Available online at www.sciencedirect.com
ScienceDirect
SoftwareX 5 (2016) 216–226
SoftwareX
www.elsevier.com/locate/softx

interThermalPhaseChangeFoam—A framework for two-phase flow simulations with thermally driven phase change

Mahdi Nabil, Alexander S. Rattner*

Department of Mechanical and Nuclear Engineering, The Pennsylvania State University, University Park, PA, 16802, United States

Received 7 April 2016; received in revised form 19 August 2016; accepted 1 October 2016

Abstract

The volume-of-fluid (VOF) approach is a mature technique for simulating two-phase flows. However, VOF simulation of phase-change heat transfer is still in its infancy. Multiple closure formulations have been proposed in the literature, each suited to different applications. While these have enabled significant research advances, few implementations are publicly available, actively maintained, or inter-operable. Here, a VOF solver is presented (interThermalPhaseChangeFoam), which incorporates an extensible framework for phase-change heat transfer modeling, enabling simulation of diverse phenomena in a single environment. The solver employs object oriented OpenFOAM library features, including Run-Time-Type-Identification to enable rapid implementation and run-time selection of phase change and surface tension force models. The solver is packaged with multiple phase change and surface tension closure models, adapted and refined from earlier studies. This code has previously been applied to study wavy film condensation, Taylor flow evaporation, nucleate boiling, and dropwise condensation. Tutorial cases are provided for simulation of horizontal film condensation, smooth and wavy falling film condensation, nucleate boiling, and bubble condensation. Validation and grid sensitivity studies, interfacial transport models, effects of spurious currents from surface tension models, effects of artificial heat transfer due to numerical factors, and parallel scaling performance are described in detail in the Supplemental Material (see Appendix A). By incorporating the framework and demonstration cases into a single environment, users can rapidly apply the solver to study phase-change processes of interest.

Keywords: Phase change; Condensation; Evaporation; Volume of fluid

Code metadata

Current code version	2.4.0.5
Permanent link to code/repository used of this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-16-00038
Legal Code License	GNU GPL V3
Code versioning system used	git
Software code languages, tools, and services used	C++
Compilation requirements, operating environments & dependencies	OpenFOAM 2.4.0 library, Linux based environment Swak4Foam extension (optional, used in some tutorial cases for initialization and boundary conditions)
If available Link to developer documentation/manual	https://github.com/MahdiNabil/CFD-PC/blob/master/README.md
Support email for questions	Alex.Rattner@gmail.com

1. Motivation and significance

Liquid–vapor phase change plays a key role in many energy-intensive processes, and is often a bottleneck for system efficiency and intensity. To address such limitations,

* Correspondence to: 236A Reber Building, University Park, PA, 16802, United States.
E-mail address: Alex.Rattner@psu.edu (A.S. Rattner).

http://dx.doi.org/10.1016/j.softx.2016.10.002

increasingly sophisticated designs are being developed for heat transfer equipment. However, due to the complex nature of two-phase transport and great ranges of scales, exact analytical models are only feasible for the simplest flows (e.g., smooth falling film evaporation [1], spherical bubble condensation [2]). The phase-change heat transfer community thus relies heavily on empirical transport correlations developed from challenging experiments and simplified models with limited ranges of validity [3]. There is a need for robust simulation tools to analyze phase change heat transfer processes in complex configurations.

A number of simulation approaches have been employed in the literature to study phase change heat transfer, including: Lagrangian tracking of individual particles (e.g., droplets and bubbles) [4–9], two-fluid Eulerian averaging of dispersed features [10–14], and interface capturing and tracking techniques [15–18]. The present effort focuses on the volume-of-fluid (VOF) interface capturing technique, which solves a single consistent set of mass, momentum, and energy equations for the whole domain. Interface dynamics are resolved using a phase fraction field (α_1) that ranges from 0 in the gas phase to 1 in the liquid phase, and is advected with the velocity field. Material properties are determined at each point, weighted by the local phase fraction. Relatively mature implementations of the Volume-of-Fluid (VOF) formulation have been incorporated into commercial software [19–21] to study adiabatic flows (i.e., no heat transfer). However, phase-change heat transfer phenomena have essentially only been simulated in research and academic codes. These investigations have enabled significant advances in understanding phase change heat transfer, but few solvers are publicly available to support routine use.

In most VOF phase-change studies, investigators have modified existing adiabatic two-phase flow solvers, and incorporated a thermal energy transport equation and coupled phase-change source terms in the governing equations. A number of closure formulations have been proposed for these source terms. One approach has been to determine the volumetric phase change heating rate in each cell using empirical rate parameters [15,22–27]. For example, Yang et al. [15] adopted the following form for condensation and evaporation cases, with rate parameters r_L and r_V.

$$\dot{q}_{pc} = \begin{cases} r_L \alpha_1 \rho_L \dfrac{T - T_{sat}}{T_{sat}} & T \geq T_{sat} \\ r_V \left(1 - \alpha_1\right) \rho_V \dfrac{T - T_{sat}}{T_{sat}} & T < T_{sat}. \end{cases} \quad (1)$$

An advantage of this approach is that it is conceptually simple, and that the source term field is relatively smooth. However, this model applies phase change effects everywhere in the domain, not just at the liquid–vapor interface. Additionally, the rate parameters r_L and r_V represent the thermal time constant of mesh cells, and must be tuned for different studies and meshes [28].

Some investigations have adapted analytical heat transfer correlations to predict local phase change rates in mesh cells [25,29]. This approach can yield accurate overall results, but most correlations do not account for local variations in heat transfer. Additionally, it may be necessary to select and validate different correlations for distinct phase change processes (e.g., bubble condensation, film evaporation).

Other studies have determined phase change rates based on sub-continuum-scale interfacial resistance in mesh cells [16,30]. This approach is valid in the limit of extremely fine grid resolution near the interface, but for many practical simulations, continuum scale convective resistance in mesh cells is dominant (i.e. $\Delta/k \gg R''_{int}$), where Δ is the mesh cell size, k is the fluid thermal conductivity, and R''_{int} represents interfacial thermal resistance.

An alternate formulation is to specify the phase change source terms so that the interface containing cells recover the local saturation temperature after each time step [28,31]. This description ensures that phase change only occurs in the vicinity of the interface, and has been validated for a number of applications [28]. However, scaling the phase change rate by the simulation time scale (i.e., volumetric rate of phase change heating $\dot{q}_{pc} = \rho c_p \left(T - T_{sat}\right) / \Delta t$, for fluid density ρ, specific heat c_p, temperature T, fluid saturation temperature T_{sat}, and time step size Δt) can lead to unstable behavior in certain cases.

The most rigorous approaches directly evaluate liquid- and vapor-phase temperature gradients at the interface to determine phase change rates [17,18,32–37]. However, this requires geometric reconstruction (surface triangulation) of the interface to determine heat transfer area in each mesh cell. This process can be complex on unstructured meshes, increases computational cost, and is not available in some popular two-phase flow solvers (e.g., interFoam [21]).

These phase change closure models are each best suited to different applications, but most are compatible in a general framework that applies coupled source terms to the governing transport equations for mass, phase-fraction, and energy. However, in almost every prior study, a new extended solver was built, resulting in a great deal of repeated software development and validation work. Investigators wishing to evaluate different published formulations may need to try multiple solvers, many of which are not actively maintained.

