



HAL
open science

A geometric VOF method for interface resolved phase change and conservative thermal energy advection

L.C. Malan, A.G. Malan, Stéphane Zaleski, P.G. Rousseau

► **To cite this version:**

L.C. Malan, A.G. Malan, Stéphane Zaleski, P.G. Rousseau. A geometric VOF method for interface resolved phase change and conservative thermal energy advection. *Journal of Computational Physics*, 2021, 426, pp.109920. 10.1016/j.jcp.2020.109920 . hal-03654968

HAL Id: hal-03654968

<https://hal.sorbonne-universite.fr/hal-03654968v1>

Submitted on 29 Apr 2022

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

A Geometric VOF Method for Interface Resolved Phase Change and Conservative Thermal Energy Advection

L.C. Malan^{a,b,*}, A.G. Malan^b, S. Zaleski^a, P.G. Rousseau^c

^a*Sorbonne Université et CNRS,*

Institut Jean Le Rond d'Alembert, UMR 7190, Paris, France

^b*Industrial CFD Research Group, Dept. Mechanical Engineering, University of Cape Town, Private Bag X3, Rondebosch 7701, South Africa*

^c*ATPRoM Research Group, Dept. Mechanical Engineering, University of Cape Town, Private Bag X3, Rondebosch 7701, South Africa*

Abstract

We present a novel numerical method to solve the incompressible Navier-Stokes equations for two-phase flows with phase change, using a one-fluid approach. Separate phases are tracked using a geometric Volume-Of-Fluid (VOF) method with piecewise linear interface construction (PLIC). Thermal energy advection is treated in conservative form and the geometric calculation of VOF fluxes at computational cell boundaries is used consistently to calculate the fluxes of heat capacity. The phase boundary is treated as sharp (infinitely thin), which leads to a discontinuity in the velocity field across the interface in the presence of phase change. The numerical difficulty of this jump is accommodated with the introduction of a novel two-step VOF advection scheme. The method has been implemented in the open source code PARIS and is validated using well-known test cases. These include an evaporating circular droplet in microgravity (2D), the Stefan problem and a 3D bubble in superheated liquid. The accuracy shown in the results were encouraging. The 2D evaporating droplet showed excellent prediction of the droplet volume evolution as well as preservation of its circular shape. A relative error of less than 1% was achieved for the Stefan problem case, using water properties at atmospheric conditions. For the final radius of the bubble in superheated liquid at a Jacob number of 0.5, a relative error of less than 6% was obtained on the coarsest grid, with less than 1% on the finest.

Keywords: VOF; incompressible; phase change; thermal energy conservation; finite volume

1. Introduction

A popular approach used in the wider problem space of the numerical solution of multiphase-flows is the so-called one-fluid formulation, in which the governing equations (incompressible Navier-Stokes equations for two-phase flow in our case) are applied to both phases [1]. Some phase tracking technique is then employed to calculate the phase-dependent fluid properties. For an overview of different interface tracking techniques and associated methods to account for phase change, the reader can refer to the work of Kharangate & Mudawar [2] and Tryggvason *et al.* [1].

In this work, the focus will be on the Volume-of-Fluid (VOF) method [3]. Even though VOF methods are relatively common for incompressible, isothermal two-phase flows, very few

*Corresponding author

Email address: lcmalan@gmail.com (L.C. Malan)

include temperature driven phase change. Some of the pertinent VOF works dealing with interface resolved phase change problems will be mentioned here as a reference. Welch and Wilson [4] developed a VOF method for phase change on staggered (MAC) grids. The interface is reconstructed using piecewise linear segments, based on the method of Youngs [5]. A time-split geometric advection technique, also by Youngs, is used to advect the interface. The energy conservation equation is treated non-conservatively. Welch and Rachidi [6] expanded this method to study film boiling with conjugate heat transfer. Agarwal *et al.* [7] used the Welch method [4] with the addition of temperature dependence on fluid properties. Schlottke and Weigand used a geometric based method to study evaporating droplets [8]. They developed an intricate, iterative method to deal with the calculation of a volume source due to mass transfer, which they apply only in interface cells (sharply). However, they neglect the velocity disparity between the interface and the liquid phase.

Kunkelmann [9] used an algebraic VOF advection technique in the OpenFOAM framework and implemented a microlayer and contact angle model to simulate nucleate boiling on unstructured meshes. The mass transfer between phases is accounted for using a smearing technique: source terms are redistributed in some finite region around the interface, which smoothes out the sharp discontinuity in velocity at the interface. Guedon [10] also used the OpenFOAM framework for arbitrary meshes, but attempted to apply a sharp interface approach instead of the smearing technique used by Kunkelmann. A ghost-fluid approach was used at the interface to apply the jump conditions. The author, however, reported difficulties from spurious currents due to unbalanced surface tension models. Another difference compared to the method of Kunkelmann, is the non-conservative treatment of energy conservation.

In summary, there remains several non-trivial challenges to the numerical treatment of the discontinuities at the interface in VOF phase change modelling schemes. More specifically, the authors are unaware of a method that combines conservative thermal energy advection with a sharp interface treatment of the velocity discontinuities that arise from phase change. This is particularly pressing due to the large heat capacity ratio (per unit volume) between liquid and gas phases of many substances of practical interest.

In this work we aim to address the above challenges. A novel method is developed which simulates the incompressible flow of pure substances undergoing phase change. A VOF method with piecewise linear interface calculation (PLIC) is employed. The interface is treated as sharp (infinitely thin), separating immiscible fluids. A novel geometric VOF advection method is developed to deal with the discontinuity in the velocity that arises from phase change. It is applied consistently to the calculation of the thermal energy advection term, which is treated in conservative form. The method is designed such that it can be easily incorporated into an existing VOF framework using geometric advection algorithms, such as that of Weymouth and Yue [11].

2. Mathematical Formulation

This work entails modelling the incompressible flow of pure substances undergoing phase change and in this section we describe the integral form of the governing equations as applied in a generic control volume, V . For a detailed derivation, refer to [12]. Fig. 1 shows volume V , which is fixed in space and bounded by surface S of arbitrary shape with outward pointing normal \mathbf{n} . Surface Γ is assumed to have zero thickness, intersects V and represents the phase boundary (interface) between two pure phases. Note that we will not consider molecular diffusion between phases in this work and will name the two phases liquid (ℓ) and gas (g) throughout. The surface Γ has a normal \mathbf{n}_Γ which points from the liquid into the gas phase. We assume

constant material properties (density, viscosity and heat conductivity) in the respective liquid and gas phases.

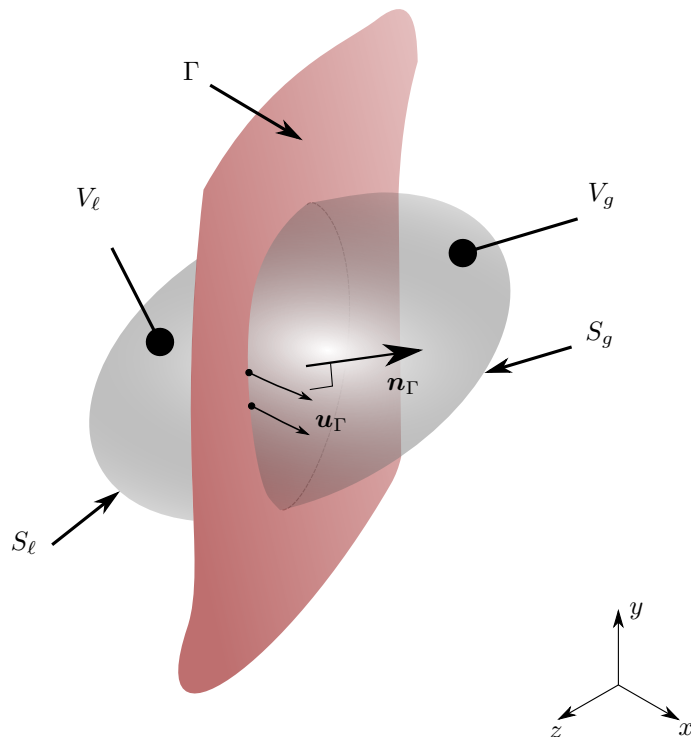


Figure 1: The control volume under consideration, V , contains a sharp interface Γ (red shading) and is bounded in space by fixed surfaces S_ℓ and S_g . The interface has velocity \mathbf{u}_Γ , which may differ from the velocity of the adjacent fluid.

For phase change at the interface, Γ , we define \dot{m}'' as the rate at which liquid is converted to gas (or vapour). It has units of mass per unit area per unit time, so that

$$\dot{m}'' = \rho_\ell (\mathbf{u}_\ell - \mathbf{u}_\Gamma) \cdot \mathbf{n}_\Gamma = \rho_g (\mathbf{u}_g - \mathbf{u}_\Gamma) \cdot \mathbf{n}_\Gamma. \quad (1)$$

Here \mathbf{u}_ℓ and \mathbf{u}_g are the respective liquid and gas phase velocities adjacent to the interface. The velocity of Γ is \mathbf{u}_Γ .

We use the Heaviside function, H , as an indicator for the liquid phase, so that:

$$H(\mathbf{x}, t) = \begin{cases} 1, & \text{if } \mathbf{x} \text{ is inside } V_\ell \\ 0, & \text{if } \mathbf{x} \text{ is inside } V_g. \end{cases} \quad (2)$$

The definition of H can be constructed by using the integral of consecutive products of one dimensional Dirac delta functions, which is extended to three dimensions for V from the derivation in [13, p.279].

The rate of change of H in V is given by

$$\int_V \frac{\partial H}{\partial t} dV + \oint_S H \mathbf{u} \cdot \mathbf{n} dA + \int_\Gamma (\mathbf{u}_\ell - \mathbf{u}_\Gamma) \cdot \mathbf{n}_\Gamma dA = 0, \quad (3)$$

where the last term on the left hand side accounts for phase change at the interface along with the previously defined convention that $H = 1$ in the liquid phase.

