for the discretization of the momentum equations.

Finally, it is worth noting that eq. (29) is discretized in non-conservative form, which becomes more and more inaccurate as both the density and specific heat capacity ratio increase. We have therefore compared the numerical results employing the conservative and the non-conservative discretization of the energy equation for a vaporizing droplet in the Results Section (Case 4.3). Indeed, we for low resolutions the conservative scheme shows better agreement with the reference data. Nevertheless, as the grid is refined the results converge to the same numerical value, regardless of the formulation.

3.6. Interface velocity construction

The calculation of the interface velocity \mathbf{u}_Γ for the advection of C is a key aspect of this method. The main challenge stems from the discontinuity of the one-fluid-formulation velocity \mathbf{u} across the interface, while \mathbf{u}_Γ as defined in eq. (3) is continuous.

To overcome this issue, we adopt a strategy that accurately extends the liquid velocity into the gas domain, making the extended field differentiable in Ω . The basic idea is that phase change induces a Stefan flow, which is responsible for the jump in the flow velocity \mathbf{u} . Hence, a divergence-free liquid-velocity extension can be obtained by subtracting this jump to \mathbf{u} . The Stefan flow velocity \mathbf{u}^S can be obtained from a velocity potential φ as follows:

$$\begin{cases} \nabla^2 \varphi &= \dot{m} \left(\frac{1}{\rho_2} - \frac{1}{\rho_1} \right) |\nabla C|, \\ \mathbf{u}^S &= \nabla \varphi. \end{cases} \quad (30)$$

The solution of the first equation in (30) can be, once more, obtained for a wide variety of boundary conditions for φ using the efficient FFT-based direct solver described in [45]. The main advantage of this approach is that, since \mathbf{u}^S is defined in the entire domain, an extended liquid velocity \mathbf{u}_1^e can be easily computed:

$$\mathbf{u}_1^e = \mathbf{u} - \mathbf{u}^S. \quad (31)$$

It can be easily shown that \mathbf{u}_1^e is divergence-free by construction. Moreover, for the same reason mentioned in section 3.2, the possibility to use direct solvers for eq. (30) allows to circumvent potential stiffness problems induced by high values of the interfacial mass flux.

Note that a similar idea has been recently developed in [61] for boiling flows, by solving for an extended velocity derived from a density-weighted variable coefficient Poisson equation, reported here for completeness:

$$\begin{cases} \nabla \cdot \left(\frac{1}{\rho} \nabla \varphi \right) = \dot{m} \left(\frac{1}{\rho_2} - \frac{1}{\rho_1} \right) |\nabla C|, \\ \mathbf{u}^S = \frac{1}{\rho} \nabla \varphi. \end{cases} \quad (32)$$

Although equation (32) directly accounts for the density contrast between the two phases like the variable-density pressure Poisson equation, it has the disadvantage of requiring the solution of a variable-coefficient Poisson equation that cannot be solved using fast FFT-based direct solvers.

In a similar way to what it is commonly done in the prediction/correction procedure for the Navier-Stokes solver, the boundary conditions for φ and \mathbf{u}_1^e have to be prescribed consistently, and are set to be the same as those of the pressure and prediction velocity, respectively.

Finally, eq. (3) can be used to compute \mathbf{u}_Γ :

$$\mathbf{u}_\Gamma = \mathbf{u}_1^e - \frac{\dot{m}}{\rho_1} \mathbf{n}. \quad (33)$$

where, again, the same boundary conditions used for \mathbf{u}_1^e are employed for \mathbf{u}_Γ . The normal vector \mathbf{n} in equation (33) is computed as detailed in section 3.4.

The interface velocity \mathbf{u}_Γ is then used to advect C , as described in section 3.1. Furthermore, it is important to stress that following this approach both \mathbf{u}_1^e and \mathbf{u}_Γ are equal to the one-fluid velocity derived from the Navier-Stokes solver \mathbf{u} in those region away from the interface where \dot{m} is not defined. Note that in principle the procedure here described can be applied also to those interface capturing methods based e.g. on level-set and conservative level-set methods while maintaining the whole domain formulation for solving the momentum equations. Few words should be also spent about the overhead associated with the computation of \mathbf{u}_Γ . For a three-dimensional case (evaluated for the last setup described in section 4.6), we found that it is relatively small, i.e. less than 2% of the total wall-clock time per time step.

We should note that C may also be advected directly using \mathbf{u}_1^e and a source term accounting for phase change on the right-hand-side. Inserting eq. (33) into (11), we get:

$$\frac{\partial C}{\partial t} + \nabla \cdot (\mathbf{u}_1^e H^{ht}) = -\frac{\dot{m}}{\rho_1} |\nabla C|. \quad (34)$$

Since \mathbf{u}_1^e is divergence-free, the original splitting advection in [38] can be used, followed by a step that accounts for the source term [61]. Test simulations reproducing the benchmarks in section 4 showed that the circular shape of a vaporizing droplet in a quiescent medium is better preserved (lower errors in the perimeter of the droplet) when the phase-change term is explicitly accounted for in the split advection steps (eq. 12), i.e. when \mathbf{u}_Γ is explicitly used to advect the interface as described above in section 3.1.

3.7. Overall solution procedure and time marching

We briefly summarize the overall solution procedure, which is also outlined in Fig. 1 for clarity. First, the volume-of-fluid function is advanced to C^{n+1} , the corresponding interface normal vector and curvature are updated, and the level-set is reconstructed. Next, the vapor mass-fraction equation is solved to obtain $Y_2^{l,n+1}$, and the interfacial mass flux \dot{m}^{n+1} is computed. Then the energy and Navier-Stokes equations can be advanced in time to obtain \mathbf{u}^{n+1} , p^{n+1} and T^{n+1} . Finally, the time marching ends with the computation of the interface velocity \mathbf{u}_Γ^{n+1} , which is used to transport C at the next time level.

The time step Δt^{n+1} is estimated from the stability constraints of the overall system:

$$\Delta t^{n+1} = C_{\Delta t} \min(\Delta t_c, \Delta t_\sigma, \Delta t_\mu, \Delta t_m, \Delta t_e)^{n+1}, \quad (35)$$

where Δt_c , Δt_σ , Δt_μ , Δt_m and Δt_e are the maximum allowable time steps due to convection, surface tension, momentum diffusion, vapor mass diffusion and thermal energy diffusion. These are determined as suggested in [62]:

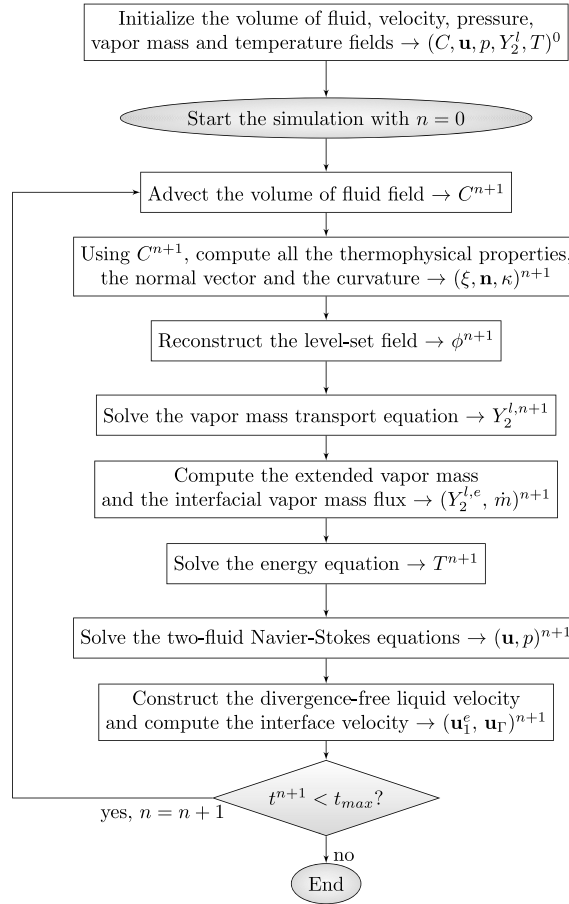


Fig. 1. Diagram illustrating the overall solution procedure. t^{n+1} is the physical corresponding to time-step $n + 1$ and t_{max} is the maximum physical time reached in the simulation.

$$\begin{aligned}
 \Delta t_c &= \left(\frac{|u_{x,\max}|}{\Delta x} + \frac{|u_{y,\max}|}{\Delta y} + \frac{|u_{z,\max}|}{\Delta z} \right)^{-1}, \\
 \Delta t_\sigma &= \sqrt{\frac{(\rho_1 + \rho_2) \min(\Delta x^3, \Delta y^3, \Delta z^3)}{4\pi\sigma}}, \\
 \Delta t_\mu &= \left[\max\left(\frac{\mu_1}{\rho_1}, \frac{\mu_2}{\rho_2}\right) \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2} \right) \right]^{-1}, \\
 \Delta t_m &= \left[D_{lg} \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2} \right) \right]^{-1}, \\
 \Delta t_e &= \left[\max\left(\frac{k_1}{\rho_1 c_{p,1}}, \frac{k_2}{\rho_2 c_{p,2}}\right) \left(\frac{2}{\Delta x^2} + \frac{2}{\Delta y^2} + \frac{2}{\Delta z^2} \right) \right]^{-1},
 \end{aligned} \tag{36}$$

where $|u_{i,\max}|$ is an estimate of the maximum value of the i th component of the flow velocity; Δt_m is only considered when the vapor mass diffusion term is discretized explicitly. Setting $C_{\Delta t} = 0.35$ in the present work was seen to be sufficient for a stable and accurate time integration.