Here, an open source extensible phase-change solver is developed: interThermalPhaseChangeFoam. This package enables the run-time selection of different phase change formulations, including those of [15,28,38]. New phase-change models can be rapidly implemented and refined by extending the C++ virtual class: thermalPhaseChangeModel, enabling users to build on previous research advances. The solver is based on the adiabatic two phase flow code interFoam [21], and provides access to extensive OpenFOAM functionality (MPI parallelization, turbulence models, discretization schemes, linear solvers). This software thus enables simulation of diverse phase change processes in a single environment.

A preliminary version of the solver algorithm has been validated for canonical phase change configurations, including the Stefan problem (horizontal film condensation) and the Nusselt problem (smooth falling film condensation) [28]. It has also been applied to study the more complex phenomena of wavy falling film condensation (Fig. 1a) [28], which had previously only been characterized empirically. Recently, this solver was

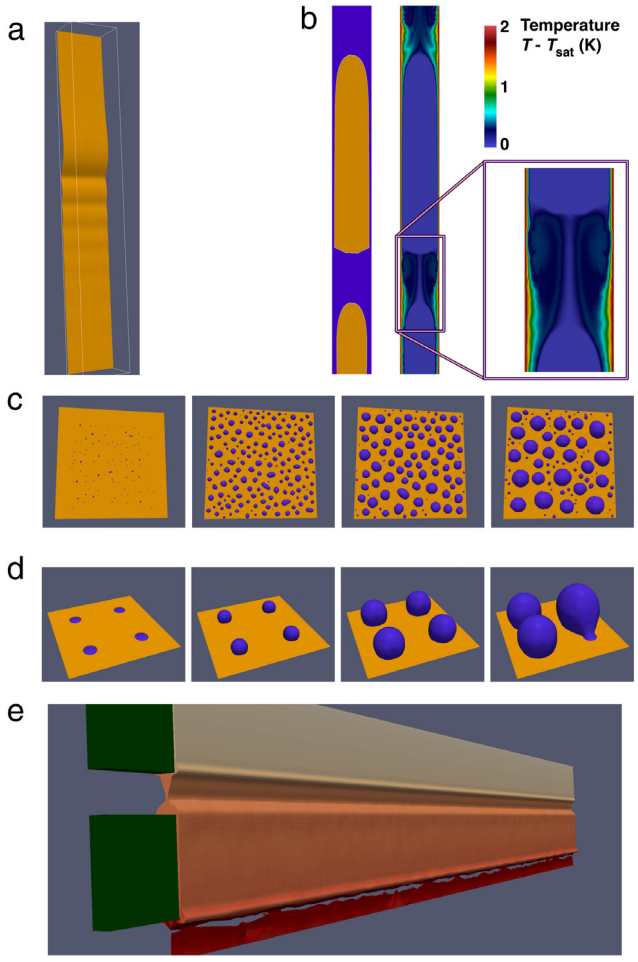

Fig. 1. Representative simulation results using interThermalPhaseChangeFoam. a. Wavy falling film condensation, b. Wake region heat transfer enhancement in Taylor flow evaporation, c. Dropwise condensation on a subcooled surface, d. Nucleate boiling on a structured surface, e. Vapor absorption into liquid film flowing over rectangular tubes (colored by liquid concentration, 768-way parallel study). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

employed to study two-phase Taylor-flow evaporation, and enabled quantification of the elongated bubble wake-region heat transfer enhancement effect (Fig. 1b) [38]. In ongoing efforts, this software is being applied to simulate dropwise condensation (Fig. 1c), interactions between bubbles in nucleate boiling (Fig. 1d), and falling film absorption over banks of high-aspect ratio tubes (Fig. 1e). The objective of this publication is to document the solver and provide validation and verification results to inform prospective users. By providing the capability to model a variety of phase-change processes in one environment, it is hoped that interThermalPhaseChangeFoam can be a useful resource for investigating two-phase transport processes and designing heat transfer equipment.

2. Software description

2.1. Governing equations and implementation

The thermal phase-change solver is extended from the adiabatic two-phase flow solver interFoam [21], which is described in detail in [39]. The code solves conservation equations for mass (via a pressure Poisson equation), momentum, thermal energy, and the phase-fraction (α_1). The mass conservation equation assumes uniform density in each phase (liquid and vapor), and applies a source term to represent the dilatation rate ($\dot{v}_{\mathrm{pc}}$, the volume source rate per unit volume) due to liquid–vapor phase change. This source term field is evaluated with the *thermalPhaseChangeModel* framework described in the following section.

$$\nabla \cdot u = \dot{v}_{pc}. \tag{2}$$

The momentum equation (Eq. (3)) is formulated accounting for varying fluid density and viscosity, as in the VOF phase change simulations of [16,24,30,35]. A generic viscous stress implementation is employed (τ_{ij}), which can apply laminar behavior or one of the RANS and LES turbulence models implemented in OpenFOAM. In the base interFoam code, the pressure is corrected for hydrostatic variations ($p' = p - \rho g h_{\mathrm{eff}}$). Here, a run-time selectable switch is provided to the user (c_{hsp}) to employ either corrected pressure, or static pressure, which is more convenient for cyclic domain simulations.

Surface tension is applied as a volumetric force in the vicinity of the interface (f_σ). The base interFoam solver employs a volumetric surface-tension force model based on that of Brackbill et al. [40], which operates on volumetric field data (α_1). Alternate implementations have been demonstrated to produce smaller spurious currents for specific two-phase flow configurations [41,42]. For example, Raeini et al. [42] proposed an approach that filters capillary forces parallel to the interface as well as small-amplitude capillary waves to dampen non-physical velocities. Geometric interface reconstruction based schemes for f_σ have been shown to be more accurate [42], but are more complicated and computationally expensive [40,43,44]. The current solver does not support interface reconstruction, which limits accuracy for cases with dominant surface tension effects [45]. The developed solver incorporates an extensible framework for surface tension force models that can be selected at runtime. The current implementation includes the models of Brackbill et al. [40] Raeini et al. [42], and Lafaurie et al. [46]. An analysis of the effect of spurious currents due to the surface-tension force model implementation on heat transfer is presented in the Supplemental Material Section 5. It demonstrates that the default implementation based on [40] can yield incorrect heat transfer rates in cases where surface tension forces are strong and fluid Prandtl numbers are high ($\mathrm{Pr} = \mu c_p/k$), causing convection due to spurious currents to be significant relative to other modes of heat transfer. For such cases, the model based on [42] yields more accurate results.

$$\frac{\partial(\rho u)}{\partial t} + \nabla \cdot (\rho u u) = -\nabla p' + \nabla \cdot \tau + c_{\mathrm{hsp}} \rho g + f_\sigma. \tag{3}$$

In the solver implementation, the divergence of the momentum equation (Eq. (3)) is evaluated in a semi-discretized fashion, and the mass conservation constraint (Eq. (2)) is applied to yield the pressure Poisson equation. The matrix equation coefficients ($1/A_{\mathrm{D}}$) are obtained from the momentum equation (Eq. (3)) as

described in [28,47].

$$\nabla \cdot \left(\frac{1}{A_D} \nabla p' \right) = \nabla \cdot u - \dot{v}_{pc}. \tag{4}$$

The pressure equation is solved and the velocity field is corrected in an alternating fashion during each time step following the OpenFOAM PISO-semi-implicit method for pressure-linked equations (PIMPLE) algorithm [21,48].

