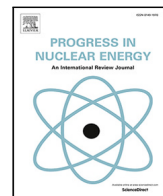




Contents lists available at ScienceDirect

Progress in Nuclear Energy

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

Development and validation of a one-dimensional solver in a CFD platform for boiling flows in bubbly regimes

C. Gómez-Zarzuela^a, C. Peña-Monferrer^b, S. Chiva^c, R. Miró^{a,*}^a Institute for Industrial, Radiophysical and Environmental Safety (ISIRYM) at the Universitat Politècnica de València, Spain^b IBM Research Europe, Hartree Centre, Warrington WA4 4AD, UK^c Department of Mechanical Engineering and Construction at the Universitat Jaume I, Castellón, Spain

ARTICLE INFO

Keywords:

CFD
One-dimensional simulation
Bubbly flow regime
Boiling flows

ABSTRACT

This paper presents a new one-dimensional solver for two-phase flow simulations where boiling is involved. The solver has been implemented within the OpenFOAM® platform. The basic formulation follows the Eulerian description of the Navier–Stokes equations. Different closure equations for one-dimensional simulations are also included, as well as a subcooled boiling model in order to perform accurate computations of the mass and heat transfer between phases. In addition to the fluid, a domain is included in order to represent the solid structure, so the solver is able to solve conjugate heat transfer problems. Two different test cases are presented in this work, first a single-phase test case in order to verify the conjugate heat transfer, and then a case based on the Bartolomej international benchmark, which consists of a vertical pipe where the fluid runs upwards while it is heated. Transient calculations were performed, and the results were compared to the TRACE system code, and to the experimental data in the corresponding case. With this calculations, the capability of this new solver to simulate one-dimensional single-phase and two-phase flows including boiling is demonstrated. This work is a first step of a final objective, which consists in allowing a 1D–3D coupling within the CFD platform, avoiding external links.

1. Introduction

Best-estimate thermal-hydraulic system codes, such as RELAP5 or TRACE, have demonstrated their excellent ability to simulate two-phase flow complex phenomena in nuclear reactors, regardless the operational conditions. This type of codes are predominantly one-dimensional with a wide range of constitutive models (so-called closure relationships), which enhance their capability to represent complex phenomena for multi-phase flow, heat transfer conditions or natural convection, among others. The time consumed during these calculations is reasonably small, and this fact consolidated system codes as a good option, overall.

However, when three-dimensional effects play an important role, such as in pressurized thermal shock transients, boron dilution events or pipe breaks, system codes become inefficient. In order to overcome the one-dimensional limitation and also to simulate complex geometries, computational fluid dynamics (CFD) codes are applied. The use of these codes in nuclear field has been increasing for the last fifteen years, providing significant improvements when dealing with multidimensional flows. The power uprates and the optimization of heat transfer as well as the use of different materials as moderator

in generation IV reactors are some of the causes for the use of CFDs. An example in this context is presented by the authors in Jeong and Han (2008), where the flow distribution in the downcomer and lower plenum of Korean nuclear power plants is assessed using the real geometry in the analysis. A few years later, Lee et al. (2014) modeled the internal structure of a reactor on the accuracy of prediction for the scaled-down APR+ (Advanced Power Reactor Plus) flow distribution, also applying real geometry.

However, until recently, there were still drawbacks that avoided the extended use of these codes in tasks such as design or licensing. Furthermore, the verification and validation of these codes is on going yet. Mahaffy et al. (2007) and Smith et al. (2015) stated the proper methodology to apply CFD in Nuclear Reactor Safety (NRS) as well as different testcases that can be presented in order to validate CFD capabilities, but the lack in experimental data sets to be reproduced delays the verification and validation process. Along with the problems in verification and validation, the other main disadvantage in CFD simulation is the computational time. Even though the computational time has improved with the evolution of the computer technology, it is

* Corresponding author.

E-mail address: rmiro@iqn.upv.es (R. Miró).<https://doi.org/10.1016/j.pnucene.2021.103680>

Received 5 May 2020; Received in revised form 27 January 2021; Accepted 4 February 2021

Available online 16 February 2021

0149-1970/© 2021 The Authors. Published by Elsevier Ltd. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

List of Symbols**Acronyms**

1D	One-dimensional
3D	Three-dimensional
BC	Boundary Condition
CFD	Computational Fluid Dynamics
CV	Control Volume
NRS	Nuclear Reactor Systems
PISO	Pressure-Implicit with Splitting of Operators
RMS	Root Mean Square error
SIMPLE	Semi-Implicit Method for Pressure-Linked Equations
TFM	Two-Fluid Model