Finally, we should note that our framework can be easily adapted to a two-phase flow undergoing temperature-induced phase change in a single-component system, i.e. boiling. This procedure is described in Appendix A.

4. Results

We present now a validation of our method against several benchmark cases. For clarity, the physical parameters defining the different setups are displayed in Table 1. Unless otherwise stated, the time step is set to be constant and determined from eq. (35) with $C_{\Delta t} = 0.35$, and the diffusion term in eq. (20) governing Y_2^l is discretized implicitly.

Table 1

Governing parameters for the cases presented in this section. Their definitions are given in the “Governing parameters” section. For all cases $l_{ref} = d_0$, where d_0 is the initial droplet diameter.

Section	Re	We	Fr	Pr	Sc	Ste	Ste ^m	λ_ρ	λ_μ	λ_{c_p}	λ_k	λ_M	u_{ref}	T_{ref}
4.1	25	0.1	∞	N/A	N/A	N/A	N/A	10 – 100	50	N/A	N/A	N/A	\dot{m}_0/ρ_2	N/A
4.2	25	0.1	∞	N/A	0.04	N/A	N/A	10 – 100	50	N/A	N/A	N/A	D_{lg}/d_0	N/A
4.3	1.50	$3.31 \cdot 10^{-5}$	∞	0.69	0.67	varied	0.07	8.33	63.68	4.16	0.23	0.62	D_{lg}/d_0	$T^{sat} - T_\infty$
4.4	8.84	3.125	1	0.7	1	varied	0.07	10	50	4.0	2.0	0.62	$\sqrt{ g d_0}$	$T^{sat} - T_\infty$
4.5	25	0.7	∞	0.7	0.035	0.17	0.0175	10	50	4.0	2.0	0.62	$u_{r,0}$	$T^{sat} - T_\infty$
4.6	90	0.5	1	0.7	0.065	0.08	0.75	10	50	4.0	2.5	0.62	$\sqrt{ g d_0}$	$T^{sat} - T_\infty$

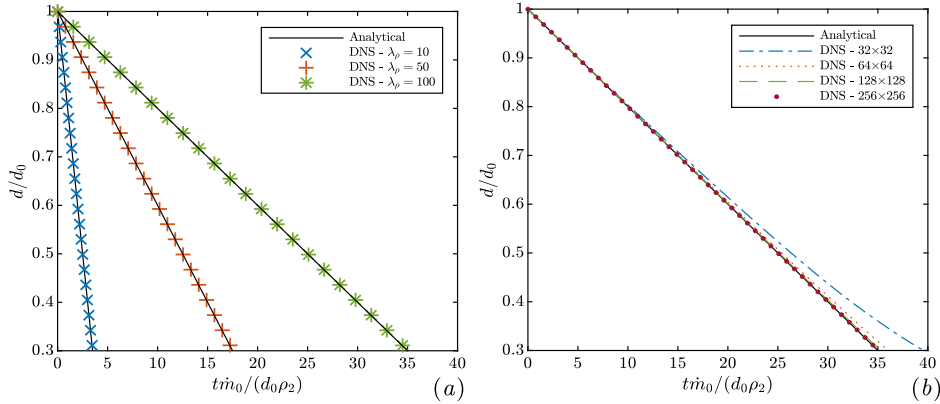


Fig. 2. (a): temporal evolution of the droplet diameter for $\lambda_\rho = 10, 50, 100$, on a 256×256 grid. (b): grid convergence test for an evaporating droplet with $\lambda = 100$.

4.1. Droplet evaporation due to a prescribed, constant mass flux

This case considers a phase-changing two-dimensional circular droplet, where evaporation is driven by a constant mass flux \dot{m}_0 . This simple configuration allows us to verify the numerical method for phase-changing two-fluid flows, decoupled from the transport equations of energy and vapor mass. Under these conditions, it is easy to show that the droplet diameter evolves in time as follows [20]:

$$\frac{d(t)}{d_0} = 1 - \left(\frac{2\dot{m}_0}{d_0\rho_1} \right) t. \quad (37)$$

The circular droplet has an initial diameter d_0 , it is centered in a square domain with dimensions $[-2d_0, 2d_0]^2$, and zero-pressure outflow boundaries. The corresponding physical parameters have been reported in Table 1. We are in particular interested in assessing the ability of the method to handle interface-normal velocity jumps across Ω (see eq. (3)). In order to test different magnitudes of this jump, we consider three density ratios, $\lambda_\rho = \{10, 50, 100\}$ while keeping the other parameters fixed.

Fig. 2(a) shows the time evolution of the normalized droplet diameter, computed from its volume, for different λ_ρ , on the finest grid considered (256×256). The numerical results show excellent agreement with the analytical solution in eq. (37). Panel (b) of the same figure shows the solution grid convergence for the case with the highest velocity jump, $\lambda_\rho = 100$. The results illustrate a very good agreement even for relatively coarse grids. This result is expected from how the interface velocity \mathbf{u}_Γ is constructed (eq. (33)). We recall that the volume deflation term in eq. (15), controlling the bulk value of C , is proportional to the interface velocity divergence. The first term contributing to \mathbf{u}_Γ in eq. (33) is a divergence-free extension of the liquid velocity, which effectively conserves the total volume-of-fluid to machine precision. The second term contains the interfacial mass flux and the interface normal. Since \dot{m} is constant in this example, the only source of numerical error in the droplet volume is the divergence of the interface normal vector, i.e. the local curvature. For an evaporating static droplet, the liquid velocity should be zero. We therefore expect the divergence-free liquid velocity extension to be $\mathbf{u}_1^e = 0$, and thus $\mathbf{u}_\Gamma = -(\dot{m}_0/\rho_1)\mathbf{n}$. This is illustrated in Fig. 3, where we display the vector field \mathbf{u} in panel (a), \mathbf{u}_1^e in (b) and \mathbf{u}_Γ in (c) for a physical time corresponding to $d/d_0 = 0.9$ and $\lambda_\rho = 100$. Panel (a) shows the expected velocity field \mathbf{u} , with a clear jump across Γ . The liquid velocity extension shown in panel (b) is, expectedly, very small, with spurious velocities about three orders of magnitude smaller than the maximum value of \mathbf{u} . Finally, the interface velocity field shows the expected values for an evaporating droplet (note that \mathbf{n} is only defined where $0 < C < 1$, and so is \mathbf{u}_Γ).

Finally, we analyze the accuracy of our method by inspecting the convergence of the droplet mass and shape errors with increasing resolution. We consider the case with most significant velocity jump, $\lambda_\rho = 100$ and a time step $\Delta t =$

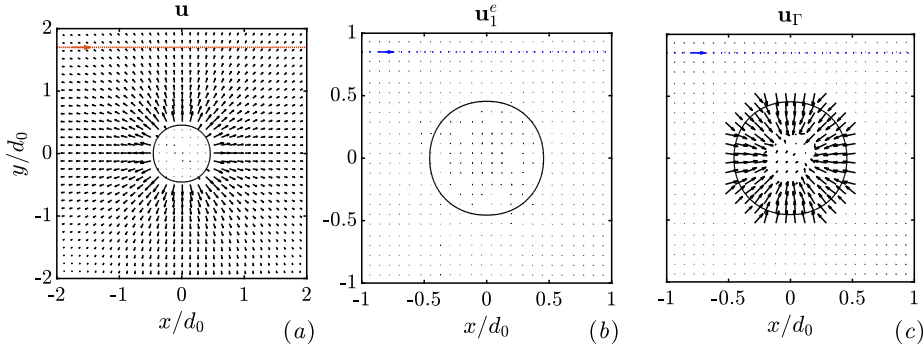


Fig. 3. Vector field of (a) flow velocity \mathbf{u} , (b) divergence-free extended liquid velocity \mathbf{u}_l^e , and (c) interface velocity \mathbf{u}_Γ , at $d/d_0 = 0.9$ for uniform evaporation on a 256×256 grid and $\lambda_\rho = 100$. The interface is represented by the solid black line. The vector scale (see top-left reference arrows) is equal to $\|\mathbf{u}\|_{\max} = \dot{m}(1/\rho_2 - 1/\rho_1)$ for panel (a), and to $0.01\|\mathbf{u}\|_{\max}$ for panels (b) and (c). Panels b and c zoom into the domain $[-d_0, d_0]^2$ and the values of \mathbf{u}_l^e and \mathbf{u}_Γ outside the range where $\min(C, 1 - C) < 10^{-8}$ have been clipped.

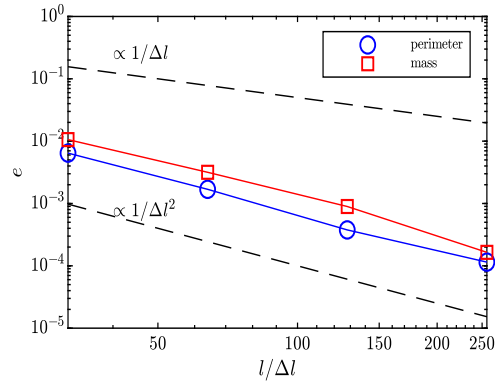


Fig. 4. Error in the normalized droplet mass, $M/M_{t=0}$, and perimeter, $P/P_{t=0}$ for $\lambda_\rho = 100$ and $d/d_0 = 0.4$. Δl and l denote the grid spacing and domain length.