A thermal energy transport equation is employed to solve for the enthalpy (i), neglecting pressure transport and viscous dissipation effects because conduction and convection heat fluxes are expected to be dominant in this phase-change process. This approximation has been successfully applied in prior phase change simulations, including those of [16,28,32]. Effective fluid thermal conductivity (k_{eff}) is evaluated assuming either molecular diffusion, or an effective diffusivity obtained from the user-selected turbulence model. The phase change heating rate ($\dot{q}_{\text{pc}}$) is determined using a run-time selected model from the phase-change framework developed in this effort. Analyses of effects of artificial heat transfer due to numerical factors are presented in Supplemental Material Section 6.

$$\frac{\partial(\rho i)}{\partial t} + \nabla \cdot (\rho u i) = \nabla \cdot (k_{\text{eff}} \nabla T) - \dot{q}_{pc}. \tag{5}$$

Finally, a hyperbolic transport equation is solved for the phase-fraction field $\alpha_1 \in [0, 1]$. The phase-fraction field varies from 0 in the vapor phase to 1 in the liquid phase. The modified velocity field (u^*) includes corrections based on the gradient of the phase field to compress the interface and counteract numerical diffusion (see [39,49,50]). The phase fraction source term ($\dot{\alpha}_{1,\text{pc}}$) is determined using the phase-change closure model.

$$\frac{\partial \alpha_1}{\partial t} + \nabla \cdot (u^* \alpha_1) = \dot{\alpha}_{1,pc}. \tag{6}$$

By default, fluid material properties μ, ρ, and k are evaluated as arithmetic phase-fraction (α_1) weighted averages of the liquid and vapor phases in each mesh cell. The fluid specific heat is evaluated with a mass weighted average ($\rho^{-1}(\alpha_1 \rho_L c_{p,L} + (1-\alpha_1)\rho_V c_{p,V})$). Arithmetic averaging of the transport properties (μ and k) has been performed in most prior studies, but can cause errors if the mesh is coarse near the interface [49]. To address this issue, a runtime-selectable switch is provided, which applies more accurate blending of the liquid and vapor properties based on the relative orientation of the interface to cell faces (i.e., whether liquid and vapor-phase transport resistances act in serial or parallel in a cell). This functionality is discussed in the Supplemental Material (Section 1). The interface is typically ~3 cells thick in these VOF simulations; therefore, discontinuities in fluid properties and flow fields (e.g., u, p', T) are not perfectly sharp. Still, accurate results can be obtained for many flows with sufficient mesh resolution, as demonstrated for the temperature field in film condensation (see Supplemental Material, Section 2). In the present implementation, dynamic adaptive mesh refinement (AMR) is not employed, but could help address this challenge.

1. Initialize simulation data and phase change model
2. **WHILE** $t < t_{\text{end}}$ **DO**
   1. Update Δt for stability (CFL and Fourier conditions)
   2. Update fluid and turbulence properties
   3. Update phase change model ($\dot{q}_{\text{pc}}$, $\dot{v}_{\text{pc}}$, $\dot{\alpha}_{\text{pc}}$)
   4. Phase equation (α_1) sub-cycle
   5. **DO** Pressure-velocity correction loop
      1. Form u equation
      2. PISO algorithm
         1. Obtain and correct face volumetric fluxes
         2. Solve p-Poisson equation (mass conservation)
         3. Correct u
   6. **LOOP**
   7. Update $i(T)$
   8. **DO** Energy Loop
      1. Solve i equation
      2. Update $T(i)$
   9. **LOOP**
3. **LOOP**

Fig. 2. Phase change flow solver algorithm summary.

The governing equations are discretized with the finite volume approach, and are solved at each simulation time step following the algorithm outlined in Fig. 2. The advection terms in the momentum and thermal energy equations ($\nabla \cdot (\rho u u)$ and $\nabla \cdot (\rho u i)$, respectively) are evaluated using the mass fluxes across each cell face ($(\rho u)_{\text{f}}$) obtained from the phase-fraction equation for each global time step. This ensures consistency between the solved α_1, u, T, and i fields.

2.2. Phase change framework

A framework is developed to support simple implementation of phase change closure models. All models inherit from a base virtual c++ class: thermalPhaseChangeModel. This parent class provides references to relevant data fields and parameters, including transport properties, temperature (T), phase fraction (α_1), saturation temperature (T_{sat}), and enthalpy of phase change (i_{LV}). Additionally, the base class provides default implementations for the dilatation rate ($\dot{v}_{\text{pc}}$) and phase fraction source term ($\dot{\alpha}_{1,\text{pc}}$) fields, as defined in most prior studies [23,28,31].

$$\dot{v}_{\text{pc}} = \frac{\dot{q}_{\text{pc}}}{i_{\text{LV}}} \left(\frac{1}{\rho_{\text{V}}} - \frac{1}{\rho_{\text{L}}} \right) \tag{7}$$

$$\dot{\alpha}_{1,\text{pc}} = -\frac{\dot{q}_{\text{pc}}}{\rho i_{\text{LV}}}. \tag{8}$$

Each new model implements a member function for the phase-change heat source term ($\dot{q}_{\text{pc}}$), and may overload the default implementations for $\dot{v}_{\text{pc}}$ and $\dot{\alpha}_{1,\text{pc}}$.

This extensible approach enables end users to select the most appropriate phase-change model for a given problem, much like selection from various turbulence models in commercial CFD codes. The provided tutorial and validation/mesh independence studies (see Supplemental Material Section 3) offer guidelines for phase change model selection and parameter values. A set of phase change models is provided with the solver, including:

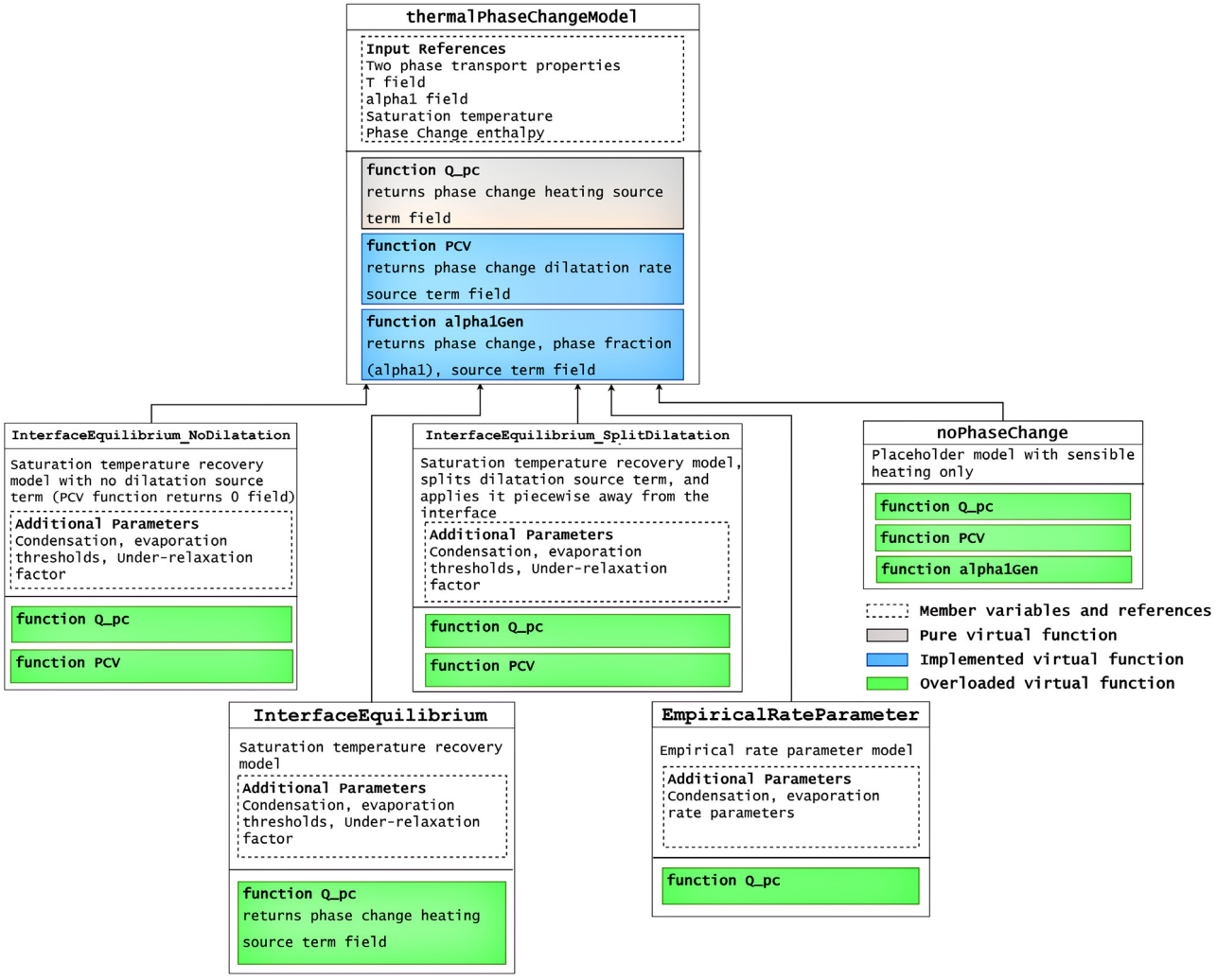

Fig. 3. Block diagram of the phase-change model framework and implemented models.

- **InterfaceEquilibrium** — An updated version of the model of Rattner and Garimella [28] that specifies the phase change heat sources so that interface cells recover the saturation temperature at each time step (i.e., thermal equilibrium on the interface). This model performs a graph scan over mesh cells, and applies phase change on the two-cell thick interface layer about user-specified threshold values of α_1. Different high and low threshold values for condensation and evaporation, respectively, can be specified, which reduce numerical smearing of the interface. Under-relaxation of the phase change rate is supported, which can improve numerical stability.
- **InterfaceEquilibrium_SplitDilatation** — A modified version of the above model, which splits the liquid and vapor portions of $\dot{v}_{\text{pc}}$, and applies them on the respective sides of the interface [38]. This approach yields better conservation of the two phases, and reduces smearing of the interface during evaporation.
- **InterfaceEquilibrium_NoDilatation** — A modified version of InterfaceEquilibrium that sets $\dot{v}_{\text{pc}}$ to 0. This model can yield accurate results with relaxed CFL time step constraints for cases without significant vapor-phase momentum effects (e.g., falling film condensation in a quiescent vapor medium).
- **EmpiricalRateParameter** — An implementation of the empirical rate parameter model of Yang et al. [15].
- **noPhaseChange** — A placeholder model with only sensible heating. This can be useful for debugging cases (e.g., determining if an instability is caused by phase change effects).

Additional details and discussion of prior validation studies of these phase change models are included in Supplemental Material Section 3. Additional phase-change closure models are currently under development to account for interfacial thermal resistance, dropwise condensation, and bubble nucleation. A UML block diagram of the phase-change-model framework is presented in Fig. 3.

As a demonstration of the phase change framework, the implementation for the EmpiricalRateParameter model is provided below (based on Eq. (1)). The constructor is omitted for brevity, and default implementations for $\dot{v}_{\text{pc}}$ and $\dot{\alpha}_{1,\text{pc}}$ are employed.

```
//Evaluates phase change heat source term field
void Foam::thermalPhaseChangeModels::EmpiricalRateParameter::calcQ_pc()
{
    const dimensionedScalar& rhol = twoPhaseProperties_.rho1();
    const dimensionedScalar& rhov = twoPhaseProperties_.rho2();

    Q_pc_ =
        pos(T_ - T_sat_)*h_lv_*rl*alpha1_*rhol*((T_ - T_sat_)/T_sat_)
      + neg(T_ - T_sat_)*h_lv_*rv*(1.0  - alpha1_)*rhov*((T_ - T_sat_)/T_sat_);
}

//Updates model parameters from user specified input file
bool Foam::thermalPhaseChangeModels:: EmpiricalRateParameter::
read(const dictionary& thermalPhaseChangeProperties)
{
   thermalPhaseChangeModel::read(thermalPhaseChangeProperties);
   thermalPhaseChangeProperties_.lookup("rl") >> rl;
   thermalPhaseChangeProperties_.lookup("rv") >> rv;
   return true;
}
```

Using this framework, phase change models can be rapidly implemented, enabling simulation of diverse phase change processes in one environment.

2.3. Use of object oriented program and OpenFOAM library features in software development

This solver employs a number of object-oriented programming and OpenFOAM library features to address challenges in modeling phase-change heat transfer (and continuum mechanics in general). OpenFOAM provides extensive operator overloading implementations to permit intuitive definitions of continuum mechanics models and governing equations. At compile-time the expression "**a**∗**b**" would be evaluated as scalar multiplication (for scalar **a**, **b**), element-wise multiplication (for scalar fields **a**, **b**), a vector outer product (for vectors/vector fields **a**, **b**), or other appropriate operations for different types of **a** and **b**. This functionality is described in [47], and, as an example, is employed here in the definition of the thermal energy transport sparse matrix equation:

```
fvScalarMatrix EEqn
(
      fvm::ddt(rho, H)
    + fvm::div(rhoPhi, H)
    - fvc::laplacian(kEff, T)
    - RelaxFactor*( fvm::laplacian(alphaEffRho, H)
                  - fvc::laplacian(alphaEffRho, H) )
    + phaseChangeModel->Q_pc()
);
```

Here, `fvm` operations are automatically applied as implicit terms in the matrix equation (left-hand side), and `fvc` operations and other entries are applied explicitly (right-hand side).

The OpenFOAM library employs the design pattern of run-time type identification and virtual function dispatch to support generalized boundary conditions, turbulence models, and viscosity models. This approach allows addition of new run-time selectable derived class implementations without any modification of the base class or application code. This also facilitates software maintainability, as the derived model classes can be individually revised, independent of other elements of the code base. Here, this design pattern is employed to enable rapid development and flexible run-time selection of phase change and surface tension force models. The virtual base classes for these two sets of models (`thermalPhaseChangeModel` and `surfaceTensionForceModel`) include definitions and functionality common to all derived classes. For example, the latent heat of phase change (h_{LV}) is defined as a protected member variable in `thermalPhaseChangeModel` so that it is automatically available to all derived model classes, which generally need h_{LV} to evaluate phase change source terms. Default (overloadable) virtual method implementations are also provided in the base class in some cases (e.g., for $\dot{v}_{pc}$ and $\dot{\alpha}_{1,pc}$), when common formulations are used in most models, reducing code repetition.

Each model base class defines a hash table of pointers to derived class constructors. At runtime, each derived model class adds an entry to the hash table so that it can be selected and evaluated. The `static New` method in the base class selects and executes a constructor from the hash table (here, found using the run time-specified derived class model name), and returns a pointer to the resulting model object. Virtual method calls for phase change and surface tension source terms then evaluate the definitions in the run-time selected model classes. This design pattern is described in detail in [51,52].