Symbols

ρ	Density
U	Velocity
τ	Shear stress term
g	Gravity
F_w	Wall friction term
h	Enthalpy
p	Pressure
q^t	Turbulent heat flux
q	Thermal heat flux
q'''	Volumetric heat source
C_w	Wall drag coefficient
f_w	Friction factor coefficient
D_h	Hydraulic diameter
Re	Reynolds number
μ	Fluid dynamic viscosity
λ	Wall heat transfer coefficient
A_w'''	Wall heat transfer area per unit of volume
Nu	Nusselt number
Pr	Prandtl number
Gr	Grashof number
β	Fluid thermal expansion
f	Filonenko term (friction factor used in wall heat transfer coefficient model)
k	Phase of the fluid (continuous or dispersed)
α_k	Void fraction of phase k
U_k	Phase k velocity
Γ_k	Mass gained by phase k
U_r	Relative velocity
Γ_{ki}	Mass transferred from phase i to phase k
τ_k	Phase k shear stress term
τ_k^t	Phase k turbulent shear stress term
h_k	Phase k enthalpy
$h_{k,sat}$	Phase k enthalpy at saturation conditions
q''_{ik}	Interfacial heat flux in phase k
a_i	Interfacial area
q'''_k	Volumetric heat source in phase k
q''_k	Thermal heat flux in phase k
A_w	Wall surface area
C_p	Heat capacity
γ	Solid conductivity
q_{cond}	Conduction heat flux

M_k	Momentum transfer term of phase k
C_i	Drag momentum term
$v_{g,j}$	Local drift velocity term
$h_{i,k}$	Phase k interfacial heat transfer coefficient
T_{sat}	Saturation temperature
σ	Superficial tension
d_b	Mean bubble diameter
κ_k	Phase k conductivity
A_{ki}	Interfacial area
C_i	Drag coefficient
V	Fluid volume
T_{ONB}	Temperature at the Onset Nucleate Boiling
Γ_{sub}	Mass transfer term due to subcooled boiling
f_{sub}	Fraction of the heat transfer from the wall to the liquid that generates evaporation near the wall when the bulk liquid temperature is subcooled
q''_{SB}	Nucleate boiling heat flux
q_{FC}	Convective heat flux
q''_{PB}	Pool Boiling heat flux
q''_{BI}	Boiling Initiation heat flux
h_{PB}	Pool Boiling heat transfer coefficient
P_r	Reduced pressure
T_{ld}	Temperature at which bubbly detachment occurs
Pe	Peclet number
$\Delta T_{ONB,sat}$	Wall superheated temperature necessary for the onset of nucleate boiling
ΔT_{sub}	Subcooling temperature
q''_{ONB}	Heat flux at the Onset Nucleate Boiling
h_{fg}	Latent heat
ϕ	Correction factor dependent on the material
θ	Contact angle

not competitive enough yet. In order to overcome this problem, the current strategy consist in the coupling of a 1-D system code with a 3-D CFD code. This coupling is characterized by the match of the boundary conditions at the interface of both domains. In a 1D–3D coupling, the general geometry is simulated in the system code, whereas the particular area where accuracy results are needed, is simulated in the CFD. Therefore, the required computational resources are reduced, obtaining accuracy results within a reasonably computational time.

In recent years, the 1D–3D coupling strategy has been applied with different codes and for diverse purposes. [Toti et al. \(2017\)](#) and [Li et al. \(2014\)](#) coupled RELAP5 and Fluent CFD code in order to perform a preliminary validation of their capabilities. The former carried out a study of natural circulation in generation IV reactors, while the latter focused the validation on the ability to simulate a loss of coolant accident. [Bertolotto et al. \(2009\)](#) coupled TRACE with CFX in order to verify the capability of these strategy to represent three dimensional mixing patterns in a double T junction component. More recently, [Godino et al. \(2018\)](#) coupled RELAP5 and OpenFOAM®. They applied this methodology to model a steam generator focusing on the single-phase thermo-hydraulics inside the pre-heater.

So far, it is clear that there are special phenomena in single-phase flows where CFD simulations can bring more detailed information, giving new perspectives and the possibility of define more accurate margins in safety limits. Nonetheless, two-phase flows still requires an extensive study before assessment.

CFD evaluation for two-phase flow is focused in different particular phenomena, like bubble area calculation or subcooled boiling. In [Kurul](#)

(1991), the authors presented a preliminary study for void fraction prediction using the Two-Fluid Model approach (TFM, Ishii and Mishima (1984)) in CFD and including subcooled boiling. Anglart and Nylund (1996) applied similar methodology for rod bundles, getting a good agreement between experimental measurements and averaged axial void at each axial point. Tu and Yeoh (2002) focused their research on low-pressure subcooled boiling flows and pointing out that validated high-pressure models were inefficient in low-pressure cases. They also incorporated the population balance model to their analysis with and without heat transfer (Yeoh and Tu, 2006), finding good agreement in void fraction distribution and liquid velocity, but failing in the vapor velocity prediction, probably due to the assumption that all bubbles have same velocity. Krepper et al. (2007) applied TFM including subcooled boiling to investigate the capability of CFD to contribute to the design of a fuel assembly. They concluded that calculating the temperature of the rod surface one can identify the region where the departure from nucleate boiling could lead to a potential damage in the surface of the pin.