The VOF represents the volumetric fraction of liquid in V . We will denote it c , from the popular usage of *colour* function

$$c = \frac{1}{V} \int_V H(\mathbf{x}, t) dV. \quad (4)$$

From the equation above, it can be deduced that $0 \leq c \leq 1$. For V completely filled with liquid $c = 1$ and $c = 0$ when only gas is present. Note that c is a scalar function of which the definition is always relative to the volume in which it is defined. Using (1), (3) and (4), we can write for c that

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}c) + \frac{\dot{m}''}{\rho_\ell} \delta_\Gamma = 0, \quad (5)$$

where δ_Γ is a Dirac delta function defined on Γ . Equation (5) is intended in a weak sense, since c is not a continuous function and is solved by integrating over a volume. Let $\bar{\phi}$ be some volume averaged material property. We then have that

$$\bar{\phi} = \phi_\ell c + (1 - c)\phi_g \quad (6)$$

with the subscripts indicating the phase. We can now write a single set of the governing equations, also referred to as the “one-fluid” approach. The fluid density (ρ), kinematic viscosity (μ) and heat capacity ($C_p = \rho c_p$) is calculated using (6), with the product of density and specific heat capacity creating a heat capacity per unit volume.

From conservation of mass in V we have that

$$\int_V \nabla \cdot \mathbf{u} dV = \left(\frac{1}{\rho_g} - \frac{1}{\rho_\ell} \right) \int_V \dot{m}'' \delta_\Gamma dV. \quad (7)$$

Here \mathbf{u} is the fluid velocity.

The conservation of momentum is written

$$\int_V \frac{\partial(\rho \mathbf{u})}{\partial t} dV + \oint_S \rho \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dA = \oint_S \boldsymbol{\tau} \cdot \mathbf{n} dA + \int_V \rho \mathbf{g} dV + \int_V \sigma \kappa \delta_\Gamma \mathbf{n}_\Gamma dV, \quad (8)$$

with $\boldsymbol{\tau}$ the stress tensor, which for our problem is

$$\boldsymbol{\tau} = -p \mathbf{I} + \mu (\nabla \mathbf{u} + (\nabla \mathbf{u})^T). \quad (9)$$

Here p is the static pressure, \mathbf{I} the unit tensor and μ the viscosity. The last integral in (8) represents the surface tension force on Γ with σ the surface tension coefficient and κ the curvature of Γ .

Conservation of thermal energy is given by

$$\int_V \frac{\partial(C_p T)}{\partial t} dV + \oint_S C_p T \mathbf{u} \cdot \mathbf{n} dA = \oint_S k \nabla T \cdot \mathbf{n} dA - \int_V \dot{q}_\Gamma \delta_\Gamma dV, \quad (10)$$

with the thermal energy flux term in the second integral calculated conservatively in this work, contrary to many approaches where that integral is written as $\overline{C_p} \oint_S T \mathbf{u} \cdot \mathbf{n} dA$. T is the temperature and k the thermal conductivity coefficient. The last term in (10) is a thermal energy source located on the interface due to phase change latent heat transfer

$$\dot{q}_\Gamma = \dot{m}'' h_{fg} = k_g \nabla_{n_\Gamma} T_g - k_\ell \nabla_{n_\Gamma} T_\ell, \quad (11)$$

with h_{fg} the latent heat of evaporation and $\nabla_{n_\Gamma} T$ the normal temperature gradient on Γ in the respective phase.

The above equations are discretized on a regular Cartesian grid prior to an iterative numerical solution process. This is described next.

3. Numerical Method

The system of equations described in the previous section is solved numerically on a marker-and-cell (MAC) type computational mesh (also known as a staggered grid), created by Harlow and Welch [14]. Discrete velocity components are located on cell faces, while other scalar variables – like pressure, VOF and temperature – are located at cell centres. The time integration procedure will first be presented to provide an overview of the scheme, after which novel numerical techniques will be detailed. These include a PLIC based mass transfer rate calculation, a novel method of dealing with interface velocity jump conditions while maintaining a sharp interface definition and the discretisation of the thermal energy advection in conservative form using consistent VOF flux calculations for thermal heat capacity.

3.1. Solution overview

We next detail the time integration procedure, assuming that all variables are known at some time step n for all computational cells and initial values provided when $n = 0$. Time integration is done with an explicit, first order (forward Euler) scheme, using a split pressure projection method [15].

1. The forward Euler time integration of the thermal energy conservation equation is first applied to each cell and reads

$$C_p T|^{n+1} = C_p T|_n + \Delta t [\nabla^h \cdot k \nabla^h T|^{n+1} - \nabla^h \cdot (C_p T \mathbf{u})|_n], \quad (12)$$

where Δt is the discrete time step. The h superscript for the gradient operator indicates that it is defined in a discrete sense. The diffusion of thermal energy is calculated implicitly. *Note that when calculating the thermal energy diffusion term, we solve each phase separately and apply an interface Dirichlet condition for temperature. This implicitly includes the effect of latent heat energy and will be detailed in Section 3.4.2.* Calculation of the thermal energy advection term will be discussed in more detail in Section 3.4.1.

The mass transfer rate at the interface, \dot{m}'' , is calculated from the temperature field, T^n

$$\dot{m}'' = \frac{1}{h_{fg}} (k_g \nabla_{n_\Gamma}^h T|_g^n - k_\ell \nabla_{n_\Gamma}^h T|_\ell^n). \quad (13)$$

This is done for all cells where $\epsilon < c < 1 - \epsilon$, with $\epsilon = 10^{-12}$ in this work. The method used to calculate this term is described in Section 3.2.

2. The VOF equation is integrated in time as

$$\frac{c^{n+1} - c^n}{\Delta t} + \nabla^h \cdot (\mathbf{u}c)|_n + \frac{\dot{m}''}{\rho_\ell} \delta_\Gamma|_n = 0. \quad (14)$$

This equation is solved numerically using a geometric (PLIC) method. The procedure is detailed in Section 3.3. All fluid properties that depend on c can now be calculated at time $n + 1$ (ρ^{n+1} , μ^{n+1} , C_p^{n+1} , κ^{n+1} , δ_Γ^{n+1} , \mathbf{n}_Γ^{n+1}).

3. The predicted velocity \mathbf{u}^* in the momentum equation is calculated from

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\mathbf{u}^n \cdot \nabla^h \mathbf{u}^n + \frac{1}{\rho^{n+1}} (\nabla^h \cdot 2\mu \mathbf{S} + \sigma \kappa \delta_\Gamma \mathbf{n}_\Gamma |^{n+1}) + \mathbf{g}. \quad (15)$$

Here \mathbf{S} is the fluid strain rate tensor, given by $\mathbf{S} = 1/2 (\nabla \mathbf{u} + (\nabla \mathbf{u})^T)$. Note that the viscous term can be calculated explicitly ($\nabla^h \cdot 2\mu \mathbf{S}^n$) or implicitly ($\nabla^h \cdot 2\mu \mathbf{S}^{n+1}$). For more detail on the discretization of this equation, refer to [16].

4. A pressure Poisson equation is then solved for the pressure at $n + 1$

$$\nabla^h \cdot \left[\frac{\Delta t}{\rho^{n+1}} \nabla^h p^{n+1} \right] = \nabla^h \cdot \mathbf{u}^* - \dot{m}'' \left(\frac{1}{\rho_g} - \frac{1}{\rho_\ell} \right) \delta_\Gamma^n \quad (16)$$

5. The one fluid velocity \mathbf{u}^{n+1} is obtained with a correction step using p^{n+1}

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho^{n+1}} \nabla^h p^{n+1} \quad (17)$$

All variables have now been solved at time step $n + 1$ and the cycle can repeat for the next time step. The numerical method presented here has been implemented in PARIS with full parallel processing capability for three dimensional flow.

3.2. Calculating the mass transfer rate

This section describes the calculation of the mass transfer rate, \dot{m}'' , as given by (13). The numerical procedure calculates a value for \dot{m}'' in all cells containing a portion of the interface, known as mixed cells. We repeat (13) here for some mixed cell i, j, k ,

$$\dot{m}''_{i,j,k} = \frac{1}{h_{fg}} \left(k_g \nabla_{n_\Gamma}^h T|_{i,j,k}^g - k_\ell \nabla_{n_\Gamma}^h T|_{i,j,k}^\ell \right), \quad (18)$$

where $\nabla_{n_\Gamma}^h$ indicates a discrete gradient either side of the interface. To calculate the normal temperature gradients, we disregard the temperature in mixed cells, since these temperatures essentially represents some volume average of the temperatures in both phases.

A method to calculate the interface normal temperature gradients was proposed by Kunkelmann [9] and also applied by Guedon [10]. Temperature gradients are calculated in pure, single-phase cells which lie adjacent to mixed cells. Kunkelmann's method uses interpolation to obtain the $c = 0.5$ iso-surface, the location of which is used to calculate a finite difference temperature gradient in these pure cells. The mass transfer rate in mixed cells are then obtained using an average from pure cell neighbours.

We employ a similar approach to Kunkelmann, but instead of using an interpolation technique for the interface location, we use the PLIC reconstruction. The process is illustrated in Fig. 2. Pure cells are indicated with filled dots, while mixed cells are indicated with open dots and contain a plane (PLIC) reconstruction of the interface. The temperature gradient normal to the interface is calculated in pure cells which are first (darker shade) or second (lighter shade) neighbours of mixed cells.