3. Illustrative examples

The packaged software includes a set of tutorial cases for: horizontal film condensation (Stefan problem), smooth falling film condensation (Nusselt problem), wavy falling film condensation, nucleate boiling (single site, axisymmetric), and rising bubble condensation in a liquid medium (axisymmetric) (Fig. 4). Automated testing scripts are included to validate tutorial output against established solutions (when possible) and earlier "known good" results. A set of grid independence studies for the Stefan case, Nusselt problem (evaporation and condensation), and bubble condensation case is included with the software package. Results and convergence data for these cases are discussed in the Supplemental Material (Section 4). In the following sections, descriptions are provided for the smooth falling film and bubble condensation tutorials.

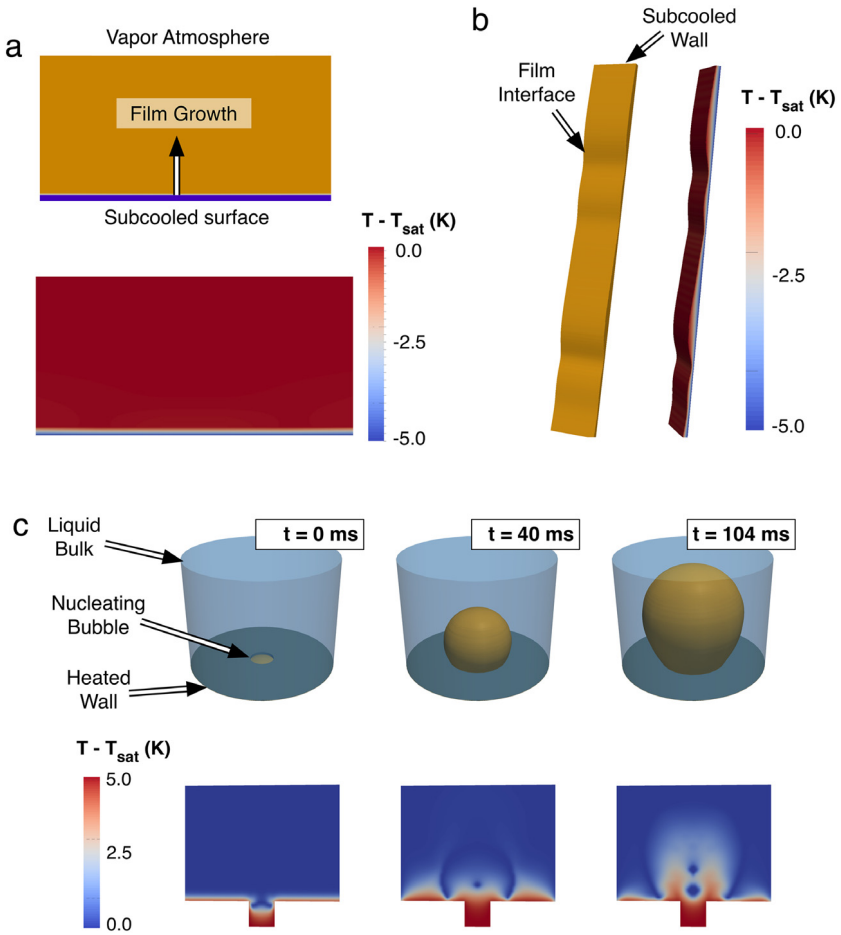

Fig. 4. Representative interface profile and temperature fields from tutorial cases for: a. Stefan problem. b. Wavy falling film condensation. c. Nucleate boiling.

3.1. Smooth falling film condensation

A tutorial case is provided with the code to simulate gravity-driven smooth falling-film condensation (Nusselt problem). Here, liquid film flows down a uniform temperature vertical subcooled surface, and condenses vapor from the quiescent atmosphere. A 2D geometry is employed with a wall height of $H = 8.0$ mm (~50 times the film thickness of 150 μm). A short inlet guide vane (250 μm long) is employed to prevent inlet film waviness. The film thickness is resolved with 25 mesh cells ($\Delta x = 6$ μm), and the length is meshed with 200 cells ($\Delta y = 40$ μm) (Fig. 5a). The case setup assumes laminar flow (Film Re ~40), and applies the InterfaceEquilibrium_SplitDilatation phase change model with no under-relaxation. Fluid properties and simulation parameters are summarized in Table 1.

Running the provided `InitScript.sh` script, generates the mesh using the OpenFOAM blockMesh utility [21], initializes the case fields (initial film profile, velocity and temperature fields), decomposes the domain, and starts a 4-way MPI parallel simulation. The simulation proceeds for 0.25 simulation seconds (ss) (Fig. 5b–c), and logs average wall heat flux values in the lower 50% of the domain every 50 time steps ($\Delta t \sim 13$ μs). This run time corresponds to approximately 4 flow cycles through the domain ($H/U_{\text{avg,film}} = 0.07$ s). A provided GNU Octave [53] script compares the average simulation heat flux after a 0.10 ss startup time ($q_{\text{w,sim}} = 16643$ W m^{-2}) with analytical results from the model of Nusselt [1]. After startup, the range of wall heat fluxes from different times is only 0.06% of the average value, indicating converged results.

$$\delta_{\text{f,an}} = \left[\frac{3\mu_{\text{L}}^2}{4\rho_{\text{L}}\left(\rho_{\text{L}} - \rho_{\text{V}}\right) g}\right]^{1/3} \text{Re}_{\text{f,sim}}^{1/3} \tag{9}$$

$$q_{\text{w,an}} = \frac{\left(T_{\text{sat}} - T_{\text{w}}\right) k_{\text{L}}}{\delta_{\text{w,an}}}. \tag{10}$$

For this case, the time-averaged analytical prediction using the simulation average Γ value is $q_{\text{w,an}} = 16929$ W m^{-2}, resulting in a 1.7% deviation. A mesh sensitivity study for this case (with condensation and evaporation) is provided in the Supplemental Material (Section 4.2), and demonstrates close agreement with analytical results for $q_{\text{w,an}}$ over a range of grid resolutions.

3.2. Bubble condensation tutorial

A second example demonstrates condensation of a small rising vapor bubble in a quiescent subcooled liquid environment. A 2D axisymmetric cylindrical domain is employed ($R = 1.5$ mm, $H = 2.0$ mm) with 105 radial cells (graded finer near the axis) and 172 vertical cells (Fig. 6a). A stationary saturated vapor bubble ($D_{\text{bub}} = 600$ μm) is initialized near the bottom

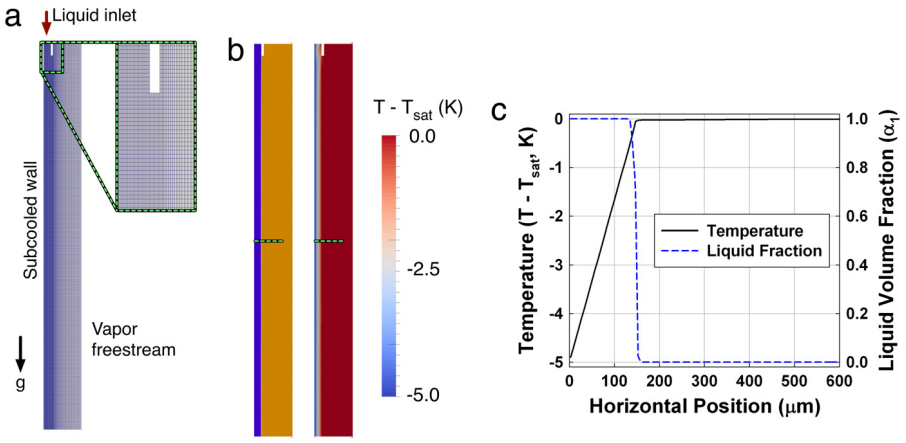

Fig. 5. a. Mesh for film condensation case, detail view of inlet region. b. Film profile at end of simulation (liquid in blue, vapor in gold) and temperature distribution (blue–red plot) at end of simulation. c. Temperature and liquid volume fraction distribution through film (along dashed lines in b), demonstrating smooth expected linear temperature profile in liquid region, and approximately 3–4 cell thick diffuse interface region. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Fluid properties and setup parameters for the smooth film condensation case.