$0.0075 d_0 \rho_2 / \dot{m}_0$, sufficiently low for errors in the temporal discretization to be negligible. Moreover, we ensure that the initial condition for C yielded the same total droplet mass for all grids. The shape accuracy is assessed by computing the perimeter from the discrete integral of $|\nabla C|$ over the entire domain as follows (for a two-dimensional configuration):

$$P = \sum_{j=1}^{N_y} \sum_{i=1}^{N_x} |\nabla C|_{i,j} \Delta x \Delta y, \quad (38)$$

where N_x and N_y are the number of grid points along the x and y direction. Fig. 4 reports the relative error in mass and perimeter, at physical time t for which $d(t)/d_0 = 0.4$, i.e.:

$$e(\Delta l, t) = \frac{|\chi_n(\Delta l, t) - \chi_a(\Delta l, t)|}{\chi_a(\Delta l, t)}, \quad (39)$$

where $\Delta l = \Delta x = \Delta y$, χ represents the mass or the perimeter and the subscript n and a refer to the numerical and analytical solution. For both observables, the order of convergence is between 1.75 and 2.0. Similar results have been obtained for the other density ratios considered here.

4.2. Isothermal droplet evaporation

This test case considers the evaporation of a two-dimensional circular droplet, using the same domain and boundary conditions as in the previous section. The difference is that the evaporation is driven by a difference between the species concentration at the interface and the domain boundary. Hence, we now solve eq. (20) with Dirichlet boundary conditions $Y_{2,\infty}^I = 0$ at the domain boundaries, and $Y_{2,\Gamma}^I = 0.5$ at the interface. The mass flux \dot{m} is then computed from eq. (26). This allows us to assess the accuracy of the method for coupling the vapor species transport to the Navier-Stokes equations, decoupled from the energy equation. The other physical parameters are reported in Table 1, and are chosen to match the

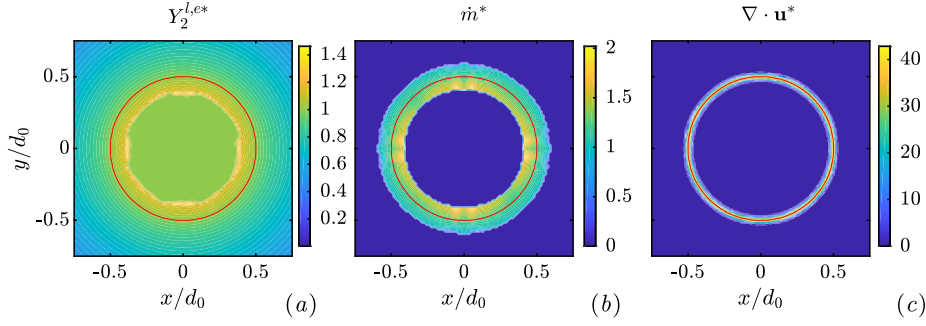


Fig. 5. Isothermal evaporation of a static droplet. (a): contours of the normalized extended vapor mass fraction field $Y_2^{l,e*} = (Y_2^{l,e} - Y_{2,\infty}^{l,e}) / (Y_{2,\Gamma}^{l,e} - Y_{2,\infty}^{l,e})$; (b): contours of the dimensionless interfacial mass flux $\dot{m}^* = \dot{m}d_0 / (\rho_2 D_{lg})$ and, (c): contours of the dimensionless mass source term $\nabla \cdot \mathbf{u}^* = (d_0^2 / D_{lg}) \dot{m} (1/\rho_2 - 1/\rho_1) \delta_\Gamma$ at $tD_{lg}/d_0^2 = 0.065$. The interface location is depicted by the solid red line. The contour plots are restricted to the sub-domain $[-0.75d_0, 0.75d_0]^2$. (For interpretation of the colors in the figure(s), the reader is referred to the web version of this article.)

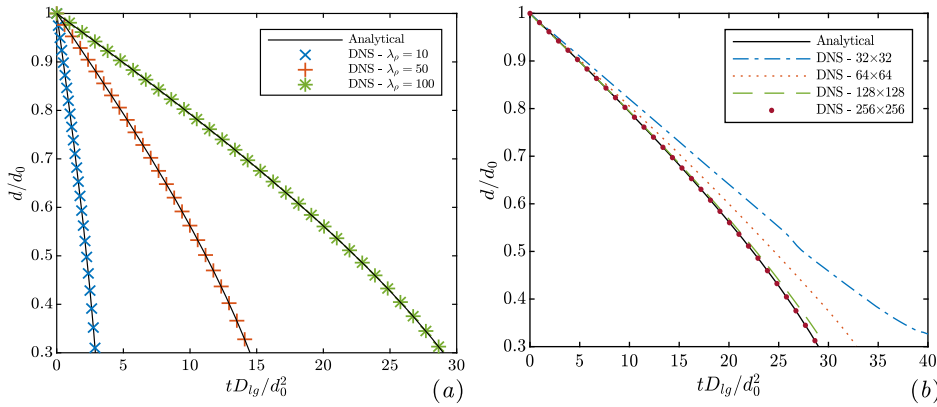


Fig. 6. (a): temporal evolution of the normalized droplet diameter for varying λ_ρ , on a 256×256 grid. (b): grid convergence test for $\lambda_\rho = 100$. The DNS discretizes the diffusion term in eq. (20) for Y_2^l implicitly.

initial shrinking rate of the setup presented in the previous section. The steady state solution of eq. (6) with $\mathbf{u} = 0$ is used as initial condition for Y_2^l .

For this system one can derive an ordinary differential equation for the droplet diameter d , assuming a static droplet in a circular domain with diameter D [12]:

$$\frac{dd^2}{dt} = -\frac{8\rho_1 D_{lg}}{\rho_2} \frac{\ln(1 + B_Y)}{\ln(D/d)}, \quad (40)$$

where $B_Y = (Y_{2,\Gamma}^l - Y_{2,\infty}^l) / (1 - Y_{2,\Gamma}^l)$ is a mass number. In the present setup $D = 4d_0$ is the domain length, and $B_Y = 1$. The equation above can therefore be solved numerically, yielding a reference solution for $d(t)$.

Fig. 5(a) reports the non-dimensional extended vapor mass fraction field for the case of $\lambda_\rho = 100$ evaluated at the highest resolution (256×256). In the gas phase, $Y_2^{l,e}$ is equal to the vapor mass field, whereas inside it is computed with the procedure explained in section 3.4 and only in the region where $0 < C < 1$. This field is used to compute the interfacial mass flux depicted in Fig. 5(b), to advect the interface at the next time-step and as a source term in the pressure Poisson equation. As shown in Fig. 5(c), this source term is zero everywhere except in a narrow region across the interface. The proposed approach leads to good numerical results in term of mass and droplet shape conservation when compared to the analytical solution, as shown next.

Fig. 6(a) presents the time history of the normalized droplet diameter, computed from its volume (i.e., in $2D$, $d = \sqrt{4A/\pi}$ where A is the surface area), for different values of λ_ρ , on the finest grid considered. The solution grid convergence is shown in panel (b) of the same figure, for $\lambda_\rho = 100$. Once more, the results show excellent agreement with the reference data, although, as expected, the deviations from the analytical solution are larger than those reported in the previous section as the interfacial mass flux \dot{m} is here computed from the vapor mass gradient and not imposed. Though these cases correspond to an implicit discretization of the diffusion of Y_2^l in eq. (20), similar results are obtained with an explicit treatment of the diffusion term. Fig. 7 reports the convergence of the numerical error with grid refinement with (panel a) and without (panel b) implicit diffusion for $\lambda_\rho = 100$, $d/d_0 = 0.4$ and fixed time step $\Delta t = 0.005 d_0^2 / D_{lg}$. As we can see, the numerical error

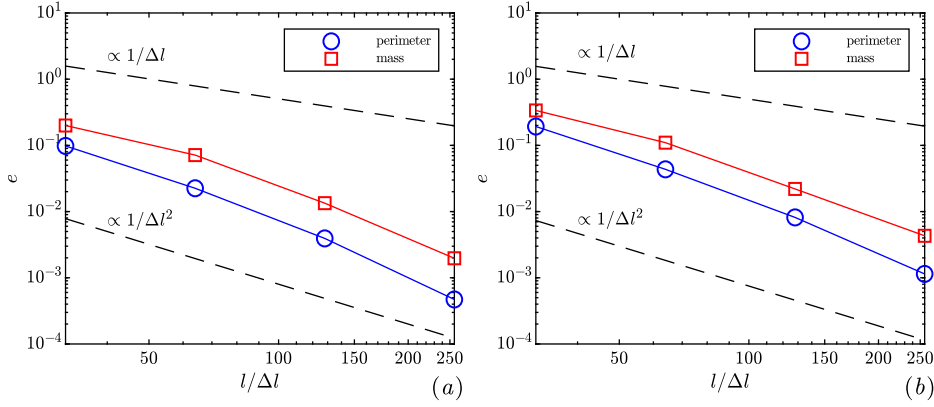


Fig. 7. Error in the normalized droplet mass, $M/M_{t=0}$, and perimeter, $P/P_{t=0}$ for $\lambda_\rho = 100$ and $d/d_0 = 0.4$. Δl and l denote the grid spacing and domain length. The two panels pertain to simulations with (a) implicit and (b) explicit treatment of the vapor mass diffusion term in eq. (20).