For a pure cell, the heat flux is simply estimated using a finite difference

$$k_p \nabla_{n_p}^h T = k_p \frac{T - T_\Gamma}{d_\Gamma}, \quad (19)$$

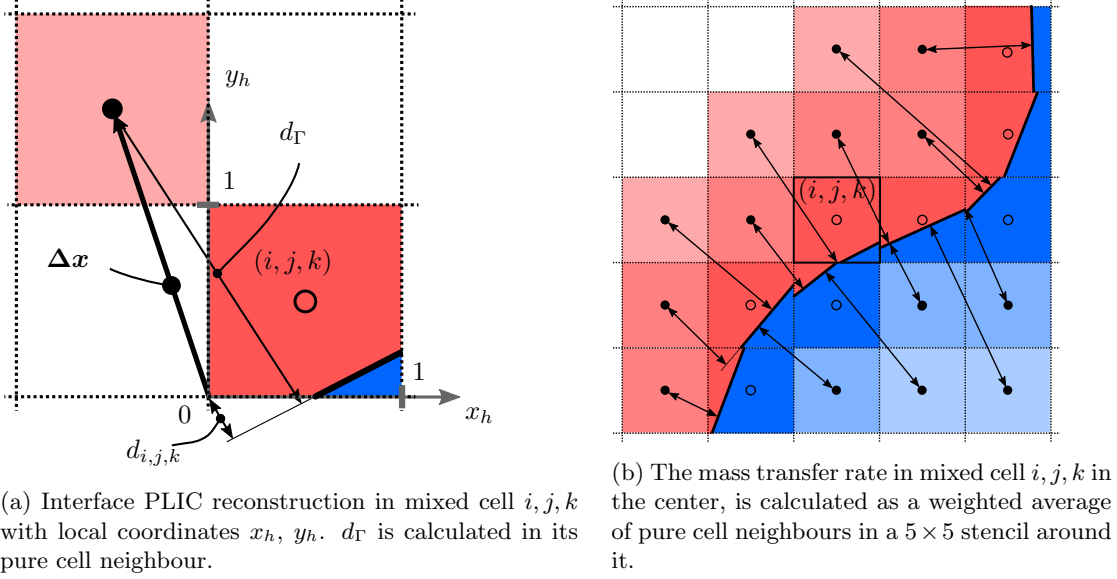


Figure 2: Mass transfer rate calculation for mixed cells.

with d_{Γ} the normal distance from the cell node to the interface and $p = \ell, g$ a phase indicator. It is assumed that $T_{\Gamma} = T_{sat}(p_{sys})$. The interface is defined in a mixed cell as a plane given by the implicit equation

$$\alpha = \mathbf{m} \cdot \mathbf{x}, \quad (20)$$

with α a constant and $\mathbf{m} = \langle m_1, m_2, m_3 \rangle$ the interface vector. A local coordinate system is defined in each cell, with its origin in the corner closest to the larger domain origin. The interface vector is scaled such that

$$|m_1| + |m_2| + |m_3| = 1, \quad (21)$$

and is calculated using the VOF field with the Mixed Youngs-centered method [17]. The plane constant α is obtained by using geometrical volume calculations [18]. In the local cell coordinate system of a mixed cell, the normal distance to the interface from the cell origin is therefore simply found by normalizing the interface vector (20)

$$d_{i,j,k} = \frac{\alpha}{\|\mathbf{m}\|} = \mathbf{n}_{\Gamma}^h \cdot \mathbf{x}, \quad (22)$$

with \mathbf{n}_{Γ}^h the planar interface normal vector.

To obtain the normal distance to the interface from any neighbouring pure cell, we apply a coordinate transformation. Let $\Delta \mathbf{x}$ be the translation vector from the mixed cell origin to the pure cell node. The normal distance from this pure cell node is then

$$d_{\Gamma} = d_{i,j,k} - \mathbf{n}_{\Gamma}^h \cdot \Delta \mathbf{x}. \quad (23)$$

Inside a pure cell, one can imagine that through the transformation in (23), the distance can be obtained to the interface representation in various neighbouring mixed cells. The interface distance is determined using the neighbouring mixed cell of which the collinearity of the plane normal in that cell and the translation vector is greatest, with the collinearity given by

$$\xi = |\mathbf{n}_\Gamma^h \cdot \Delta \mathbf{x}|. \quad (24)$$

The interface heat flux can now be calculated in the pure cell using (19), since its temperature is known and the interface normal distance is calculated, Fig. 2a. This is done for all pure cells that are first or second neighbours to mixed cells. Finally, the fluxes required to compute phase change in (18) are computed by using a weighted average of pure cell heat fluxes. This is done by considering all the first and second, pure cell neighbours in the desired phase within a 5×5 stencil, as shown in Fig. 2b. The example of liquid is given here, but the same applies for the gas phase:

$$\nabla_{n_\ell}^h T|_{i,j,k}^\ell = \sum_{q=1}^n \chi_q \frac{T_q^\ell - T_\Gamma}{d_{\Gamma,q}}, \quad (25)$$

where q is an index for all n neighbours in the stencil. The normalized weighting factor χ_q is given by

$$\chi_q = \frac{\gamma_q}{\sum_{c=1}^n \gamma_c}, \quad (26)$$

with γ determined by the collinearity of the neighbour and the square of its distance from the mixed cell

$$\gamma_q = \frac{\xi_q}{\|\Delta \mathbf{x}_q\|^2} \quad (27)$$

The temperature gradients calculated in (25) on either side of the interface are then used in (18) to obtain \dot{m}'' in all mixed cells.

3.3. Geometric VOF advection for phase change

Several VOF advection schemes exist for incompressible, divergence free flow fields. For more details on geometric methods, refer to [19],[13, p.95]. The schemes available in PARIS include the method of Li [20] (sometimes referred to as *CIAM* or Lagrangian Explicit) and the conservative method of Weymouth and Yue [11]. However, these methods cannot be applied directly to (14), due to the existence of the phase change source term.

In this work we present a novel method to solve (14) using existing geometric advection methods designed for divergence-free velocity fields, while retaining the sharp interface approach. The method splits (14) into two steps:

$$\frac{\partial c}{\partial t} \approx \frac{c^{n+1} - c^n}{\Delta t} = \frac{1}{V} \int_V \frac{\partial H_1}{\partial t} + \frac{\partial H_2}{\partial t} dV, \quad (28)$$

where the numeric subscripts indicate the step numbers.

The first step can be seen as a material advection of the liquid, in which the Heaviside function (3) gets advected with the liquid velocity field. The second step accounts for the phase change term, or the relative velocity between the interface and liquid velocities. These steps are detailed next and are shown schematically in Fig. 3.

3.3.1. Step 1: Advecting with an extended, divergence free liquid velocity

The first step advects the Heaviside function with the liquid velocity \mathbf{u}_ℓ , to produce an intermediate VOF field, \tilde{c} :

$$\frac{\tilde{c} - c^n}{\Delta t} = \frac{1}{V} \int_V \frac{\partial H_1}{\partial t} dV = -\frac{1}{V} \oint_S H^n \mathbf{u}_\ell \cdot \mathbf{n} dA = -\nabla \cdot (\mathbf{u}_\ell c). \quad (29)$$

This step effectively assumes that the VOF field is advected at the velocity of the liquid phase, which is indeed the case everywhere inside the liquid. In order to use the existing geometric advection schemes, the velocity in the liquid is extended across the interface in a manner that ensures

$$\nabla \cdot \mathbf{u}_\ell = 0. \quad (30)$$

An existing technique to achieve this using the ghost-fluid method has been proposed by [21]. We propose here a novel and simple technique to obtain \mathbf{u}_ℓ , which is readily implementable into existing incompressible flow solvers. After solution of equations (16) and (17), the one-fluid velocity field (\mathbf{u}^{n+1}) is obtained, which will contain a discontinuity at the interface when phase change occurs (for phases with different densities).

We now construct a sub-domain around the interface, of which a 2D example is shown in Fig. 3b. The sub-domain is delimited on the liquid side of the interface by defining all the cell faces between mixed and liquid cells as domain boundaries on which a no-slip boundary condition is applied (red line). On the other side of the interface, all the cell faces between second and third pure gas cell neighbours (blue dotted line) are considered as free-flow faces (a fixed pressure is applied with a zero velocity gradient normal to the boundary).

A Poisson problem is set up on this newly created sub-domain in a manner similar to (16)

$$\nabla^h \cdot \left[\frac{\Delta t}{\rho^{n+1}} \nabla^h \tilde{p} \right] = \dot{m}'' \left(\frac{1}{\rho_g} - \frac{1}{\rho_\ell} \right) \delta_\Gamma, \quad (31)$$

with the same phase change source term, but of opposite sign. This equation is solved with the same convergence tolerance as for the previous problem. A phase change related velocity correction is then obtained on the cell faces of the newly defined sub-domain, similar to (17)

$$\tilde{\mathbf{u}} = -\frac{\Delta t}{\rho^{n+1}} \nabla^h \tilde{p}^{n+1}. \quad (32)$$

with $\tilde{\mathbf{u}}$ the newly obtained velocity field. It is defined everywhere on the cell faces inside the light blue sub-domain in Fig. 3b, elsewhere it is zero. The divergence free liquid velocity field can now be found using

$$\mathbf{u}_\ell^{n+1} = \mathbf{u}^{n+1} + \tilde{\mathbf{u}}. \quad (33)$$

This step extends the velocity of the liquid phase across the interface for all faces up to the third neighbours in the gas phase. Velocities that are further into the gas phase are not of concern, as they do not affect the VOF field. The \mathbf{u}_ℓ field is divergence free inside the liquid and up to the extension into the gas phase by construction

$$\nabla \cdot \mathbf{u}_\ell = \nabla \cdot \mathbf{u}^{n+1} + \nabla \cdot \tilde{\mathbf{u}} = \dot{m}'' \left(\frac{1}{\rho_g} - \frac{1}{\rho_\ell} \right) \delta_\Gamma - \dot{m}'' \left(\frac{1}{\rho_g} - \frac{1}{\rho_\ell} \right) \delta_\Gamma = 0. \quad (34)$$

We now use existing geometric advection schemes to calculate \tilde{c} , shown in Fig. 3c.