Fluid property	Liquid value	Vapor value
Dynamic viscosity (μ, kg m^{-1} s^{-1})	5.0×10^{-4}	2.0×10^{6}
Density (ρ, kg m^{-3})	500	20
Thermal conductivity (k, W m^{-1} K^{-1})	0.50	0.02
Specific heat (c_p, kJ kg^{-1} K^{-1})	2.0	1.5
Surface tension (σ, kg s^{-2})	0.04	
Enthalpy of phase change (i_{LV}, kJ kg^{-1})	2000	
Other case parameters		
Inlet liquid channel thickness (μm)	150	
Wall subcool ($T_{\mathrm{sat}} - T_w$, K)	5	
Inlet liquid mass flux (Γ, kg m^{-1} s^{-1})	0.005	
Inlet film Reynolds number ($\mathrm{Re_f} = 4\Gamma/\mu_{\mathrm{L}}$)	40	

of the domain, and rises due to buoyancy. The surrounding liquid medium is subcooled by 1 K, causing the bubble to condense and shrink as it rises. A bubble tracking procedure is applied at the end of each simulation time step to adjust the domain and top boundary (inlet) velocities by the average vapor-phase vertical velocity. This kinematic transformation effectively transforms the simulation to the reference frame of the bubble, fixing the position of the rising bubble to the center of the domain. This approach permits use of a smaller mesh, and streamlines post-processing and analysis. The refined transport property blending models described in Supplemental Material Section 1 are also applied to improve resolution of momentum and thermal boundary layers around the bubble. Phase change is modeled using the *InterfaceEquilibrium_SplitDilatation* model. A second implementation of this case is also provided with the code to demonstrate the use of the Yang et al. [15] model. Fluid properties and case parameters are summarized in Table 2. The selected liquid viscosity is relatively high (4.5×10^{-3} kg m^{-1} s^{-1}) and the surface tension is relatively low (0.005 kg s^{-2}), reducing spurious current for this case. A high liquid thermal conductivity (1 W m^{-1} K^{-1}) is employed to yield a thick thermal boundary layer, which can be resolved on a coarse mesh for quick computation. A slightly modified version of this case is presented in Supplemental Material Section 7 with varying liquid viscosity to demonstrate applicability of this solver to cases with stronger advection transport contributions (here maximum $\mathrm{Re_{bub}} = 73$, in Supplemental Material Section 7, $\mathrm{Re_{bub}} = 2.7 - 93.8$).

Table 2
Fluid properties and setup parameters for the bubble condensation case.

Fluid property	Liquid value	Vapor value
Dynamic viscosity (μ, kg m^{-1} s^{-1})	45×10^{3}	5.0×10^{4}
Density (ρ, kg m^{-3})	900	10
Thermal conductivity (k, W m^{-1} K^{-1})	1.0	0.02
Specific heat (c_p, kJ kg^{-1} K^{-1})	2.0	2.5
Surface tension (σ, kg s^{-2})	0.05	
Liquid Prandtl number (Pr_L)	9.0	
Enthalpy of phase change (i_{LV}, kJ kg^{-1})	2000	
Saturation temperature (T_{sat}, K)	100	
Other case parameters		
Initial bubble diameter (μm)	60	
Liquid subcool ($T_{\mathrm{sat}} - T_{\mathrm{L}}$, K)	1	

The provided initialization script generates the mesh, sets the initial conditions, and initiates the solver. The case is computed

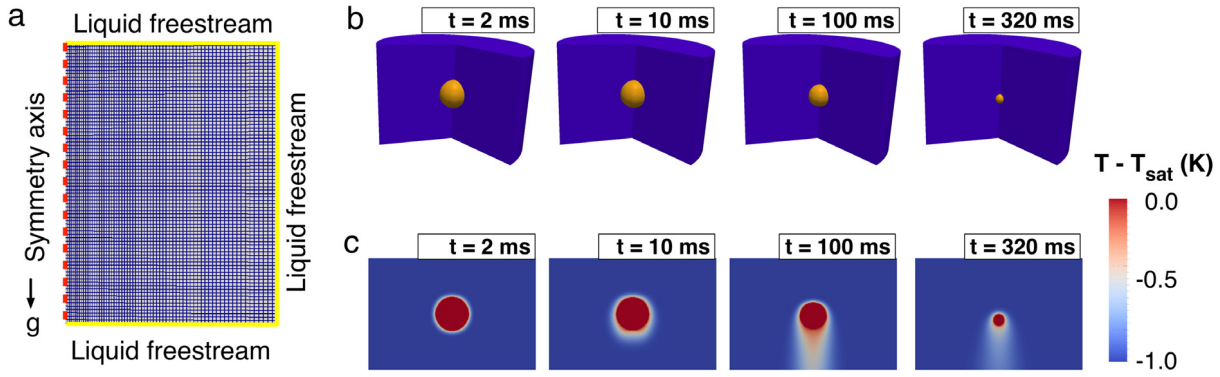

Fig. 6. a. Mesh for bubble condensation case. b. Time series of bubble rise and condensation. c. Time series of temperature distribution.

for 0.32 ss, during which time the bubble rises approximately 15 mm and its diameter reduces from 600 to 198 μm (Fig. 6b).

The simulated average heat transfer coefficient reduces from an initially high value (∼600,000 W m^{-2} K^{-1}) due to the initially thin thermal boundary layer) to ∼11.5 kW m^{-2} K^{-1} as the bubble diameter and rise velocity reduce. After ∼0.25 s, the convection coefficient begins to increase due to the reducing thermal transport length scale (D_{bub}). These results are compared with predictions using the correlation of Ranz and Marshall [54] suggested by Kim and Park [55] for $Re_{bub} = \rho_L D_{bub} U_{bub}/\mu_L < 200$ (here, $Re_{bub} < 7.3$).

$$Nu_{bub} = 2 + 0.6 Re_{bub}^{0.5} Pr_L^{0.33} \quad (11)$$

$$h_{bub} = \frac{Nu_{bub} k_L}{D_{bub}}. \quad (12)$$

After a startup time of 0.1 ss, the time-averaged ($t = 0.1 - 0.32$ ss) absolute deviation of the simulation h_{bub} from the analytical value is 8.4%, which is reasonably accurate for empirical phase change correlations (Fig. 7). As the bubble size reduces, it becomes somewhat under-resolved on the mesh. Grid sensitivity results for this case, with and without $\dot{\alpha}_{1,pc}$ and $\dot{v}_{pc}$ are discussed in the Supplemental Material (Section 4.3). Overall, heat fluxes and bubble rise velocities were found to oscillate slightly with mesh resolution (∼10% of range) over a wide span of mesh sizes.