There are also different investigations using TFM approach in CFD: Drzewiecki et al. (2012) carried out a sensitivity study of the parameters of subcooled boiling and two-phase flow models, showing that the bubble diameter is the most influential parameter to the void fraction distribution. Corzo et al. (2012) presented a one-dimensional TFM application implemented in *Octave*, which based the governing equations from a solver taken from OpenFOAM® code. More recently, Colombo and Fairweather (2016) undertook their research trying to avoid tunable models and replacing it by mechanistic models and simulating 20 different experiments in vertical pipes with subcooled boiling. Also, Rollins (2018) developed a TFM CFD solver considering phase change that included a wide range of different closure correlations for the diverse interfacial forces and it was verified and validated for different experiments, showing good agreement.

The work presented in this paper consists in a new one-dimensional solver developed in the CFD code OpenFOAM®, used for both single-phase and two-phase flow simulations. The final purpose of this solver is getting a one-dimensional solver within a CFD framework, substituting the system code in 1D–3D couplings, and simplifying the coupling by being both models within the same platform. This paper is focused on the one-dimensional solver development. The built-in solvers used as a basis in this work only calculate the fluid parameters, without studying the wall temperature or solid heat conduction implications. Furthermore, the standard thermophysical models that calculates the fluid parameters are also limited to be constant or dependent of one variable. These disadvantages along with the computational time and resources required to solve the 3D mesh, where the time step is several times smaller than the order of magnitude of a one-dimensional simulation, brings the necessity of applying modifications and simplifications to this solver. Once the new solver is implemented, the verification and validation for single-phase flows simulation is presented. Then, same procedure is undertaken for two-phase flows simulations. This paper is organized as follows: Section 2 presents the mathematical formulation used in the development. Then, Section 3 describes the basis of the solver, since the new code departs from a pre-defined solver included in the selected CFD code, that is, within OpenFOAM®. In this section it is also described the methodology followed in this work and the final loop defined in the new solver. Section 4 presents the different testcases simulated in this work. All simulations are compared with the results obtained by the system code TRACE for the same conditions. TRACE has been chosen for this verification due to its extended validation in the scientific community. The results are also compared to the Bartolomej benchmark experimental results (Bartolomej and Chanturiya, 1967) and to the results obtained with a new three-dimensional solver developed in OpenFOAM® (Rollins, 2018). This new solver also considers boiling. Last, the discussion along with the conclusions and some guidelines for future steps are presented in Section 5.

2. Mathematical formulation

The solver developed in this work was implemented within the OpenFOAM® framework (Weller et al., 1998). OpenFOAM® was meant as a set of libraries that allows to solve the fluid-dynamic equations. It is open-sourced, which is one of its main advantages, due to the total control provided to the user over the code. It is composed of a variety of solvers, where each one is meant to solve particular conditions. In this case, the two-phase flow analysis is based on the solver called *twoPhaseEulerFoam* (OpenFOAM, 2017a), which is a widely used solver for Eulerian simulations based on the TFM approach.

However, in order to start from a simpler case, such a single-phase simulations, a different solver was used. This initial solver was *buoyantPimpleFoam* (OpenFOAM, 2017b), which is a *solver for turbulent flows of compressible fluids in transient simulations* (OpenFOAM, 2017c). Once the single-phase simulations were verified, the implementation was moved to the two-phase solver. Then, the latter was adapted so that it was able to work with single-phase and two-phase flows.

The single-phase version of the new solver presented in this work must be able to work with both liquid and gas states. Although starting with an incompressible solver would have been the natural option, this was discarded. In OpenFOAM® incompressible solvers, pressure is normalized by a constant fluid density. Then, the density is no longer involved in the calculation. One needs a thermophysical model, which are generally a constant value of density or the ideal gas law. Therefore, we decided to start with a compressible solver, where density is involved in the calculation since it is not assumed constant.

In order to undertake this work, the general structure of *buoyantPimpleFoam* remained unchanged, but the equations were replaced by those used in the one-dimensional reference code, which in this case, the system code TRACE was selected. Therefore, the final solver keeps the structure of the original CFD solver, but the equation corresponds to a one-dimensional system, which is the main purpose of this work.