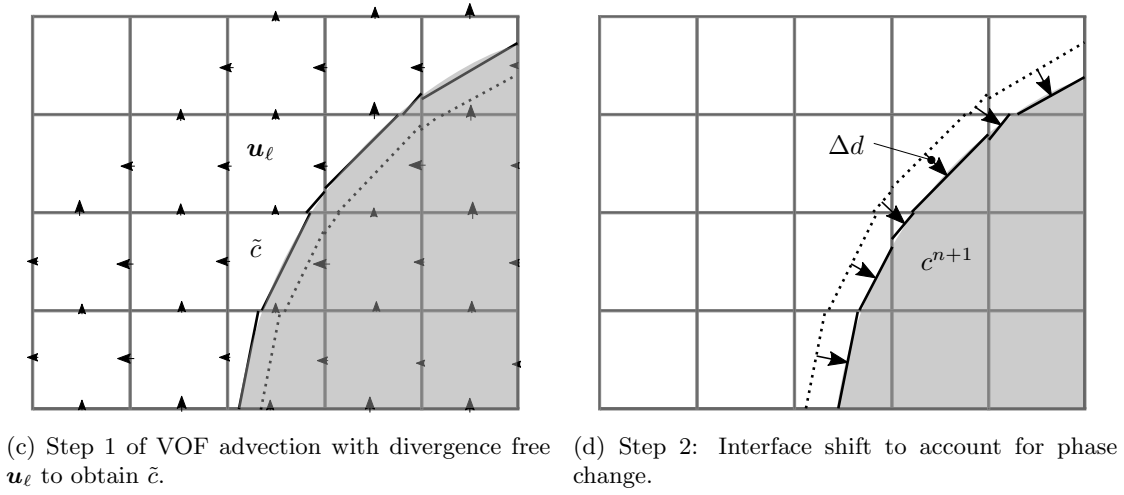
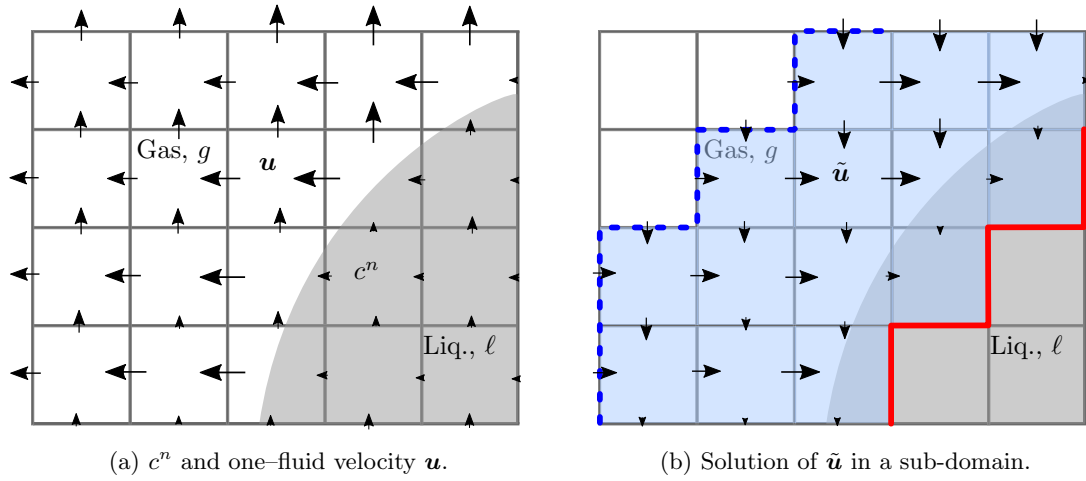


Figure 3: Schematic of split geometric VOF advection method for an evaporating droplet.

3.3.2. Step 2: Shifting the interface to account for phase change

In the second step of the split VOF equation, the interface shift due to phase change is accounted for

$$\frac{c^{n+1} - \tilde{c}}{\Delta t} = \frac{1}{V} \int_V \frac{\partial H_2}{\partial t} dV = -\frac{1}{V} \int_{\Gamma} (\mathbf{u}_\ell - \mathbf{u}_\Gamma) \cdot \mathbf{n}_\Gamma dA = -\frac{\dot{m}''}{\rho_\ell}. \quad (35)$$

This is illustrated in Fig. 3d. This step accounts for the relative movement between the interface and neighbouring liquid molecules (bearing in mind that the liquid is the tracked phase). The relative velocity $(\mathbf{u}_\Gamma - \mathbf{u}_\ell) \cdot \mathbf{n}_\Gamma$ is a consequence of mass transfer, (1).

To satisfy (35), we need to shift the interface in the normal direction. We define the shift in the interface Δd

$$\Delta d = (\mathbf{u}_\Gamma - \mathbf{u}_\ell) \cdot \mathbf{n}_\Gamma = -\frac{\dot{m}'' \Delta t}{\rho_\ell h}, \quad (36)$$

with h the cell length used to re-scale the interface shift to the local cell coordinate system (see Fig. 2a), wherein the PLIC plane constant is defined (20). We once again use the PLIC reconstruction and existing geometric VOF library in PARIS [18] to calculate c^{n+1}

$$\begin{aligned} \alpha^{n+1} &= \tilde{\alpha} + \|\mathbf{m}\| \Delta d \dots 0 \leq \alpha^{n+1} \leq 1 \\ c^{n+1} &= f(\alpha^{n+1}, \mathbf{m}). \end{aligned} \quad (37)$$

The movement is capped to avoid local over- and undershoots and respect $0 \leq c^{n+1} \leq 1$. When they do occur, the clipped amount is accounted for in the neighbour that is located in the direction of interface movement in order to still respect mass conservation.

When this step is completed, the VOF equation (14) is satisfied. The advection of thermal energy is solved such that fluxes in heat capacity are calculated in a consistent manner with the VOF fluxes, as detailed next.

3.4. Energy conservation

A thermal energy conservation equation (12) was added to PARIS to facilitate the simulation of phase change flows.

3.4.1. Calculating the energy advection term

The advection term is calculated as

$$C_p^* T^* - C_p^n T^n = -\Delta t \sum_f C_p^n|_f T_f^n u_f^n A_f, \quad (38)$$

where $C_p^* T^*$ indicates the thermal energy per unit volume after advection is considered. The right-hand-side represents the sum of thermal energy fluxes at all f cell faces for time step $n + 1$. Each face has an area $A_f = h^2$ (in 3D) and face velocity $u_f = \mathbf{u} \cdot \mathbf{n}_f$, which is calculated explicitly on the staggered (MAC) grid. Care needs to be taken when calculating $C_p^n|_f$ as well as T_f^n , respectively the fluxed heat capacity and face temperature. One reason is that the volumetric heat capacity ratio becomes significant in many problems of interest. Consider, as an example, the properties of water at atmospheric conditions in Table 1. In this case $C_{p,\ell}/C_{p,g} \approx 3200$, which motivates why a conservative formulation is preferable, as opposed to the non-conservative approximation

Phase	Density ρ [$kg.m^3$]	Specific heat capacity c_p [$kJ.kg^{-1}.K^{-1}$]	Vol. heat capacity C_p [$kJ.m^{-3}.K^{-1}$]
Gas (vapour)	0.6	2.080	1.248
Liquid	958	4.216	4.039×10^3

Table 1: Properties for saturated water at atmospheric conditions

$$C_p^* T^* - C_p^n T^n|_{i,j,k} = -\Delta t C_p^n|_{i,j,k} \sum_f T_f^n u_f^n A_f. \quad (39)$$

A flux-consistent, geometric advection technique is employed. The term flux-consistent is used, since the flux terms in (38) are calculated using the same geometric advection procedure as for the VOF function. Note that the flux-consistent implementation here is inspired by a similar method already present in PARIS, but applied to the momentum term [22].

In the same manner as the directionally split VOF advection methods ([20, 11]) three separate, consecutive sweeps are performed, one for each coordinate direction (in 3D). Let s be the sweep counter, so that $s = 1$ corresponds to time step n , $s = 1, 4$ and $s = 4$ corresponds to time step $n + 1$. Before the sweeps start, a thermal energy e_t is calculated for every cell

$$e_t^{(s=1)} = e_t^n = C_p^n T^n, \quad (40)$$

where the cell temperature is known from the solution at time step n and the volumetric heat capacity in a cell is obtained by using a volume average of the phase heat capacities

$$C_p^{(s=1)} = C_p^n = C_{p,\ell} c + C_{p,g} (1 - c). \quad (41)$$

For each direction sweep, the following process is followed:

1. The face temperatures T_f are obtained by using a fifth order WENO reconstruction scheme [23].
2. The fluxed volumes and their respective VOF values are known from the geometric VOF flux calculations at each of the two faces in the specific sweep direction. This allows us to calculate a volume weighted volumetric heat capacity $C_p|_f$ and the fluxed thermal energy at each face from the temperature in the WENO scheme $C_p|_f T_f$. We now have the intermediate thermal energy in the cell (after the sweep) and can calculate the intermediate temperature

$$T^{(s)} = \frac{e_t^{(s)}}{C_p^{(s)}}, \quad (42)$$

3. The thermal energy and VOF sweeps are done in tandem for every sweep direction. Before the next direction is swept, the interface is reconstructed from the newly calculated intermediate VOF $c^{(s+1)}$, along with the thermal energy at the same sweep $e_t^{(s+1)}$. This ensures that the heat capacity that is used at each fluxing step is consistent with the VOF function. The face temperatures are calculated by using the updated temperature field after each sweep.

After the three coordinate directions have been swept using \mathbf{u}_ℓ , (just like the VOF process) the energy change due to phase change is calculated by

$$\Delta e_t = \Delta c (C_{p,\ell} - C_{p,g}) T_\Gamma \quad (43)$$

3.4.2. Calculating the energy diffusion term

The heat diffusion term needs to be evaluated to complete the integration of (12) in time. This term is calculated implicitly using a similar approach to Sato and Ničeno [24]. A similar idea was also imposed in [25] and [26] using the ghost fluid method. The saturation temperature at the interface is applied directly by solving the diffusion term separately for each phase. The interface is treated as a boundary, where a Dirichlet boundary condition is specified for the temperature. This is equivalent to solving two heat diffusion equations with constant fluid properties, with the interface an arbitrary boundary separating the two phases in which each respective problem is solved. The fact that the temperature gradient discontinuity is explicitly enforced by applying an interface boundary condition accounts for latent heat absorption or release during phase change as was also directly pointed out in [24] and used in [25] and [26].

First, the phase of a specific node is determined by simply evaluating the VOF function: A cell node is determined to be inside the liquid whenever $c \geq 0.5$, otherwise it is in the gas. Fig. 4 shows two 2D stencils used to compute the diffusion term for a liquid (left hand side) and gas (right hand side) node.

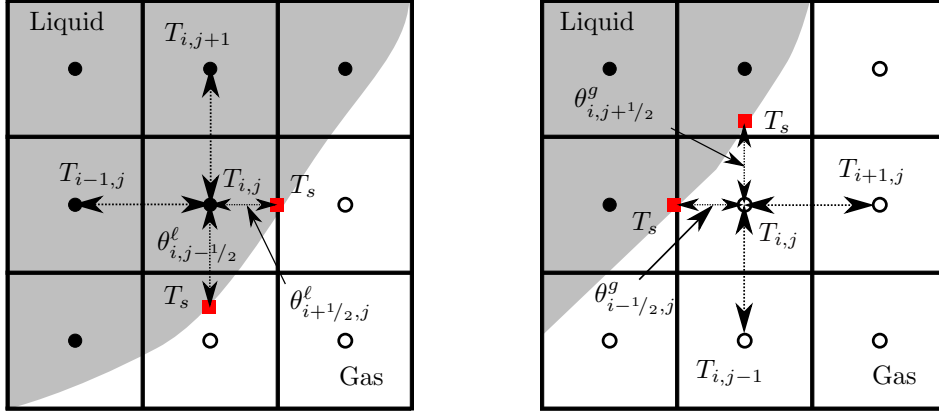


Figure 4: Discretisation for temperature neighbouring the interface. On the left the computational stencil for a liquid cell is shown, with the stencil for its neighbour in the gas on the right.