Table 2

Thermophysical properties of the two-phase system used for comparison with psychrometric data. Other governing parameters are: $d_0 = 2.5 \cdot 10^{-4}$ m, $T^{sat} = 373.15$ K, $h_{lv} = 2.33 \cdot 10^6$ J/kg, $\sigma = 0.072$ N/m and $D_{lg} = 2.23 \cdot 10^{-5}$ m²/s.

Phase	ρ [kg/m ³]	μ [kg/(m s)]	c_p [J/(kg K)]	k [W/(m K)]	M [kg/kmol]
Gas	1.2	$1.79 \cdot 10^{-5}$	1006	0.026	29.0
Liquid	10	$1.14 \cdot 10^{-3}$	4186	0.006	18.0

is slightly larger when the vapor mass diffusion is discretized explicitly. Yet, both approaches show an error convergence between 1.75 and 2.0 as the grid is refined.

4.3. Fully-coupled system – reproduction of psychrometric data

Next we consider the solution of the fully coupled system in the same configuration of the previous sections. In this case, evaporation is driven by the partial pressure of the vaporized liquid near the interface, which is smaller than the corresponding saturated value at the interface $p_{2,\Gamma}^{l,sat}$. We recall that $p_{2,\Gamma}^{l,sat}$ is related to the interface temperature through the Clausius-Clapeyron relation (eq. (7)).

We consider a stationary two-dimensional circular droplet with the same geometry and outflow conditions of the previous cases. The energy equation is solved with Dirichlet temperature boundary conditions T_∞ at all boundaries. As in the previous case, the steady state solution of eq. (6) with $\mathbf{u} = 0$ is used as initial condition for Y_2^l , with Dirichlet boundary conditions $Y_{2,\infty}^l$ at the domain boundaries, and now with prescribed interfacial value $Y_{2,\Gamma}^l$ computed from eqs. (7) and (8).

Similarly to the recent work in [12], we use psychrometric data to validate our numerical method. A water droplet is immersed in air with relative humidity ψ and so-called dry bulb temperature, T_{db} , equal to the initial droplet temperature. As evaporation is triggered, the droplet is cooled and its temperature decreases. This evaporative cooling is counterbalanced by a conductive heat flux from the warmer air to the droplet. Eventually, a uniform equilibrium temperature is reached inside the droplet – so-called wet-bulb temperature, T_{wb} . Accordingly, in our computational domain the temperature boundary condition is set to the desired dry bulb temperature $T_\infty = T_{db}$, and the corresponding mass fraction prescribed at the boundary, $Y_{2,\infty}^l$; this is computed from the desired air relative humidity ψ at the dry bulb temperature T_{db} (cf. eq. 7):

$$Y_{2,\infty}^l = \frac{\psi p_{2,\infty}^{l,sat} M_l}{(p_t - \psi p_{2,\infty}^{l,sat}) M_g + \psi p_{2,\infty}^{l,sat} M_l} \quad (41)$$

with $p_{2,\infty}^{l,sat}$ computed from eq. (8) evaluated at T_{db} .

For simplicity, in addition to the non-dimensional governing parameters in Table 1, we report the system properties in Table 2 and its caption. As in [12], the liquid density and thermal conductivity are set smaller than those of water, while keeping the thermal diffusivity constant. The reasons behind this choice are twofold. First, the present methodology becomes less accurate as the density ratios increases (as for most of interface-capturing methods) and using $\lambda_\rho = 1000$ would require to greatly reduce the time-step. Second, the interfacial mass flux decreases almost linearly when increasing the density ratios and therefore, the time to reach the wet-bulb temperature in this configuration where convection effects are limited would be unnecessarily long to test the method.

Fig. 8(a) illustrates the time evolution of the droplet temperature profile, for $T_{db} = 313$ K and $\psi = 50\%$. Indeed, an equilibrium droplet (wet bulb) temperature T_{wb} is attained after a transient due to the mechanisms described above. The velocity

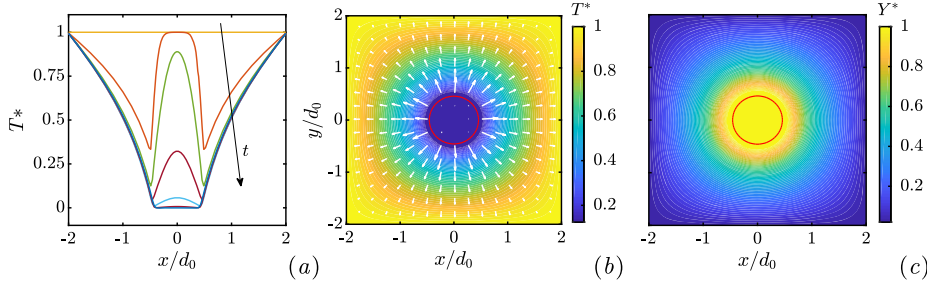


Fig. 8. Evaporating static droplet with $T_{db} = 313$ K and $\psi = 50\%$. (a): time evolution of the normalized temperature profile $T^* = (T - T_{wb,0})/(T_\infty - T_{wb,0})$ along $x = 0$ at $tD_{lg}/d_0^2 = \{0, 0.95, 3.80, 11.4, 22.8, 34.2, 45.6\}$. (b,c): velocity field with contours of T^* (b), and contours of normalized vapor mass fraction $Y^* = (Y_2^l - Y_{2,\infty}^l)/(Y_{2,\Gamma}^l - Y_{2,\infty}^l)$ (c) at $t = 45.6 d_0^2/D_{lg}$.

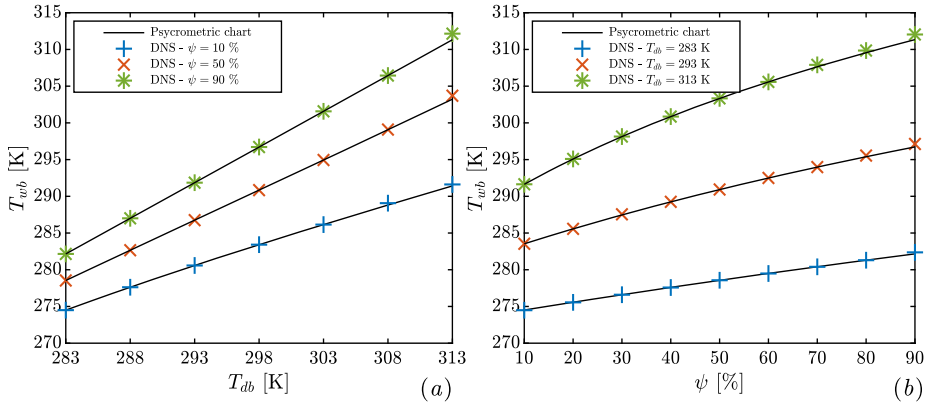


Fig. 9. Numerical results of wet bulb temperature T_{wb} of an evaporating static droplet, compared to the psychrometric chart values varying (a) dry bulb temperature T_{db} and (b) relative humidities ψ , on a 256×256 grid.

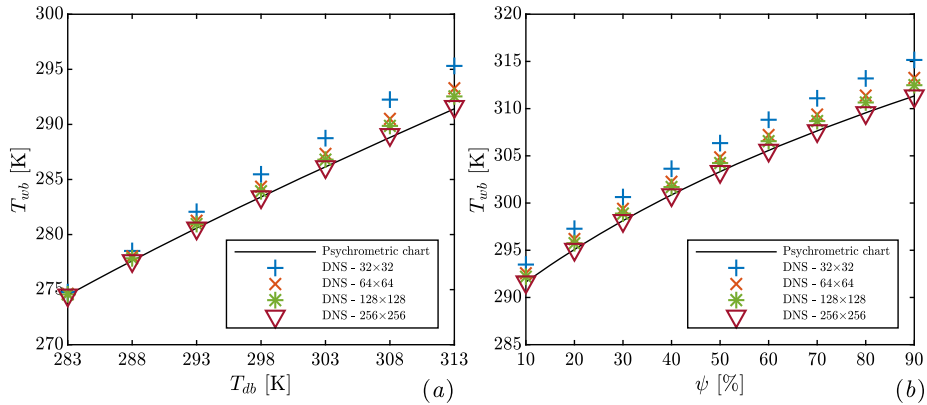


Fig. 10. Grid convergence study for the wet bulb temperature T_{wb} of an evaporating static droplet: (a) fixed relative humidity $\psi = 10\%$, and (b) fixed dry bulb temperature $T_{db} = 283$ K.

vector field, temperature and mass fraction at this equilibrium condition are shown in panels (b,c) of the same figure. We performed simulations for several combinations of T_{db} and ψ , and compared the resulting equilibrium droplet temperature to the expected value of the wet-bulb temperature T_{wb} from psychrometric data, see e.g. [63]. Fig. 9 displays the numerical results of wet bulb temperature to psychrometric data for several combinations of ψ and T_{db} , on the finest grid considered (256×256). The agreement is very good, with slight deviations for the highest temperature and relative humidity considered. A similar trend has been reported in [12], where the discrepancies have been attributed to the assumption of constant thermophysical properties, which actually vary in a not negligible way at high values of T_{db} and ψ .

In Fig. 10 we present the grid convergence of the solution pertaining the target wet bulb temperature T_{wb} for varying T_{db} (panel a) and ψ (panel b). Despite the larger grid-sensitivity of the results at higher temperatures for coarser grids, all cases converge towards the expected value.