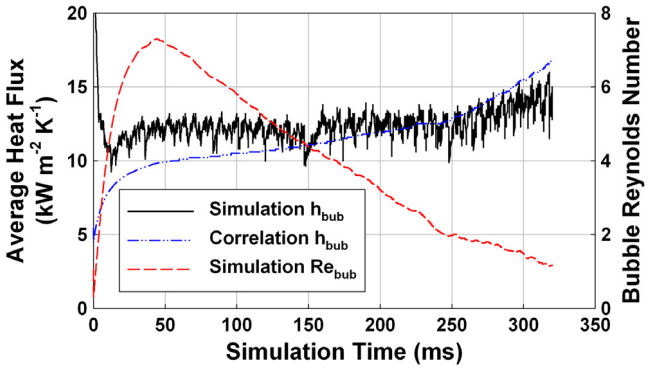

Fig. 7. Evolution of bubble Reynolds number, and comparison of simulation and correlation bubble heat transfer coefficient.

4. Impact and conclusions

interThermalPhaseChangeFoam enables simulation of a broad range of condensation, boiling, and evaporation processes in a single environment. It includes a set of independently validated phase change models, which are suited to different applications. The solver provides a framework to rapidly implement new models. To the authors' knowledge, no publically available thermal phase change VOF code supports multiple phase change formulations. Users familiar with the OpenFOAM framework can install the software, and adapt the validated tutorial cases to study new phase-change heat transfer problems in a number of hours. In contrast, developing and testing a new phase-change solver, as typically performed in prior studies, can require weeks of effort.

One limitation of this phase-change VOF solver is that it does not support geometric interface reconstruction. This limits the potential accuracy of surface tension force calculations, and can cause spurious currents. The VOF formulation conserves phase masses well, but the slightly smeared interface (typically ∼3 cells) may be unacceptable for cases with thin boundary layers near the interface. Similarly, interface area is not available in each cell, which is needed to accurately evaluate interfacial thermal resistance [18]. This is usually a minor effect, except in very small scale flows. Many of these interface-resolving limitations can be overcome with a sufficiently fine mesh near the interface. However, users must be careful, as extremely fine grids can amplify spurious currents in the OpenFOAM interFoam formulation [56]. Thus, the current solver is best suited to cases where bulk liquid- and vapor-domain transport is important, as opposed to cases with dominant interface dynamics. The provided implementations for the Raeini et al. [42] sharp surface tension force model and improved transport property blending can improve accuracy for such cases. The solver also assumes uniform densities in the two phases. This approximation is common for thermally driven phase change, but should be carefully considered for cases with large pressure or temperature variations (e.g., due to capillary pressures in very small bubbles).

The OpenFOAM library enables nearly automatic domain decomposition parallelization over MPI. However, scaling performance is somewhat limited, with throughput typically beginning to plateau with more than a few hundred processes at 10,000–50,000 cells per process (see scaling study in Supplemental Material Section 8 and [57]). Additionally, the OpenFOAM library does not support increasingly common hybrid

(MPI-OpenMP) and heterogeneous (GPU or coprocessors) parallelization methodologies. This parallel performance is sufficient for many engineering applications, but should be considered carefully when planning large-scale studies.

In earlier work, preliminary versions of this solver have been applied to perform detailed VOF simulation studies of wavy falling film phase change [28] and Taylor flow evaporation [38], which had previously been characterized primarily empirically (Fig. 1a–b). In ongoing efforts, this solver is being applied to study dropwise condensation, nucleate boiling, and vapor absorption over tube banks (Fig. 1c–e). For the dropwise condensation problem, a new phase change model is being implemented in this framework that captures both macroscale and sub-grid scale heat transfer contributions (available in the experimental DropwiseSGS branch of the gitHub repository). The developers are currently working with other research groups to apply the code to multiscale flow boiling processes, porous media flows, and condensation on structured surfaces. This software is hoped to enable and accelerate the simulation of phase change heat transfer processes, and thus serve as an asset to the thermal sciences and energy engineering communities.

Acknowledgments

The authors wish to acknowledge generous financial support from the U.S. Department of Energy through the Krell Institute (contract DE-FG02-97ER25308), and computing resources from the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under (contract DE-AC02-05CH11231). We also wish to acknowledge Sanjay Adhikari, a PhD student, who helped develop some of the testing scripts for tutorial cases.

This work was financially supported, in part, by the U.S. Department of Energy through the Krell Institute (contract DE-FG02-97ER25308). The funding agency did not actively participate in the research effort or publication process.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at http://dx.doi.org/10.1016/j.softx.2016.10.002.

References

[1] Nusselt W. The surface condensation of steam. Z Ver Dtsch Ing 1916;60:569–75.
[2] Ruckenstein E. On heat transfer between vapour bubbles in motion and the boiling liquid from which they are generated. Chem Eng Sci 1959;10(1–2):22–30.
[3] Lee MS, Aute V, Riaz A, Radermacher R. A review on direct two-phase, phase change flow simulation methods and their applications. In: international refrigeration and air conditioning conference, 2012.
[4] Crowe CT. Review—Numerical models for dilute gas-particle flows. J Fluids Eng 1982;104(3):297.
[5] Mostafa AA, Mongia HC. On the modeling of turbulent evaporating sprays: eulerian versus lagrangian approach. Int J Heat Mass Transfer 1987;30(12):2583–93.
[6] Kolaitis DI, Founti MA. A comparative study of numerical models for Eulerian–Lagrangian simulations of turbulent evaporating sprays. Int J Heat Fluid Flow 2006;27(3):424–35.
[7] Glicksman LR, Hunt AW. Numerical simulation of dropwise condensation. Int J Heat Mass Transfer 1972;15(11):2251–69.
[8] Meakin P. Dropwise condensation: the deposition growth and coalescence of fluid droplets. Phys Scr 1992;44:31–41.
[9] Mei M, Yu B, Zou M, Luo L. A numerical study on growth mechanism of dropwise condensation. Int J Heat Mass Transfer 2011;54(9–10):2004–13.
[10] Krepper E, Koncar B, Egorov Y. CFD modelling of subcooled boiling - concept, validation and application to fuel assembly design. Nucl Eng Des 2007;237:716–31.
[11] Koncar B, Krepper E. CFD simulation of convective flow boiling of refrigerant in a vertical annulus. Nucl Eng Des 2008;238:693–706.
[12] Talebi S, Abbasi F, Davilu H. A 2d numerical simulation of sub-cooled flow boiling at low-pressure and low-flow rates. Nucl Eng Des 2009;239:140–6.
[13] Garma R, Stiriba Y, Bourouis M, Bellagi A. Numerical investigations of the heating distribution effect on the boiling flow in bubble pumps. Int J Hydrogen Energy 2014;39(27):15256–60.
[14] Jo SW, Sherif SA, Lear WE. Numerical simulation of saturated flow boiling heat transfer of ammonia/water mixture in bubble pumps for absorption–diffusion refrigerators. J Therm Sci Eng Appl 2014;6(1).
[15] Yang Z, Peng XF, Ye P. Numerical and experimental investigation of two phase flow during boiling in a coiled tube. Int J Heat Mass Transfer 2008;51(5–6):1003–16.
[16] Kunkelmann C, Stephan P, Kunkelman C, Stephan P, Kunkelmann C, Stephan P, Kunkelman C, Stephan P. CFD simulation of boiling flows using the volume-of-fluid method within OpenFOAM. Numer Heat Transf 2009;56(8):631–46.
[17] Mukherjee A, Kandlikar SG. Numerical simulation of growth of a vapor bubble during flow boiling of water in a microchannel. Microfluid Nanofluid 2005;1:137–45.
[18] Juric D, Tryggvason G. Computations of boiling flows. Int J Multiph Flow 1998;24(3):387–410.
[19] ANSYS FLUENT 17.0. ANSYS, Inc., Canonsburg, PA, 2016.
[20] COMSOL Multiphysics. COMSOL, 2015.
[21] The OpenFOAM Foundation, OpenFOAM 2.4.0, 2015. [Online]. Available: http://openfoam.com/.
[22] Yuan J, Wilhelmsson C, Sunden B. Analysis of water condensation and two-phase flow in a channel relevant for plate heat exchangers. Adv Comput Methods Heat Transf 2006;53:351–60.
[23] De Schepper SCK, Heynderickx GJ, Marin GB. Modeling the evaporation of a hydrocarbon feedstock in the convection section of a steam cracker. Comput Chem Eng 2009;33(1):122–32.
[24] Fang C, David M, Rogacs A, Goodson K. Volume of fluid simulation of boiling two-phase flow in a vapor-venting microchannel. Front Heat Mass Transf 2010;1(1).
[25] Thiele R. Modeling of direct contact condensation with OpenFOAM (M.S. Thesis), Stockholm, Sweden: Royal Institute of Technology; 2010.
[26] Liu Z, Sunden B, Yuan J. VOF modeling and analysis of filmwise condensation between vertical parallel plates. Heat Transf Res 2012;43(1):47–68.
[27] Wilson J, Wehking J, Trautman M, Blue M, Kumar R. Modeling phase change heat transfer of liquid/vapor systems in free and porous media. In: ASME international mechanical engineering congress and exposition, 2015.
[28] Rattner AS, Garimella S. Simple mechanistically consistent formulation for volume-of-fluid based computations of condensing flows. J Heat Transfer 2014;136(7):71501.
[29] Jeon SS, Kim SJ, Park GC. CFD simulation of condensing vapor bubble using vof model. World Acad Sci Eng Technol 2009;60:209–15.
[30] Hardt S, Wondra F. Evaporation model for interfacial flows based on a continuum-field representation of the source terms. J Comput Phys 2008;227:5871–95.
[31] Onishi H, Kawamura M, Tada Y, Takimoto A. Numerical analysis on heat transfer characteristics of looped minichannel using phase-change VOF method. In: ASME international conference on nanochannels, microchannels, and minichannels, 2013.
[32] Esmaeeli A, Tryggvason G. Computations of film boiling. Part I: numerical method. Int J Heat Mass Transfer 2004;47(25):5451–61.