Consider the liquid node on the left as an example. The finite volume discretisation of the diffusion term for the 2D cell at position i, j is approximated using the irregular stencil as

$$\frac{1}{V} \int_{V_{i,j}} \nabla \cdot (k \nabla T) dV \approx k_\ell \left(\frac{\nabla_y^h T_{i,j+1/2} - \nabla_y^h T_{i,j-1/2}}{1/2 (h + \theta_{i,j-1/2}^\ell)} + \frac{\nabla_x^h T_{i+1/2,j} - \nabla_x^h T_{i-1/2,j}}{1/2 (\theta_{i+1/2,j}^\ell + h)} \right), \quad (44)$$

noting that $\Delta x = \Delta y = h$ and $k = k_\ell$ since we are only considering the liquid.

The temperature gradient is approximated using a finite difference. Consider again the liquid node:

$$\nabla_x^h T_{i+1/2,j} = \frac{T_\Gamma - T_{i,j}}{\theta_{i+1/2,j}^\ell}; \quad \nabla_x^h T_{i-1/2,j} = \frac{T_{i,j} - T_{i-1,j}}{h}. \quad (45)$$

For the temperature gradient to the right of the node in question, the interface distance θ^ℓ is used to calculate the finite difference. A standard finite difference is used when the temperature gradient is approximated between two nodes in the same phase.

The distance to the interface from a node is simply taken as the height function in that direction. When the interface configuration is such that a height cannot be obtained in the required direction, the distance is approximated by using a plane reconstruction of the interface in the staggered volume. The distance is capped so that $\epsilon h \leq \theta \leq h$, with a value of $\epsilon = 10^{-3}$ typically used.

3.5. A note on Momentum Conservation

Before proceeding to present results, a word is due on the numerical treatment of the momentum conservation equation. For the purpose of this work a non-conservative momentum advection formulation was used. The two-step advection process was not applied to the existing momentum conserving scheme [22] in PARIS due to additional numerical routines required to do so. One challenge is the fact that the momentum control volume is staggered to the pressure and thermal energy control volumes on the staggered (MAC) grid.

The error introduced with the use of the non-conservative discretization of the momentum equation manifests in the force balance on the interface. Without a conservative discretisation, an error is introduced for the vapour recoil pressure. In the non-conservative discretisation, the momentum advection term is calculated for a control volume as

$$\bar{\rho} \oint_S \mathbf{u} (\mathbf{u} \cdot \mathbf{n}) dA, \quad (46)$$

with $\bar{\rho}$ the cell averaged density. For mixed cells, the assumption of momentum fluxes at constant (average) density is incorrect, therefore the resulting pressure does not match the correct recoil value. By applying a consistent, conservative discretisation of the advection term, the correct vapour recoil pressure may be obtained. Alternatively, as pointed out in [27], one can enforce the correct momentum jump using a ghost-fluid type method.

The error in vapour recoil pressure was not found to be detrimental for the purpose of this work, but a conservative, consistent implementation for the momentum is envisaged in future work.

4. Simulation Cases

In this section we validate the novel numerical scheme with analytical benchmark test-cases. The numerical method was implemented in PARIS with full parallel processing capability in three spatial dimensions using the MPI libraries [28].

4.1. Interface distance calculation

This test case measures the accuracy of the PLIC-based interface distance calculation, as explained in Section 3.2. It is clear from (19) that the interface distance calculation is important when the normal temperature gradients are calculated for the mass transfer rate. As a first validation test, the VOF field was initialized as a plane in 3D. The theoretical interface distances were compared to the calculated distances in PARIS for all first and second pure cell neighbours. The calculated distances were found to correspond to the theoretical ones to machine precision (as expected).

Next, a 3D droplet test was performed. A droplet of radius $R_0 = 0.25 + \delta_R$ is initialized at $x_0 + \delta_x, y_0 + \delta_y, z_0 + \delta_z$ of a unit cube, with $x_0 = y_0 = z_0 = 0.5$. The δ -values are each a randomly generated number such that $-h \leq \delta \leq h$, with $h = 1/N$ the cell size for a mesh with N^3 grid points. Five different resolutions were tested: $N = \{16, 32, 64, 128, 256\}$. At each mesh resolution, 100 different droplets were generated for a total of 500 tests.

For each droplet, the calculated interface normal distances d^h were calculated for all first and second pure cell neighbours (in each phase). The relative error for a pure cell is calculated as

$$e_{i,j,k} = \frac{|d_{i,j,k}^h - d|}{R_0}. \quad (47)$$

For each droplet, the max error is recorded as well as an L_2 -norm of all n first- and second pure cell neighbours to mixed cells, with

$$L_2 = \sqrt{\frac{\sum_q^n e_q^2}{n}}. \quad (48)$$

The results are presented in Fig. 5. The convergence rate for the (L_2) average distance error is around second order at lower mesh resolutions. For higher mesh resolutions it is between first and second order.

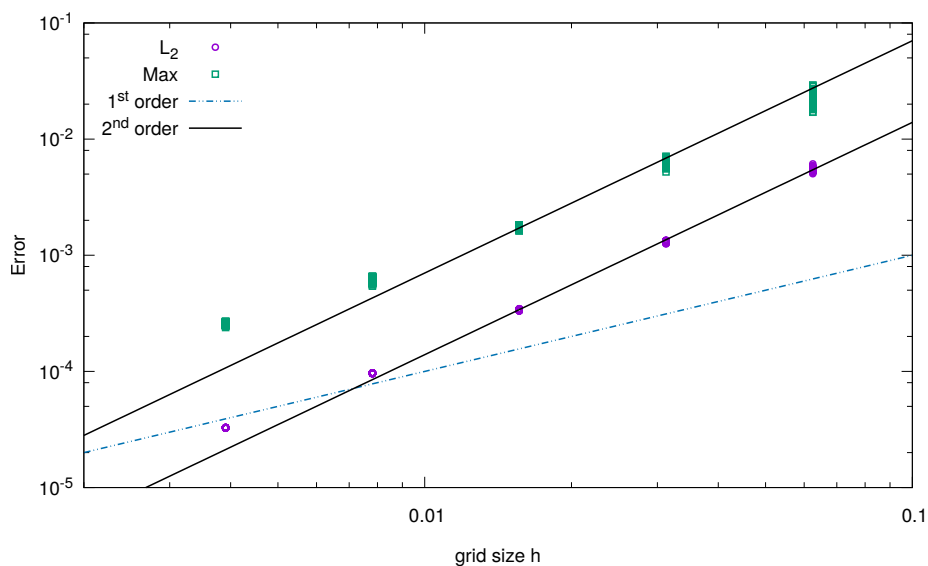


Figure 5: Convergence of the PLIC interface distance calculation.

4.2. Two-dimensional droplet evaporating test

One crucial aspect of the novel numerical method is the proposed two-step VOF advection procedure. This includes the correct calculation of a divergence free liquid velocity and subsequent phase change adjustment to ensure mass conservation. A two-dimensional evaporating droplet is modelled to validate this part of the implementation. The discretization errors that may be present from the mass transfer rate calculations are removed by specifying a constant mass transfer rate of $\dot{m}'' = 0.05$, ensuring that this test specifically evaluates the two-step VOF-advection procedure.

A two-dimensional, circular liquid droplet with initial radius $R = 0.23$ is initialized in the center of a square domain of unit length. The density ratio used in this test is $\rho^\ell/\rho_g = 2$. The units in this problem are irrelevant and we can consider the problem dimensionless. The boundaries are given an outflow boundary condition (fixed pressure and zero normal gradient

for the velocity). The constant mass transfer rate causes the droplet to evaporate. The radius evolution over time is simply

$$R(t) = R_0 - \dot{m}''t. \quad (49)$$

Three grid resolutions were used: $N_x = 32, 64$ and 128 with respective time step sizes of $t = 0.002 s$, $t = 0.001 s$ and $t = 0.0005 s$. The time evolution of the droplet volume is shown in Fig. 6 and compared to the analytical solution. As shown, an accurate solution is achieved on all meshes. Fig. 7 shows a time sequence of the droplet evolution for the finest resolution. The velocity magnitude is shown, revealing resulting radial flow patterns. It is encouraging to note that the circular shape is maintained throughout and the velocity field around the droplet remains symmetric. We present a zoomed snapshot of the interface in Fig. 8 with velocity vectors and the colour scale representing their magnitude. There is a clear jump across the interface due to the imposed, constant rate of mass transfer per unit area of interface.

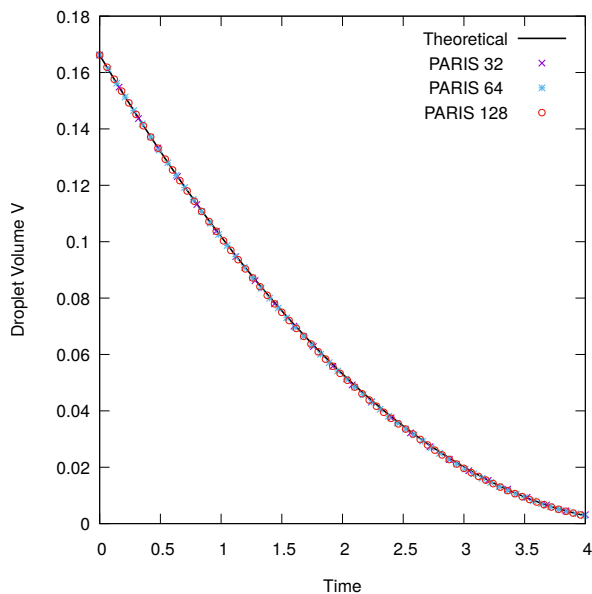


Figure 6: The volume evolution an evaporating cylinder in 2D with constant rate of evaporation.