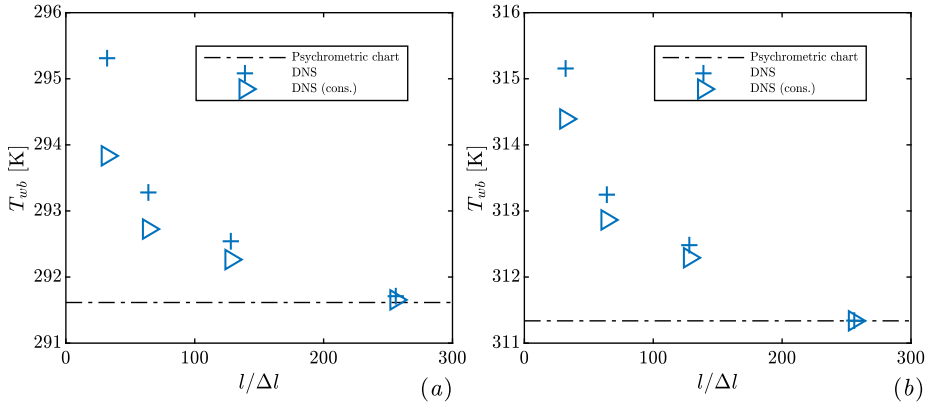


Fig. 11. Comparison between the non-conservative and the conservative discretization (the latter labeled with *cons.* in the figure) of the energy equation for the numerical calculation of the wet-bulb temperature: (a) dry-bulb temperature and relative humidity $T_{db} = 313$ K, $\psi = 10\%$ and, (b) dry-bulb temperature and relative humidity $T_{db} = 283$ K, $\psi = 90\%$.

Finally, we further investigate the large deviations that occur at higher T_{db} and ψ by performing a comparison between the fully conservative discretization and the non-conservative discretization of the energy equation. As Fig. 11 shows, when a conservative discretization is employed the error is clearly reduced at lower resolution, whereas the difference is less appreciable when a highly refined grid is employed.

4.4. Sedimenting droplet in a confining container

This configuration illustrates the ability of the method to handle droplet evaporation under pronounced deformations, with close interaction with solid boundaries and large temperature gradients. We consider an evaporating droplet with initial diameter d_0 flowing down a confining container with hot walls under the action of gravity, acting in the negative y direction. The droplet is initially at rest in a domain with dimensions $[0, 2.5d_0] \times [0, 10d_0]$, in an off-centered position at the top of the container $\mathbf{x}_c = [0.25, 0.9]d_0$. No-slip and no-penetration (wall) boundary conditions are prescribed at all boundaries, except for the top side, where a zero-pressure outflow is prescribed. Dirichlet boundary conditions for temperature (T_w) and mass fraction ($Y_{2,w} = 0$) are prescribed at the walls, and (zero) Neumann at the outflow. The initial temperature field is uniform with temperature $T_0 = 0.95 T^{sat} < T_w$, whereas the mass fraction field is initialized as described in the previous section. The relevant flow parameters are reported in Table 1. Three cases are considered: a droplet with (1) mild and (2) high evaporation rates, and (3) a reference case without phase change. These different evaporation rates are achieved by varying the wall temperature while keeping the other parameters constant, namely $T_w = 1.15 T^{sat}$ and $T_w = 2.54 T^{sat}$ for the two phase-changing cases. To better qualify the evaporation regime, we define a wall Stefan number based on the difference between the imposed wall temperature and the initial temperature, i.e., $Ste_w \equiv c_{p,2}(T_w - T_0)/h_{lv}$.

Fig. 12 depicts the time history of the droplet sedimentation for the three cases under consideration. All cases show an oscillatory trajectory, due to droplet-wall interactions in this confined geometry and to the droplet inertia. As the evaporation rate (i.e. Ste_w) increases, the frequency of the droplet oscillation grows, due to the wall-repelling force caused by the Stefan flow. This is particularly evident for the case with $Ste_w = 0.26$, where the droplet starts moving with an almost horizontal velocity, and remains in a levitating state at late times. The droplet dynamics can be better understood by inspecting the temperature and velocity fields. These are shown in Fig. 13 for the case with $Ste_w = 0.26$ at three different instants. Since $T_w > T_0$, the temperature at the side of the droplet closest to the wall is larger than that on the other side at early times. The Stefan flow is therefore larger on the warmer side of the droplet, which generates a strong wall-repelling force. After some time, heat conduction reduces the temperature gradients in the gaseous phase, and the amplitude of the droplet oscillation reduces. Finally, at later times, the droplet reaches the bottom side of the container with significant mass loss, and the Stefan flow is sufficiently high to sustain the droplet weight. The droplet therefore remains in a levitating, Leidenfrost-like state in the middle of the container until its complete evaporation.

The solution grid convergence for this more complex case is illustrated for the setup with highest mass transfer, $Ste_w = 0.26$, confirming that 128×512 grid points (about 50 grid points over the initial droplet diameter) suffice for accurately resolving this problem (see Fig. 14).

4.5. Coalescence of two droplets

In this section, we consider the case of two two-dimensional circular droplets interacting and eventually coalescing while evaporating in a hot gas. This test shows the ability of the proposed methodology to handle a complex topological change (i.e., merging), providing stable and convergent results. The two droplets with initial diameter d_0 move towards each other with initial velocity $u_{r,0}/2$ in a domain with dimensions $[0, 10d_0] \times [0, 5d_0]$, initially, at position $[4, 2.5]d_0$ and $[6, 2.5]d_0$.

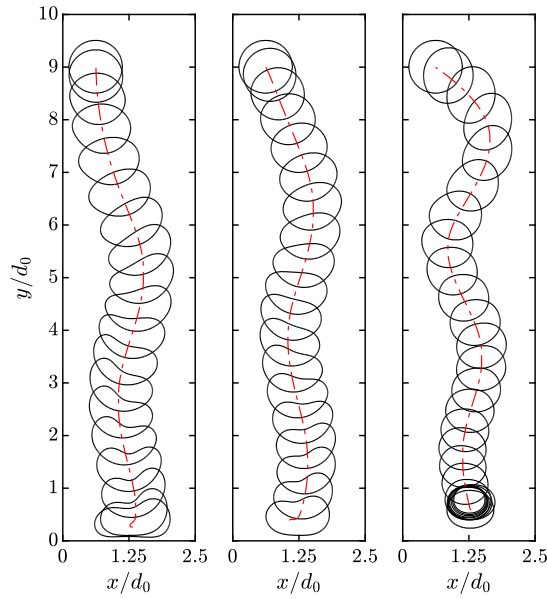


Fig. 12. Trajectory of the droplet's center of mass (red dashed-dotted line) of a sedimenting droplet in a confining container on a 128×512 grid for (from left to right) $Ste_w = 0$ (i.e. no phase change), $Ste_w = 0.033$ and $Ste_w = 0.26$. The interface is shown by the solid black lines with a time interval equal to $0.67\sqrt{d_0/|g|}$.

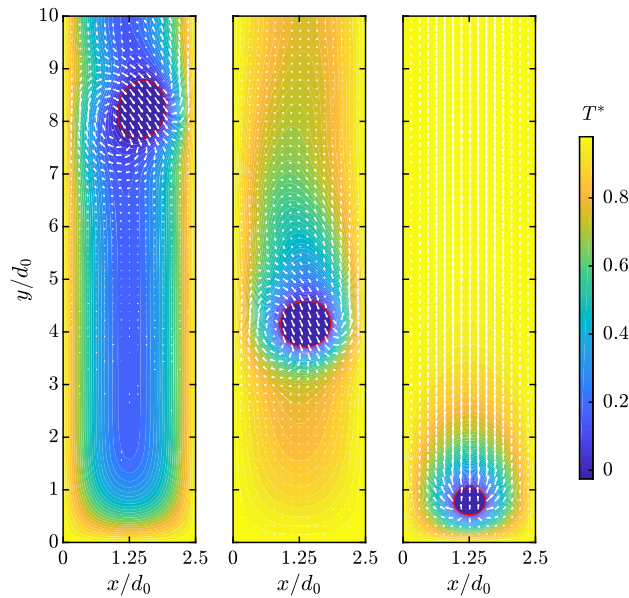


Fig. 13. Evaporation of a sedimenting droplet in a confining container at $Ste_w = 0.26$ on a 128×512 grid. Contour plots of normalized temperature ($T^* = (T - T_0)/(T_w - T_0)$) at three time-instants (from left to right): $t\sqrt{|g|}/d_0 = 1.70, 6.75$ and 26.9 . The interface location is depicted by the solid red line.

The relative velocity is chosen to give an impact Reynolds number $Re_{d_0} = \rho_2 u_{r,0} d_0 / \mu_2 = 25$. The initial temperature field is equal to $T_{g,0} = 1.88 T^{sat}$ in the gas phase, $T_{l,0} = 0.95 T^{sat}$ in the liquid. The mass fraction field is initialized with the steady state solution of eq. (6), as done for the previous cases. Zero-pressure outflow boundary conditions are prescribed for velocity and pressure, whereas a Dirichlet boundary condition is prescribed for temperature (equal to $T_{g,0}$) and vapor mass fraction (equal to $Y_{2,\infty}^l = 0$). The remaining governing parameters have been reported in Table 1, with $C_{\Delta t} = 0.1$ for a stable numerical integration.