[33] Esmaeeli A, Tryggvason G. Computations of film boiling. Part II: multi-mode film boiling. Int J Heat Mass Transfer 2004;47(25): 5463–5476.
[34] Son G, Dhir VK. Numerical simulation of nucleate boiling on a horizontal surface at high heat fluxes. Int J Heat Mass Transfer 2008;51(9–10): 2566–82.
[35] Subramaniam V, Garimella S. From measurements of hydrodynamics to computation of species transport in falling films. Int J Refrig 2009;32(4): 607–26.
[36] Tao YJ, Huai XL, Li ZG, Yu-Jia T, Xiu-Lan H, Zhi-Gang L. Numerical simulation of vapor bubble growth and heat transfer in a thin liquid film. Chin. Phys Lett 2009;26(7):74701.
[37] Zhang SP, Ni MJ, Ma HY. VOF method for simulation of multiphase incompressible flows with phase change. In: international conference on fluid mechanics, 2011, pp. 579–581.
[38] Rattner AS. Single-pressure absorption refrigeration systems for low-source-temperature applications [Ph.D. thesis]. Atlanta, GA: Georgia Institute of Technology; 2015.
[39] Deshpande SS, Anumolu L, Trujillo MF. Evaluating the performance of the two-phase flow solver interfoam. Comput Sci Discov 2012;5.
[40] Brackbill JU, Kothe DB, Zemach C. A continuum method for modeling surface-tension. J Comput Phys 1992;100(2):335–54.
[41] Renardy Y, Renardy M. PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method. J Comput Phys 2002;183(2): 400–21.
[42] Raeini AQ, Blunt MJ, Bijeljic B. Modelling two-phase flow in porous media at the pore scale using the volume-of-fluid method. J Comput Phys 2012;17(1):5653–68.
[43] Kothe DB, Mjolsness RC, Torrey MD. RIPPLE: a computer program for incompressible flows with free surfaces, Los Alamos National Lab, LA-12007-MS, Los Alamos, New Mexico, 1991.
[44] Hotchkiss RS. Simulation of tank draining phenomena with the NASA SOLA-VOF code, Los Alamos Scientific Lab., NM (USA), 1979.
[45] Özkan F, Wörner M, Wenka A, Soyhan HS. Critical evaluation of CFD codes for interfacial simulation of bubble-train flow in a narrow channel. Int J Numer Methods Fluids 2007;55(6):537–64.
[46] Lafaurie B, Nardone C, Scardovelli R, Zaleski S, Zanetti G. Modelling merging and fragmentation in multiphase flows with SURFER. J Comput Phys 1994;113(1):134–47.
[47] Weller HG, Tabor G, Jasak H, Fureby C. A tensorial approach to computational continuum mechanics using object-oriented techniques. Comput Phys 1998;12(6):620–31.
[48] Patankar SV. Numerical heat transfer and fluid flow. New York, NY: Taylor & Francis Group; 1980.
[49] Marschall H, Hinterberger K, Schuler C, Habla F, Hinrichsen O, Schüler C, Habla F, Hinrichsen O, Schuler C, Habla F, Hinrichsen O. Numerical simulation of species transfer across fluid interfaces in free-surface flows using OpenFOAM. Chem Eng Sci 2012;78:111–27.
[50] Rusche H. Computational fluid dynamics of dispersed two-phase flows at high phase fractions [Ph.D thesis]. London, UK: Imperial College of Science, Technology & Medicine; 2002.
[51] Gaden D. OpenFOAM guide/runTimeSelection mechanism, 2012. [Online]. Available: https://openfoamwiki.net/index.php/OpenFOAM_guide/runTimeSelection_mechanism#declareRunTimeSelectionTable. [Accessed: 11-Aug-2016].
[52] Maric T, Höpken J, Mooney K. The OpenFOAM technology primer. Duisburg, Germany: SourceFlux GmbH; 2014.
[53] Eaton JW, Bateman D, Hauberg S, Wehbring R. GNU Octave Version 4.0.0 Manual: A high-level interactive language for numerical computations. 2015.
[54] Ranz WE, Marshall WR. Evaporation from drops. Chem Eng Prog 1952; 48(3):141–6.
[55] Kim S-J, Park G-C. Interfacial heat transfer of condensing bubble in subcooled boiling flow at low pressure. Int J Heat Mass Transfer 2011; 54(13–14):2962–74.
[56] Raees F, Van Der Heul DR, Vuik C. Evaluation of the interface-capturing algorithm of OpenFoam for the simulation of incompressible immiscible two-phase flow, Report, no. March 2016, p. 43, 2011.
[57] Culpo M. Current bottlenecks in the scalability of OpenFOAM on massively parallel clusters, 2011. [Online]. Available: http://www.prace-ri.eu/IMG/pdf/Current_Bottlenecks_in_the_Scalability_of_OpenFOAM_on_Massively_Parallel_Clusters-2.pdf.