4.3. Stefan problem

The Stefan problem was used by Welch and Wilson [4] to validate their VOF-based method and has since often been used to validate incompressible phase change methods [10, 24]. In this problem, a liquid at saturation temperature is initially at rest. A thin vapour layer is present between the liquid and a wall, which is at a fixed, elevated temperature of T_{sup} degrees above the saturation temperature, $T_{wall} = T_{sat} + T_{sup}$. Gravity is neglected and an outflow boundary condition is used opposite the heated wall.

The temperature gradient between the wall and the interface, which is at saturated temperature, causes phase change and the vapour layer to grow in time. The liquid is pushed out of the domain by the growing vapour layer. Let the thickness of the vapour layer be δ . An analytical solution is available for the evolution of the interface thickness and temperature profile [4]. The interface thickness is given by

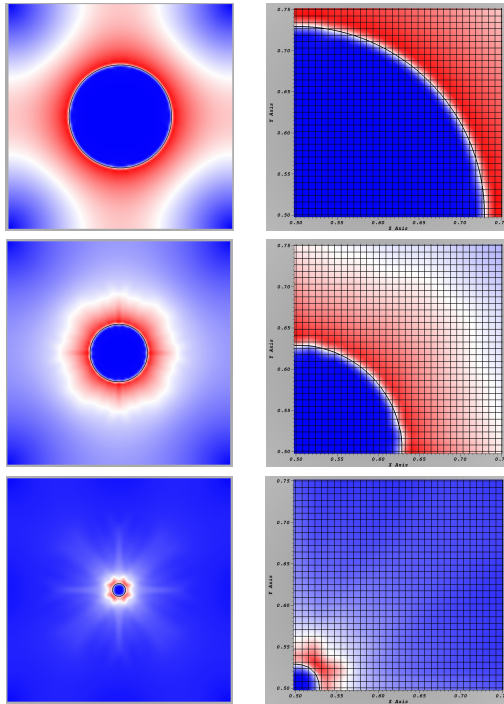


Figure 7: A time series showing the velocity magnitude for the finest resolution simulation ($N_x = 128$), taken at 0.01, 2.01 and 4.01s. A zoom showing the mesh is on the right.

$$\delta(t) = 2\lambda\sqrt{\alpha t}, \quad (50)$$

with t the time and $\alpha = k_g / \rho_g c_{p,g}$ the thermal diffusivity of the vapour phase. λ is found by solving the transcendental equation

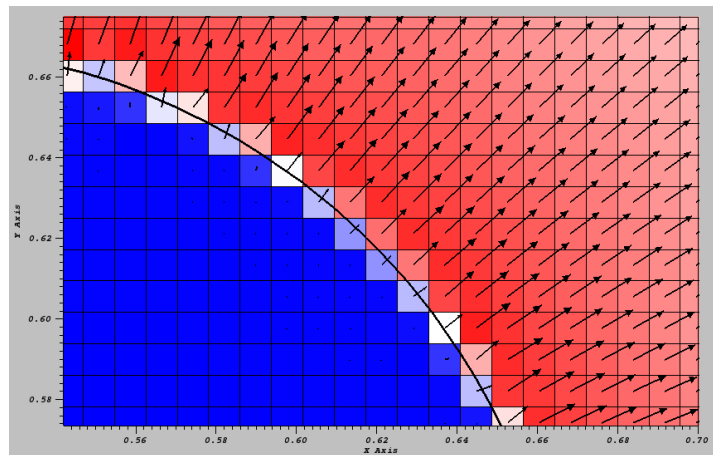


Figure 8: Zoomed snapshot of the interface with the velocity magnitude for the finest resolution simulation ($N_x = 128$) including velocity vectors to show the velocity jump.

Property	Units	Liquid	Vapour
Density ρ	$[kg.m^3]$	958	0.6
Specific heat c_p	$[kJ.kg^{-1}.K^{-1}]$	4216	2080
Viscosity μ	$[Pa.s]$	2.82×10^{-4}	1.23×10^{-5}
Thermal conductivity k	$[W.m^{-1}.K^{-1}]$	0.68	0.025
Latent heat h_{fg}	$[J.kg^{-1}]$	2.256×10^6	-
Surface tension σ	$[N.m^{-1}]$	0.059	-

Table 2: Properties for saturated water at atmospheric conditions

Grid points	δ PARIS	Relative error %
$N_x = 64$	1.93088×10^{-3}	0.62
$N_x = 128$	1.92650×10^{-3}	0.39
$N_x = 256$	1.92349×10^{-3}	0.23

Table 3: Relative errors for Stefan problem interface location at $t = 10 s$.

$$\lambda \exp(\lambda^2) \operatorname{erf}(\lambda) = \frac{c_{p,g} T_{sup}}{h_{fg} \sqrt{\pi}}. \quad (51)$$

A two-dimensional test case was defined with a domain size of $10 mm$, a $T_{sup} = 10 K$ superheat and fluid properties for water at atmospheric conditions. These properties are given in Table 2. The problem can be characterized by the Jacob number, given by

$$Ja = \frac{\rho_l c_{p,l} (T_\infty - T_{sat})}{\rho_g h_{fg}}. \quad (52)$$

For the properties used in this case, $Ja = 29.84$. The initial temperature field was T_{sat} in the liquid and a linear profile from the interface to the wall temperature. Three grid spacings were used on the $10 mm$ domain: $N_x = 64, 128, 256$ with respective time step sizes of $t = 0.002 s$, $t = 0.001 s$ and $t = 0.0005 s$. The initial vapour layer thickness was taken as $322.5 \mu m$, which corresponds to a time $t = 0.282435 s$, if the vapour layer would have grown from zero thickness at time $t = 0 s$. The simulated time was $10 s$. A zoomed section view showing the interface, temperature field and velocity vectors are shown in Fig. 10. There is a linear temperature gradient from the superheated wall to the interface, across which we clearly see the velocity jump due phase change.

The theoretical solution of the evolution of the interface thickness was obtained by solving (51) numerically with a relative convergence error of 10^{-6} for λ . The results from PARIS for the three grid resolutions are plotted in Fig. 9. An excellent agreement was obtained. The relative errors of the interface position at $t = 10 s$ for the three grid resolutions are presented in Table 3. The theoretical solution of (50) for $t = 10 s$ is $\delta(10) = 1.919 \times 10^{-3} m$.

The results show an error of less than 1% for the coarsest grid. Since the errors obtained here are small, the spatial convergence rate will be determined with a more demanding test case.

4.4. Bubble in superheated liquid

A more demanding test case is that of a three-dimensional gas bubble at saturated temperature in a superheated liquid. The superheated temperature is denoted T_∞ and no gravity force is applied. There is an analytical solution to the problem assuming spherical symmetry, obtained by Scriven [29]. The solution gives the bubble radius R as

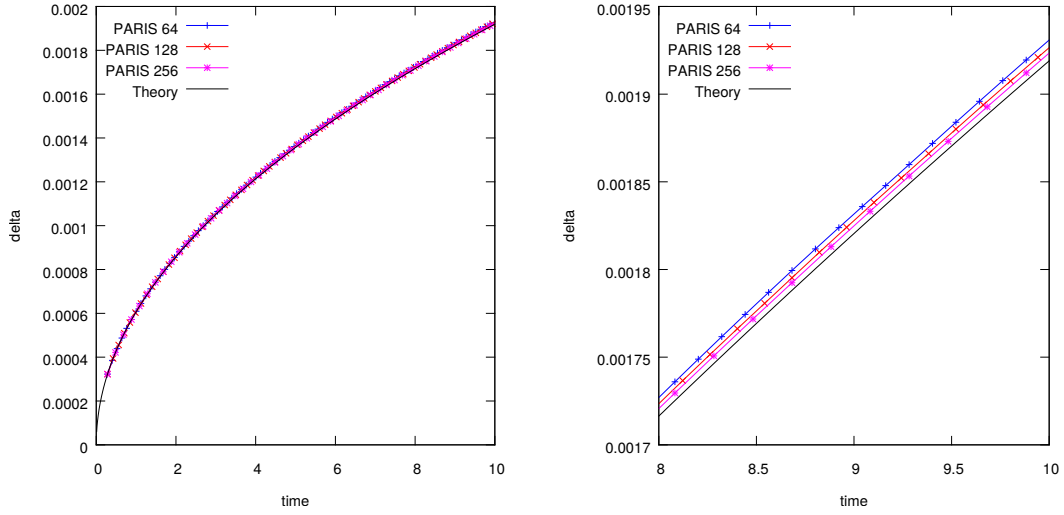


Figure 9: The interface time evolution results for the Stefan problem.

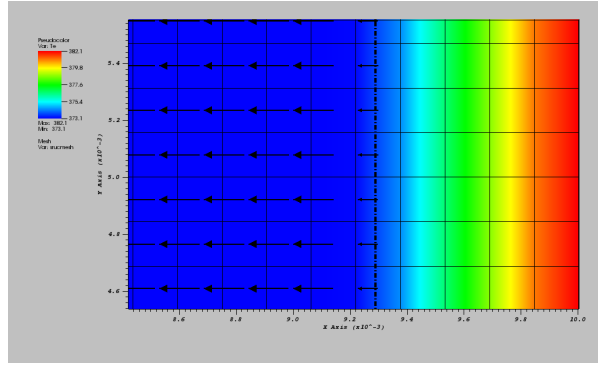


Figure 10: Zoom of an interface snapshot showing the velocity jump across the interface (dashed line) in the Stefan problem with colours representing the temperature.

$$R = 2 \beta_g \sqrt{\frac{k_\ell}{c_{p,\ell} \rho_\ell}} t \quad (53)$$

with t the time and the fluid properties as defined before. The value of β_g , sometimes referred to as the “growth constant” is obtained by solving

$$\frac{\rho_\ell c_{p,\ell} (T_\infty - T_{sat})}{\rho_g (h_{fg} + (c_{p,\ell} - c_{p,g}) (T_\infty - T_{sat}))} = 2 \beta_g^2 \int_0^1 \exp \left(-\beta_g^2 \left((1 - \zeta)^{-2} - 2 \left(1 - \frac{\rho_g}{\rho_\ell} \right) \zeta - 1 \right) \right) d\zeta \quad (54)$$