Before presenting the results, it is worth recalling that all interface-capturing methods based on the fully Eulerian description of the interface are able to handle automatically topological changes such as coalescence, merging and break-up. This ability comes at the price of having little or no control on the so-called numerical coalescence which occurs when two

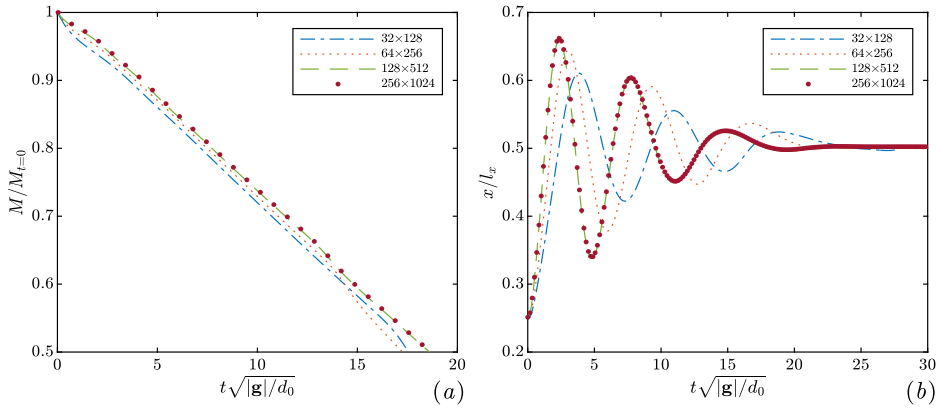


Fig. 14. Grid convergence test for the sedimenting droplet in a confining container with $Ste_w = 0.26$, for the droplet mass (a), and x -component of the trajectory (b), showing oscillations along the container centerline.

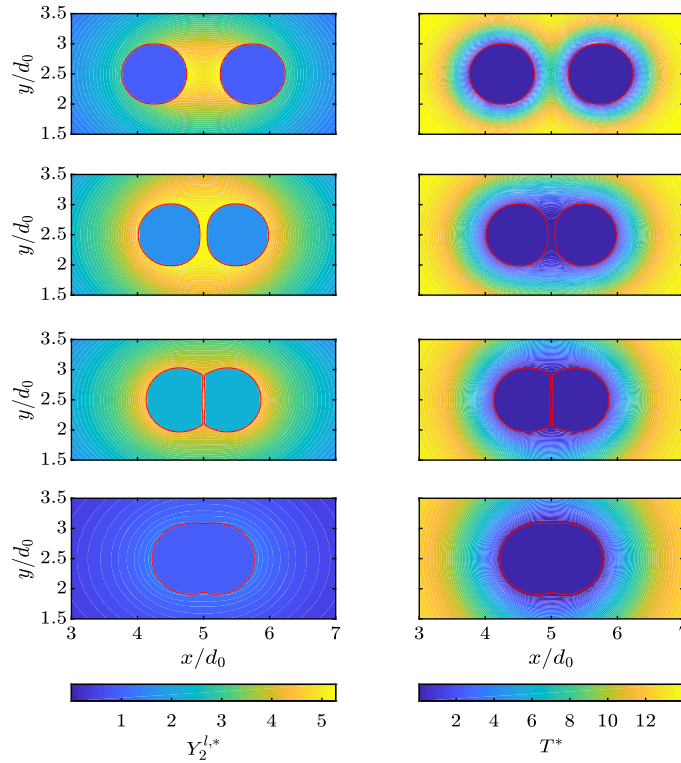


Fig. 15. Contours of the vaporised liquid mass fraction $Y_2^{l,*} = (Y_2^l - Y_{2,\infty}^l)/(Y_{2,\Gamma}^l - Y_{2,\infty}^l)$ (on the left) and of the dimensionless temperature $T^* = (T - T_\infty)/(T_{wb,f} - T_\infty)$ at time $tu_{r,0}/d_0 = 0.78, 1.75, 3.5$ and 7.0 from top to bottom. $T_{wb,f}$ represents the wet bulb temperature of the droplet (uniform from $t^* > 1$) and $Y_{2,\Gamma}^l$ the interfacial saturation value evaluated at $T_{wb,f}$. The interface is depicted by the iso-contour $C = 0.5$ from the simulation with resolution 512×256 . The data are shown in the sub-domain $[3, 7]d_0 \times [1.5, 3.5]d_0$.

interfaces are less than one grid cell apart. This is known to over-predict the coalescence rate and can be counteracted by using short-range forces [47], or consider multiple interface fronts as in [64]. Since the influence of the coalescence and merging rate for evaporating droplets is out of the scope of the present work, in the following, we limit to show that the proposed methodology is capable of handling topology changes, providing grid-converging results.

The first instances of the simulations are characterized by the highest evaporation rate. As shown in Fig. 15, the highest gradients of the mass fraction occur at the early stages when the two droplets are separated and move in opposite direction. Furthermore, the evaporation rate is enhanced by the high temperature gradients between the gas phase and the droplet interface, as it emerges from the time history of the normalized total liquid mass in the system depicted in Fig. 16. As the droplets approach and eventually merge, the evaporation rate decreases because the mass transfer of the liquid in the inert gas decreases the temperature around the droplet and thus reduces the saturation value of the mass fraction at the

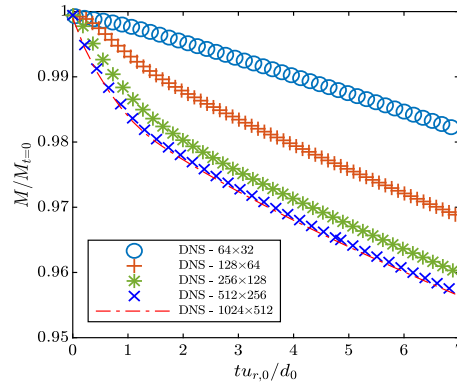


Fig. 16. Temporal evolution of the normalized total liquid mass in the system, $M/M_{t=0}$, as a function of the dimensionless time $t u_{r,0}/d_0$ for simulations on different grids (see the figure legend).

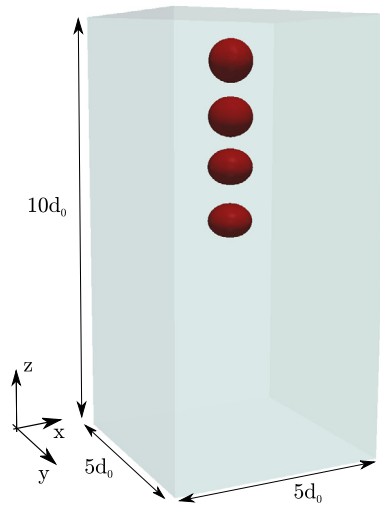


Fig. 17. Illustration of the three-dimensional domain with a sedimenting evaporating droplet at four physical times $t\sqrt{|g|/d_0} = 0, 0.96, 1.91$ and 2.87 . The droplet surface is depicted as the locus of points where $C = 0.5$.

interface. In addition, the droplet velocity decreases and the evaporation process becomes diffusion dominated. Finally, we note that the simulation is grid convergent, with differences in the history of total liquid mass of less than 1% on the two finest grids (50 and 100 grid points over the droplet diameter).

4.6. Droplet settling in a three-dimensional periodic domain

Finally, we perform a three-dimensional simulation for a fully coupled case, where a single droplet moves, deforms and evaporates in a non-isothermal environment. In this configuration, depicted in Fig. 17, a spherical droplet with initial diameter d_0 settles in a domain with dimensions $[0, 5d_0] \times [0, 5d_0] \times [0, 10d_0]$, under the effect of gravity acting in the negative z direction. The droplet is initially at rest and centered at the top of the domain $\mathbf{x}_c = [2.5, 2.5, 9]d_0$. The domain is periodic in all directions, discretized on a $320 \times 320 \times 640$ grid. The net weight of the system is subtracted to the fluid momentum balance at each time step to yield zero net acceleration, thereby avoiding a constant acceleration of the entire system [6]. The initial temperature field is uniform and given by $T_0 = 0.95 T^{sat}$, while the mass fraction field is initialized with the steady state solution of eq. (6), as done for the previous cases. The remaining governing parameters have been reported in Table 1 and we set $C_{\Delta t} = 0.1$.

Fig. 18 presents planar contours of vapor mass fraction for different time instants. From the two-dimensional distributions of vapor mass fraction in Fig. 18 we qualitatively observe that initially the evaporation process is dominated by diffusion, since the falling velocity is still limited (left panel in Fig. 18). As time evolves, the droplet accelerates, deforms and starts to lose mass at a faster rate (see the last two panels of Fig. 18, where the formation of a wake is clear). At this stage, both the increased surface area due to the droplet deformation, and the increasing convective effects tend to increase the mass transfer rate, with the latter effect expected to be more significant. This change in the behavior of the droplet evaporation occurs at $t\sqrt{|g|/d_0} \approx 2$ as it is clear from the time history of the droplet mass depicted in Fig. 19.