The solution is obtained numerically using the native numerical tools in the GNU package OCTAVE . The temperature field is then given by

Property	Units	Liquid	Vapour
Density ρ	$[kg.m^3]$	2.5	0.25
Specific heat c_p	$[J.kg^{-1}.K^{-1}]$	2.5	1.0
Viscosity μ	$[Pa.s]$	7.0×10^{-3}	7.0×10^{-4}
Thermal conductivity k	$[W.m^{-1}.K^{-1}]$	0.07	0.007
Surface tension σ	$[N.m^{-1}]$	0.001	-
Latent heat h_{fg}	$[J.kg^{-1}]$	100.0	-

Table 4: Properties for the bubble in superheated liquid case.

$$\begin{aligned}
T(r < R) &= T_{sat} \\
T(r > R) &= T_{\infty} - 2\beta_g^2 \left(\frac{\rho_g (h_{fg} + (c_{p,\ell} - c_{p,g})(T_{\infty} - T_{sat}))}{\rho_{\ell} c_{p,\ell}} \right) \\
&\quad \int_{1-R/r}^1 \exp \left(-\beta_g^2 \left((1-\zeta)^{-2} - 2 \left(1 - \frac{\rho_g}{\rho_{\ell}} \right) \zeta - 1 \right) \right) d\zeta \quad (55)
\end{aligned}$$

4.4.1. Ja 0.5 test case

A test case is defined with fluid properties given in Table 4. These properties do not necessarily correspond to a specific fluid, but was chosen for the purposes of a test with $\rho_{\ell}/\rho_g = \mu_{\ell}/\mu_g = k_{\ell}/k_g = 10$. The heat capacity ratio is $c_{p,\ell}/c_{p,g} = 2.5$.

The saturated temperature is $T_{sat} = 1K$ with the temperature at infinity in the liquid $T_{\infty} = 3K$. This results in $Ja = 0.5$. A bubble of initial radius $R(t_0) = 0.12$ is placed in the center of a unit cube. The initial temperature is taken from the analytical solution and the simulation is run for a time $t_f \approx 4 \times t_0$, with t_f the final time.

Grid points	Radius R PARIS	Relative error %
$N = 64^3$	2.279×10^{-1}	5.56
$N = 128^3$	2.361×10^{-1}	2.15
$N = 256^3$	2.404×10^{-1}	0.37

Table 5: Relative errors for the bubble radius R at $t = 4t_0$ for Ja 0.5.

Three test cases were run: $N = 64^3, 128^3, 256^3$ with respective time step sizes of $t = 0.01s$, $t = 0.005s$ and $t = 0.0025s$. The time evolution of the bubble radius in the simulations is compared to the theoretical value in Fig. 11. The theoretical value of the bubble radius at the final time was determined by the analytical solution: $R(t_f = 2.2156s) = 0.2413$ and compared to simulation. The results are shown in Table 5, along with the relative error compared to the theoretical value. The largest error was 5.56% on the coarsest grid.

One important source of the error is the resolution of the thermal boundary layer in the superheated liquid. Lower resolution results in an inaccurate calculation of the temperature gradient in the liquid, which leads to an inaccurate mass transfer rate, (18). A time sequence of the simulation is shown in Fig. 12, indicating the bubble growth and superheated temperature around the interface. It is encouraging to notice that the thermal boundary layer as well as the bubble shape remains spherically symmetric along with a relatively accurate prediction of the bubble radius evolution.

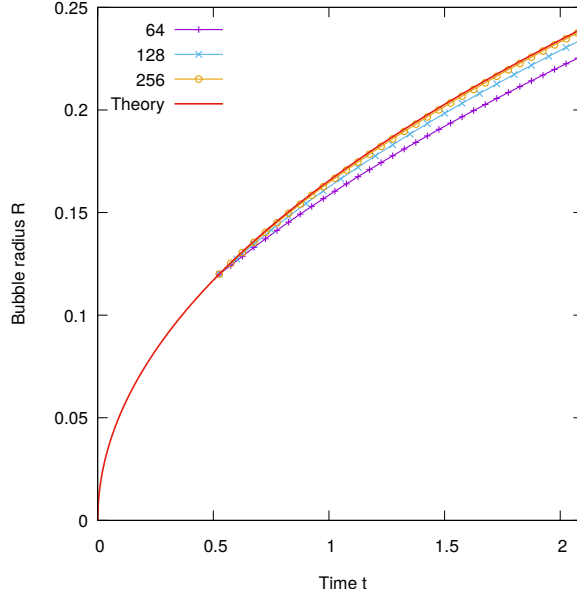


Figure 11: The time evolution of the bubble radius for a bubble at saturated temperature in a superheated liquid for Ja 0.5.

4.4.2. Ja 2.15 test case

A similar saturated bubble in superheated liquid test case is defined with fluid properties given in Table 6. The saturated temperature is $T_{sat} = 513.15 K$ with the temperature at infinity in the liquid $T_{\infty} = 519.15 K$ so that $Ja = 2.15$. A bubble of initial radius $R(t_0) = 1 mm$ is placed in the center of a cube with length $5 mm$. The initial conditions are again determined from the analytical solution, as in the previous case. The case is again run for a time $t_f \approx 4 \times t_0$ and $t_0 = 0.24265 s$.

Property	Units	Liquid	Vapour
Density ρ	$kg.m^3$	813.6	16.73
Specific heat c_p	$J.kg^{-1}.K^{-1}$	4.772×10^3	3.510×10^3
Viscosity μ	$[Pa.s]$	1.105×10^{-4}	1.722×10^{-5}
Thermal conductivity k	$[W.m^{-1}.K^{-1}]$	6.320×10^{-1}	4.870×10^{-2}
Surface tension σ	$[N.m^{-1}]$	2.84×10^{-3}	-
Latent heat h_{fg}	$[J.kg^{-1}]$	6.476×10^5	-

Table 6: Properties for the bubble in Ja 2.15 superheated liquid case.

Three test cases were run: $N = 64^3, 128^3, 192^3$ with respective time step sizes of $t = 1 \times 10^{-4} s$, $t = 5 \times 10^{-5} s$ and $t = 3.33 \times 10^{-5} s$. The time evolution of the bubble radius in the simulations is compared to the theoretical value in Fig. 13. Results obtained for the bubble radius R at $t = 2 t_0$ and $t = 4 t_0$ are presented in Table 7, with the relative errors compared to respective theoretical values.

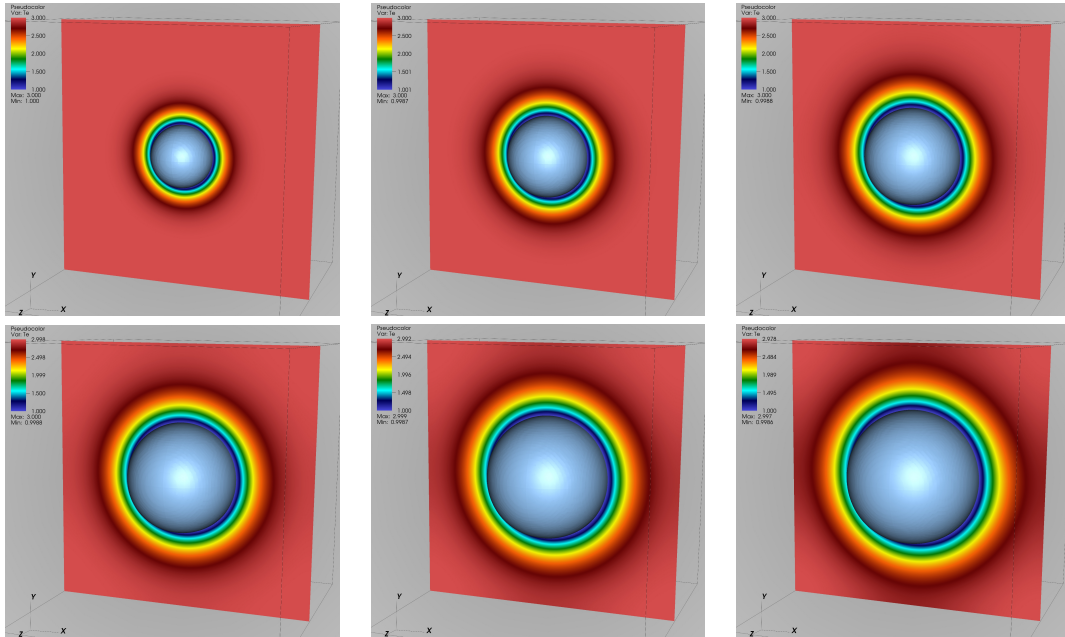


Figure 12: A time sequence presentation of a vapor bubble in superheated liquid, taken at 0.4s intervals starting at $t = 0$ s at the top left. The blue surface indicates the $c = 0.5$ iso-surface, which is a representation of the interface. The colour scale shows temperature at a mid-plane through the bubble.

Grid points	$R(2t_0)$ [m]	Rel. error $R(2t_0)$ %	$R(4t_0)$ [m]	Rel. error $R(4t_0)$ %
$N = 64^3$	1.347×10^{-3}	4.77	1.863×10^{-3}	6.87
$N = 128^3$	1.395×10^{-3}	1.37	1.993×10^{-3}	0.36
$N = 192^3$	1.411×10^{-3}	0.21	2.006×10^{-3}	0.29

Table 7: Relative errors in the Ja 2.15 case for the bubble radius R at $t = 2t_0$ and $t = 4t_0$. Theoretical solution: $R(2t_0) = 1.414 \times 10^{-3}$ and $R(4t_0) = 2.00 \times 10^{-3}$.

5. Conclusion

The Direct Numerical Simulation of flows with phase change was studied via a geometric PLIC VOF scheme. The discontinuity in velocity across the interface due to phase change poses a particular challenge to conventional VOF schemes that rely on a smooth, divergence-free velocity field. In this work a novel two-step VOF advection method was proposed. In the first step, the interface is advected with a divergence free liquid velocity, which is obtained from the solution of a Poisson problem that decomposes the one-fluid velocity into a liquid velocity and a phase-change component. In the second step, the phase change component is accounted for with an explicit interface shift in the local normal direction.

The geometric treatment of VOF advection is applied consistently to the thermal energy advection term. The thermal energy diffusion term is solved implicitly, using a similar technique to Sato and Ničeno [24], where an asymmetric stencil is used to apply the interface temperature directly.

The method was implemented in PARIS (with full parallel computation capability) and was tested using three benchmark test cases. The two-step VOF advection method was tested on a two-dimensional evaporating droplet. The results on three different grids were indistinguish-

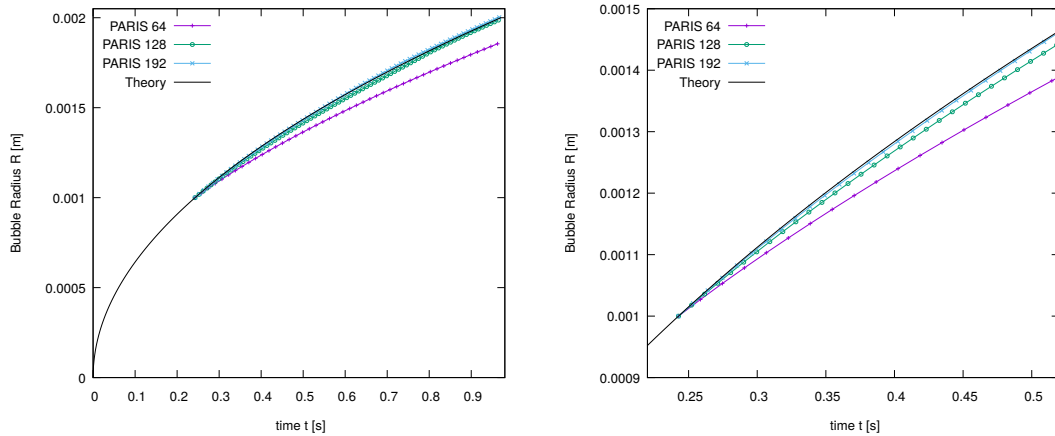


Figure 13: The time evolution of the bubble radius for a bubble at saturated temperature in a superheated liquid for Ja 2.15.

able and showed excellent agreement to the $1/t^{1/2}$ evolution of the volume. The one-dimensional Stefan problem was solved to excellent accuracy. Two cases of a three dimensional bubble at saturated temperature in a superheated liquid was simulated at three different grid resolutions, with respective Jacob numbers $Ja = 0.5$ and $Ja = 2.15$. It was found that the spherical shape was preserved and the radius evolution was predicted with good accuracy, especially in the high resolution case where the temperature gradient in the thermal boundary layer could be captured with greater accuracy than in the cases with lower resolution.

Acknowledgements

This work is based on research supported by the National Research Foundation of South Africa (Grant Numbers: 89916). The opinions, findings and conclusions or recommendations expressed is that of the authors alone, and the NRF accepts no liability whatsoever in this regard. The authors thank Roy Horwitz for the creation of a numerical solver for the theoretical bubble radius and temperature profile in the flashing saturated bubble test case that was used to validate the numerical results. The authors acknowledge the Centre for High Performance Computing (CHPC), South Africa, for providing computational resources to this research project.

Screen shots presented in this work are from VISIT [30].

- [1] G. Tryggvason, A. Esmaceli, N. Al-Rawahi, Direct numerical simulations of flows with phase change, *Computers & structures* 83 (6) (2005) 445–453. doi:10.1016/j.compstruc.2004.05.021.
- [2] C. R. Kharangate, I. Mudawar, [Review of computational studies on boiling and condensation](#), *International Journal of Heat and Mass Transfer* 108, Part A (2017) 1164–1196. doi:10.1016/j.ijheatmasstransfer.2016.12.065. URL [//www.sciencedirect.com/science/article/pii/S0017931016320051](http://www.sciencedirect.com/science/article/pii/S0017931016320051)
- [3] C. W. Hirt, B. D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, *Journal of Computational Physics*, vol. 39, Jan. 1981, p. 201-225. Research sponsored by the U.S. Department of Energy and Electric Power Research Institute; 39 (1981) 201–225. doi:10.1016/0021-9991(81)90145-5.

- [4] S. W. J. Welch, J. Wilson, A volume of fluid based method for fluid flows with phase change, *Journal of Computational Physics*, Volume 160, Issue 2, pp. 662-682 (2000). 160 (2000) 662–682. doi:10.1006/jcph.2000.6481.
- [5] D. Youngs, *Numerical Methods for Fluid Dynamics*, Vol. 24, Academic Press, 1982, Ch. Time-Dependent Multi-material Flow with Large Fluid Distortion, pp. 273–285.
- [6] S. W. J. Welch, T. Rachidi, Numerical computation of film boiling including conjugate heat transfer, *Numerical Heat Transfer: Part B: Fundamentals*, vol. 42, issue 1, pp. 35-53 42 (2002) 35–53. doi:10.1080/10407790190053824.
- [7] D. K. Agarwal, S. W. J. Welch, G. Biswas, F. Durst, Planar simulation of bubble growth in film boiling in near-critical water using a variant of the vof method, *Journal of heat transfer* 126 (3) (2004) 329–338. doi:10.1115/1.1737779.
- [8] J. Schlottke, B. Weigand, Direct numerical simulation of evaporating droplets, *Journal of Computational Physics*, Volume 227, Issue 10, p. 5215-5237. 227 (2008) 5215–5237. doi:10.1016/j.jcp.2008.01.042.
- [9] C. Kunkelmann, Numerical modeling and investigation of boiling phenomena, Ph.D. thesis, Technische Universität (2011).
- [10] G. Guédon, Two-phase heat and mass transfer modeling: Flexible numerical methods for energy engineering analyses, Ph.D. thesis (2013).
- [11] G. D. Weymouth, D. K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on cartesian-grids, *Journal of Computational Physics*, Volume 229, Issue 8, p. 2853-2865. 229 (2010) 2853–2865. doi:10.1016/j.jcp.2009.12.018.
- [12] L. C. 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, Sorbonne Université, thesis available at <http://hdl.handle.net/11427/27898> and <https://tel.archives-ouvertes.fr/tel-01875719> (2018).
- [13] G. Tryggvason, R. Scardovelli, S. Zaleski, *Direct Numerical Simulations of Gas-Liquid Multiphase Flows*, Cambridge University Press, 2011.
URL <https://books.google.co.za/books?id=nY5bjSYq-AEC>
- [14] F. H. Harlow, J. E. Welch, Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, *Physics of Fluids*, Volume 8, Issue 12, p.2182-2189 8 (1965) 2182–2189. doi:10.1063/1.1761178.
- [15] A. J. Chorin, On the convergence of discrete approximations to the navier-stokes equations, *Mathematics of computation* 23 (106) (1969) 341–353. doi:10.1090/S0025-5718-1969-0242393-5.
- [16] W. Aniszewski, T. Arrufat, M. Ciraiesi-Esposito, S. Dabiri, D. Fuster, Y. Ling, J. Lu, L. Malan, S. Pal, R. Scardovelli, G. Tryggvason, P. Yecko, S. Zaleski, *Parallel, robust, interface simulator (paris)*, working paper or preprint, available at <https://hal.sorbonne-universite.fr/hal-02112617> (Apr. 2019).
URL <https://hal.sorbonne-universite.fr/hal-02112617>
- [17] E. Aulisa, S. Manservigi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry, *Journal of Computational Physics* 225 (2) (2007) 2301–2319. doi:10.1016/j.jcp.2007.03.015.

- [18] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, *Journal of Computational Physics* 164 (1) (2000) 228–237. doi:10.1006/jcph.2000.6567.
- [19] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split eulerian-lagrangian advection, *International Journal For Numerical Methods In Fluids* 41 (2003) 251–274.
- [20] J. Li, Calcul d’interface affine par morceaux, *Comptes rendus de l’Académie des sciences. Série II, Mécanique, physique, chimie, astronomie* 320 (8) (1995) 391–396.
- [21] S. Tanguy, T. Ménard, A. Berlemont, A level set method for vaporizing two-phase flows, *Journal of Computational Physics* 221 (2) (2007) 837–853. doi:10.1016/j.jcp.2006.07.003.
- [22] 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*, working paper or preprint, available at <https://arxiv.org/abs/1811.12327> (2018). arXiv:1811.12327. URL <https://arxiv.org/abs/1811.12327>
- [23] C.-W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems, *Siam Review* 51 (2009) 82–126.
- [24] Y. Sato, B. Ničeno, A sharp-interface phase change model for a mass-conservative interface tracking method, *Journal of Computational Physics* 249 (2013) 127–161. doi:10.1016/j.jcp.2013.04.035.
- [25] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible navier stokes equations with phase change, *Journal of Computational Physics* 222 (2007) 536–555. doi:10.1016/j.jcp.2006.07.035.
- [26] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, C. Colin, Benchmarks and numerical methods for the simulation of boiling flows, *Journal of Computational Physics* 264 (2014) 1–22.
- [27] D. Q. Nguyen, R. P. Fedkiw, M. Kang, A boundary condition capturing method for incompressible flame discontinuities, *Journal of Computational Physics* 172 (1) (2001) 71–98. doi:10.1006/jcph.2001.6812.
- [28] E. Gabriel, G. E. Fagg, G. Bosilca, T. Angskun, J. J. Dongarra, J. M. Squyres, V. Sahay, P. Kambadur, B. Barrett, A. Lumsdaine, R. H. Castain, D. J. Daniel, R. L. Graham, T. S. Woodall, Open mpi: Goals, concept, and design of a next generation mpi implementation, in: *Proceedings, 11th European PVM/MPI Users’ Group Meeting, Budapest, Hungary, 2004*, pp. 97–104.
- [29] L. E. Scriven, On the dynamics of phase growth, *Chemical engineering science* 10 (1-2) (1959) 1–13. doi:10.1016/0009-2509(59)80019-1.
- [30] H. Childs, E. Brugger, B. Whitlock, J. Meredith, S. Ahern, D. Pugmire, K. Biagas, M. Miller, C. Harrison, G. H. Weber, H. Krishnan, T. Fogal, A. Sanderson, C. Garth, E. W. Bethel, D. Camp, O. Rübel, M. Durant, J. M. Favre, P. Navrátil, Visit: An end-user tool for visualizing and analyzing very large data, in: *High Performance Visualization—Enabling Extreme-Scale Scientific Insight, 2012*, pp. 357–372.