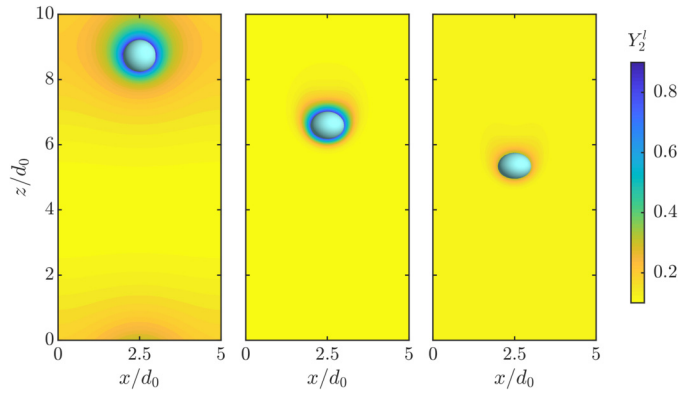


Fig. 18. Contours of vaporised liquid mass fraction Y_2^l in the $y = 2.5 d_0$ plane, for time instants $t\sqrt{g/d_0} = 0.06, 1.91$ and 2.87 (from left to right). The interface is depicted through iso-contours of $C = 0.5$.

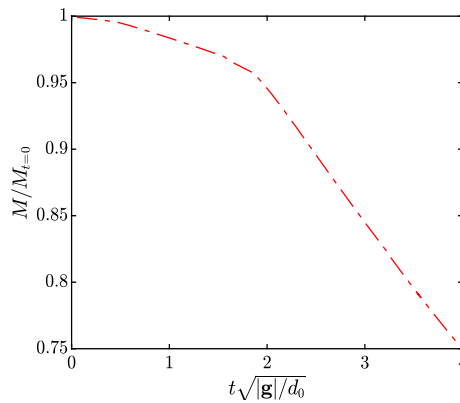


Fig. 19. Temporal evolution of the normalized droplet mass, $M/M_{t=0}$, as a function of the dimensionless time $t\sqrt{g/d_0}$.

5. Conclusions

We have presented a numerical method for interface-resolved simulations of phase changing two-fluid flows using a volume-of-fluid method. The solver is based on an algebraic MTHINC VoF, implemented in an efficient, FFT-based two-fluid finite-difference Navier-Stokes solver. To circumvent the issues related to the jump in fluid velocity across the interface, we transport the VoF function using an interface velocity composed of two parts: (1) a divergence-free extension of the liquid velocity in the entire domain, and (2) a irrotational term accounting for the phase change. This approach requires a simple extension of a standard split advection method with a volume deflation step, and, unlike other approaches in the literature, can be easily applied to other algebraic or geometrical VoF methods. Furthermore, it can be used in phase-change solvers based on the level-set and conservative level-set method when maintaining the whole domain formulation in solving the Navier-Stokes equations. The divergence-free velocity extension is computed with the aid of a direct fast Poisson solver with negligible computational overhead, about 2% of the total cost.

Evaporation is handled by reconstructing a level-set field from the VoF function, which allows us to benefit from well-established level-set methods for solving phase-change problems; see e.g. [20]. The equation of transport of vaporized liquid mass is solved in the gaseous domain, with a Dirichlet boundary condition at the interface, computed from the thermodynamic equilibrium defined by the Clausius-Clapeyron relation. This term can be discretized in time implicitly as in [54], or explicitly. A second-order pde-based extrapolation technique allow us to define the interphase mass flux in a band around the interface, i.e. where the VoF function $0 < C < 1$. As a consequence, the continuity, momentum and energy equations can be solved with a so-called whole-domain formulation, where source terms accounting for the velocity, stress and heat flux jumps along the interface-normal direction are easily incorporated with a CSF-like approach.

The numerical method has been extensively verified and validated against different benchmark cases of increasing complexity. The results illustrate the excellent mass-preserving nature of the method, and the ability of the overall approach to reproduce psychrometric data. Moreover, we show that the method can handle large deformations for a droplet evaporating in the presence of walls, and demonstrate its potential for three-dimensional simulations of evaporating flows. Further, as shown in the Appendix A, a direct extension of the method can be used to simulate boiling flows. Note finally that the specific implementation presented here might be improved with a discretization of the energy equation in conservative

form, and a momentum-preserving method as in [65] to more efficiently solve problems at high density ratios. Overall, we believe that our method has the right ingredients to serve as a base for massive, high-fidelity simulations of phase-changing turbulent flows.

Declaration of competing interest

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

Acknowledgements

The work is supported by INTERFACE, under the project *Hybrid multiscale modelling of transport phenomena for energy efficient processes*, financed by the Swedish Research Council (VR), and by the European Research Council grant, no. ERC-2013-CoG-616186, TRITOS. The computer time was provided by SNIC (Swedish National Infrastructure for Computing) and by the National Infrastructure for High Performance Computing and Data Storage in Norway (project no. NN9561K). Stéphane Zaleski is acknowledged for the useful discussions, and for pointing out the PhD thesis in [61], which reports a similar approach for constructing a divergence-free velocity extension in the context of boiling flows.

Appendix A. Application of the method for boiling simulations

Here we briefly explain how to extend the numerical framework described for evaporation to study temperature-induced phase change (i.e., boiling), which occurs between a liquid phase and its vapor. From a physical point of view, boiling starts when the partial pressure of liquid in the gaseous phase is equal to the pressure p_t that the surrounding environment exerts on the liquid itself. By fixing p_t and postulating thermodynamic equilibrium at the interface [1], the Clausius-Clapeyron relation indicates that the interfacial temperature T_Γ is constant and equal to the saturation temperature T^{sat} , evaluated at p_t . Under this assumption, together with the incompressibility constrain on both phases and weak viscous dissipation, the governing equations reduce to the following form, see e.g. [1,9],

$$\nabla \cdot \mathbf{u} = \dot{m} \left(\frac{1}{\rho_2} - \frac{1}{\rho_1} \right) \delta_\Gamma, \quad (\text{A.1a})$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot \left(\mu \left(\nabla \mathbf{u} + \nabla \mathbf{u}^T \right) \right) + \rho \mathbf{g} + \sigma \kappa \delta_\Gamma \mathbf{n}, \quad (\text{A.1b})$$

$$\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = \nabla \cdot (k \nabla T), \quad (\text{A.1c})$$

$$\dot{m} h_{lv} = k_2 \nabla_\Gamma T_2 \cdot \mathbf{n} - k_1 \nabla_\Gamma T_1 \cdot \mathbf{n}; \quad (\text{A.1d})$$

First, note that the continuity (A.1a) and momentum (A.1b) equations are independent of the phase change mechanism. Therefore, their numerical solution follows the procedure reported in section 3.2. Moreover, employing a VoF method to capture the interface dynamics, the interface representation (section 3.1) and the interface velocity construction (section 3.6) remain formally unchanged, with the minor modification of constructing the interface velocity from a divergence-free extension of the vapor velocity, instead of that of the liquid.

To solve the boiling problem, the main difference is that the energy eq. (A.1c) is solved with a Dirichlet boundary condition at the interface: $T_\Gamma = T^{sat}$. In practice, we solve eq. (A.1c) with the same schemes used for Y_2^l , but for a temperature field T_1 in Ω_1 , and T_2 in Ω_2 , separately, with $T_\Gamma = T^{sat}$ imposed at the interface. The energy equation is therefore solved in each subdomain with constant thermophysical properties (i.e. $\rho = \rho_i$, $c_p = c_{pi}$ and $k = k_i$ in Ω_i ($i = 1, 2$)). Then, following the procedure reported in [18], the temperature field in the liquid domain T_1 is extrapolated into the vapor domain Ω_2 , and the temperature field in the vapor domain T_2 is extrapolated into the liquid domain Ω_1 using the procedure described in section 3.4. The resulting extended fields T_1^e and T_2^e are continuously differentiable across Γ . Accordingly, they can be used to compute \dot{m} in eq. (A.1d) using a central difference scheme. This allows to define \dot{m} in a band around the interface, covering the region where the VoF function is $0 < C < 1$.

References

- [1] M. Ishii, T. Hibiki, *Thermo-Fluid Dynamics of Two-Phase Flow*, Springer Science & Business Media, 2010.
- [2] C.R. Kharangate, I. Mudawar, Review of computational studies on boiling and condensation, *Int. J. Heat Mass Transf.* 108 (2017) 1164–1196.
- [3] S. Elghobashi, Direct numerical simulation of turbulent flows laden with droplets or bubbles, *Annu. Rev. Fluid Mech.* 51 (2019) 217–244.
- [4] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, *J. Comput. Phys.* 100 (1) (1992) 25–37.
- [5] S. Osher, R. Fedkiw, *Level Set Methods and Dynamic Implicit Surfaces*, vol. 153, Springer Science & Business Media, 2006.
- [6] G. Tryggvason, R. Scardovelli, S. Zaleski, *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*, Cambridge University Press, 2011.
- [7] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, *J. Comput. Phys.* 169 (2) (2001) 556–593.
- [8] D.M. Anderson, G.B. McFadden, A.A. Wheeler, Diffuse-interface methods in fluid mechanics, *Annu. Rev. Fluid Mech.* 30 (1) (1998) 139–165.

- [9] D. Juric, G. Tryggvason, Computations of boiling flows, *Int. J. Multiph. Flow* 24 (3) (1998) 387–410.
- [10] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part I: numerical method, *Int. J. Heat Mass Transf.* 47 (25) (2004) 5451–5461.
- [11] A. Esmaeeli, G. Tryggvason, Computations of film boiling. Part II: multi-mode film boiling, *Int. J. Heat Mass Transf.* 47 (25) (2004) 5463–5476.
- [12] M. Irfan, M. Muradoglu, A front tracking method for direct numerical simulation of evaporation process in a multiphase system, *J. Comput. Phys.* 337 (2017) 132–153.
- [13] M. Irfan, M. Muradoglu, A front tracking method for particle-resolved simulation of evaporation and combustion of a fuel droplet, *Comput. Fluids* 174 (2018) 283–299.
- [14] M.N. Farooqi, D. Izbassarov, M. Muradoglu, D. Unat, Communication analysis and optimization of 3d front tracking method for multiphase flow simulations, *Int. J. High Perform. Comput. Appl.* 33 (1) (2019) 67–80.
- [15] Y. Ling, D. Fuster, G. Tryggvason, S. Zaleski, A two-phase mixing layer between parallel gas and liquid streams: multiphase turbulence statistics and influence of interfacial instability, *J. Fluid Mech.* 859 (2019) 268–307.
- [16] G. Son, V.K. Dhir, Numerical simulation of film boiling near critical pressures with a level set method, *J. Heat Transf.* 120 (1) (1998) 183–192.
- [17] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, *J. Comput. Phys.* 222 (2) (2007) 536–555.
- [18] 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.
- [19] M.S. Lee, A. Riaz, V. Aute, Direct numerical simulation of incompressible multiphase flow with phase change, *J. Comput. Phys.* 344 (2017) 381–418.
- [20] S. Tanguy, T. Ménard, A. Berlemont, A level set method for vaporizing two-phase flows, *J. Comput. Phys.* 221 (2) (2007) 837–853.
- [21] L.R. Villegas, R. Alis, M. Lepilliez, S. Tanguy, A ghost fluid/level set method for boiling flows and liquid evaporation: application to the Leidenfrost effect, *J. Comput. Phys.* 316 (2016) 789–813.
- [22] K. Luo, C. Shao, Y. Yang, J. Fan, A mass conserving level set method for detailed numerical simulation of liquid atomization, *J. Comput. Phys.* 298 (2015) 495–519.
- [23] Z. Ge, J.-C. Loiseau, O. Tammisola, L. Brandt, An efficient mass-preserving interface-correction level set/ghost fluid method for droplet suspensions under depletion forces, *J. Comput. Phys.* 353 (2018) 435–459.
- [24] S.W. Welch, J. Wilson, A volume of fluid based method for fluid flows with phase change, *J. Comput. Phys.* 160 (2) (2000) 662–682.
- [25] D. Agarwal, S. Welch, G. Biswas, F. Durst, Planar simulation of bubble growth in film boiling in near-critical water using a variant of the vof method, *J. Heat Transf.* 126 (3) (2004) 329–338.
- [26] M. Yuan, Y. Yang, T. Li, Z. Hu, Numerical simulation of film boiling on a sphere with a volume of fluid interface tracking method, *Int. J. Heat Mass Transf.* 51 (7–8) (2008) 1646–1657.
- [27] J. Schlottke, B. Weigand, Direct numerical simulation of evaporating droplets, *J. Comput. Phys.* 227 (10) (2008) 5215–5237.
- [28] 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.
- [29] G. Tomar, G. Biswas, A. Sharma, A. Agrawal, Numerical simulation of bubble growth in film boiling using a coupled level-set and volume-of-fluid method, *Phys. Fluids* 17 (11) (2005) 112103.
- [30] Y. Sato, B. Ničeno, A sharp-interface phase change model for a mass-conservative interface tracking method, *J. Comput. Phys.* 249 (2013) 127–161.
- [31] O. Ubbink, R. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, *J. Comput. Phys.* 153 (1) (1999) 26–50.
- [32] D.A. Hoang, V. van Steijn, L.M. Portela, M.T. Kreutzer, C.R. Kleijn, Benchmark numerical simulations of segmented two-phase flows in microchannels using the volume of fluid method, *Comput. Fluids* 86 (2013) 28–36.
- [33] A. Prosperetti, G. Tryggvason, *Computational Methods for Multiphase Flow*, Cambridge University Press, 2009.
- [34] T. Bergman, A. Lavine, F. Incropera, D. Dewitt, *Fundamentals of heat and mass transfer*, Instructor 201609.
- [35] S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi, Y. Matsumoto, F. Xiao, An interface capturing method with a continuous function: the THINC method with multi-dimensional reconstruction, *J. Comput. Phys.* 231 (5) (2012) 2328–2358.
- [36] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, *J. Comput. Phys.* 130 (2) (1997) 269–282.
- [37] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, A geometrical area-preserving volume-of-fluid advection method, *J. Comput. Phys.* 192 (1) (2003) 355–364.
- [38] G.D. Weymouth, D.K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on cartesian-grids, *J. Comput. Phys.* 229 (8) (2010) 2853–2865.
- [39] A.J. Chorin, Numerical solution of the Navier–Stokes equations, *Math. Comput.* 22 (104) (1968) 745–762.
- [40] S. Dong, J. Shen, A time-stepping scheme involving constant coefficient matrices for phase-field simulations of two-phase incompressible flows with large density ratios, *J. Comput. Phys.* 231 (17) (2012) 5788–5804.
- [41] M.S. Dodd, A. Ferrante, A fast pressure-correction method for incompressible two-fluid flows, *J. Comput. Phys.* 273 (2014) 416–434.
- [42] D.L. Youngs, An interface tracking method for a 3d eulerian hydrodynamics code, Atomic Weapons Research Establishment (AWRE) Technical Report 44 (92) (1984) 35.
- [43] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, *J. Comput. Phys.* 100 (2) (1992) 335–354.
- [44] B.P. Leonard, A stable and accurate convective modelling procedure based on quadratic upstream interpolation, *Comput. Methods Appl. Mech. Eng.* 19 (1) (1979) 59–98.
- [45] P. Costa, A fft-based finite-difference solver for massively-parallel direct numerical simulations of turbulent flows, *Comput. Math. Appl.* 76 (8) (2018) 1853–1862.
- [46] M.E. Rosti, F. De Vita, L. Brandt, Numerical simulations of emulsions in shear flows, *Acta Mech.* 230 (2) (2019) 667–682.
- [47] F. De Vita, M.E. Rosti, S. Caserta, L. Brandt, On the effect of coalescence on the rheology of emulsions, *J. Fluid Mech.* 880 (2019) 969–991.
- [48] M.E. Rosti, Z. Ge, S.S. Jain, M.S. Dodd, L. Brandt, Droplets in homogeneous shear turbulence, *J. Fluid Mech.* 876 (2019) 962–984, <https://doi.org/10.1017/jfm.2019.581>.
- [49] G. Russo, P. Smereka, A remark on computing distance functions, *J. Comput. Phys.* 163 (1) (2000) 51–67.
- [50] A. Albadawi, D. Donoghue, A. Robinson, D. Murray, Y. Delauré, Influence of surface tension implementation in volume of fluid and coupled volume of fluid with level set methods for bubble growth and detachment, *Int. J. Multiph. Flow* 53 (2013) 11–28.
- [51] X.-D. Liu, R.P. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, *J. Comput. Phys.* 160 (1) (2000) 151–178.
- [52] B. Fornberg, Generation of finite difference formulas on arbitrarily spaced grids, *Math. Comput.* 51 (184) (1988) 699–706.
- [53] M. Dodd, Direct Numerical Simulation of Droplet-Laden Isotropic Turbulence, 2017.
- [54] F. Gibou, R.P. Fedkiw, L.-T. Cheng, M. Kang, A second-order-accurate symmetric discretization of the Poisson equation on irregular domains, *J. Comput. Phys.* 176 (1) (2002) 205–227.
- [55] R.D. Falgout, J.E. Jones, U.M. Yang, The design and implementation of hypre, a library of parallel high performance preconditioners, in: *Numerical Solution of Partial Differential Equations on Parallel Computers*, Springer, 2006, pp. 267–294.

- [56] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, *J. Comput. Phys.* 193 (1) (2004) 349–355.
- [57] M. Sussman, E. Fatemi, P. Smereka, S. Osher, An improved level set method for incompressible two-phase flows, *Comput. Fluids* 27 (5–6) (1998) 663–680.
- [58] 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.
- [59] M. Chai, K. Luo, C. Shao, H. Wang, J. Fan, A coupled vaporization model based on temperature/species gradients for detailed numerical simulations using conservative level set method, *Int. J. Heat Mass Transf.* 127 (2018) 743–760.
- [60] M. Castro, B. Costa, W.S. Don, High order weighted essentially non-oscillatory weno-z schemes for hyperbolic conservation laws, *J. Comput. Phys.* 230 (5) (2011) 1766–1792.
- [61] L. Malan, Direct numerical simulation of free-surface and interfacial flow using the vof method: cavitating bubble clouds and phase change, Ph.D. thesis, University of Cape Town, 2018.
- [62] M. Kang, R.P. Fedkiw, X.-D. Liu, A boundary condition capturing method for multiphase incompressible flow, *J. Sci. Comput.* 15 (3) (2000) 323–360.
- [63] S. Herrmann, H.-J. Kretzschmar, D.P. Gatley, Thermodynamic properties of real moist air, dry air, steam, water, and ice (rp-1485), *HVAC & R Res.* 15 (5) (2009) 961–986.
- [64] M. Kwakkel, W.-P. Breugem, B.J. Boersma, Extension of a clsvof method for droplet-laden flows with a coalescence/breakup model, *J. Comput. Phys.* 253 (2013) 166–188.
- [65] D. Fuster, T. Arrufat, M. Cialesi-Esposito, Y. Ling, L. Malan, S. Pal, R. Scardovelli, G. Tryggvason, S. Zaleski, A momentum-conserving, consistent, volume-of-fluid method for incompressible flow on staggered grids, arXiv preprint arXiv:1811.12327.