The built-in solver *buoyantPimpleFoam* is characterized by its ability to simulate in 3D flows that are

- Compressible
- Turbulent
- Able to consider convective heat
- In transient conditions

The substitution of the three dimensional Navier–Stokes equations by the TRACE one-dimensional system of equations implies the adaptation of the features listed above. Regarding to the compressibility, the new system of equations is used also for compressible fluids, so this feature is conserved. In turn, 3D turbulence is not considered in the new 1D solver implemented in this work. Instead, a wall friction model is considered in order to take into account the axial contribution of the diffusion term. The wall friction model included is also taken from TRACE system code, in order to ensure the correct performance of the new code when inter-comparison of results with the former is applied. The ability of heat transfer is also considered in the new solver by including TRACE energy equation. Finally, the transient condition is remained, since the set of equations used as reference is time-dependent.

Up here, the new solver is summarized as a solver based on the built-in *buoyantPimpleFoam* but with one-dimensional conservation equations. Once this structure was assessed, new features were also implemented in the new 1D solver. In particular, the addition of a new domain to consider the solid wall and the integration of the corresponding wall heat transfer model were also carried out. Again, both solid conduction equation and wall heat transfer correlation were based on TRACE equations in order to avoid discrepancies in the simulation comparisons performed afterwards. Section 2.1 presents the new equations implemented and the differences with the original, along with the new closure models added to the code.

Similar methodology was followed to implement the 1D two-phase solver. In this case, the OpenFOAM® solver used as a basis was twoPhaseEulerFoam as mentioned before. Again, the set of original equations of this solver was replaced by the 1D system of conservation equations found in TRACE. In this case, twoPhaseEulerFoam in fluids with the following features:

- With 2 compressible phases, where one is dispersed in the other
- Turbulent flows
- Able to transfer heat between phases
- In transient conditions

Again, in this case, when the substitution of the equations is applied, these properties are adapted. Regarding to the Eulerian model, this is conserved since the 1D system of equations is also based on the Eulerian approach. On the other hand, and similarly to the previous case, the 3D turbulence is replaced by a 1D wall friction model. And finally, both heat transfer between phases and transient conditions are remained since the 1D energy equation considers the interfacial heat transfer and all the implemented equations are time-dependent. In this case, the different closure correlations integrated in the new solver were selected due to its contribution in the reference system code used in this work. Following same methodology as in the single-phase stage, once the fluid equations were assessed, the solid domain was applied. Section 2.2 presents the new implemented Eulerian system of equations, as well as the different closure models included in the new 1D solver.

Once the modified 1D two-phase flow solver was evaluated and its capability for two-phase flow simulations tested, in order to merge both single and two-phase solvers, this last solver was adapted to work with single-phase simulations, assuming that a fluid is single-phase when the void fraction, which is the ratio of presence of dispersed phase respect to the continuous one, is below $1e-7$. The final solver is called my1DTPFoam.

2.1. Single-phase flow governing equations

This section presents the arrangements applied to the original Navier–Stokes equations found in the solver buoyantPimpleFoam to obtain the final set of one-dimensional conservation equations.

The general Navier–Stokes equation for continuity in single-phase flows reads as follows

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho U) = 0. \quad (1)$$

where ρ represents the density of the fluid. Since this work aims at one-dimensional simulations, it is necessary to choose a preferential direction and neglect the rest. Selecting x axis as the axial direction, the one-dimensional mass conservation equation is given as

$$\frac{\partial \rho}{\partial t} + \frac{\partial (\rho U)}{\partial x} = 0. \quad (2)$$

The terms in Eq. (2) represents the net change of density in time and the net mass flow out of the boundaries which are normal to the x direction.

The momentum conservation equation found in the original solver is given by the following

$$\frac{\partial (\rho U)}{\partial t} + \nabla \cdot (\rho U U) = -\nabla p + \rho g + \nabla \cdot (2\mu_{eff} D(U)) - \nabla \cdot \left(\frac{2}{3} \mu_{eff} (\nabla \cdot U) \right). \quad (3)$$

In Eq. (3), g is the gravity and the last two terms in the right-hand side represent the momentum diffusion term or stress tensor, also known as turbulence. The effective viscosity μ_{eff} accounts for the sum of molecular and diffusive viscosity.

In general, in one-dimensional calculations the turbulence or diffusion term is simplified and only the axial contribution of this term needs to be considered. In order to take into account this term, a wall

friction model is added to the system to replace the turbulence model. Thus, considering wall friction model and one preferential direction, the one-dimensional momentum equation is

$$\frac{\partial (\rho U)}{\partial t} + \frac{\partial (\rho U U)}{\partial x} = -\frac{\partial p}{\partial x} + \rho g + F_w. \quad (4)$$

The different terms in Eq. (4) represent, from left-hand side, the rate of increase of momentum per unit time and unit volume, respectively, and in the right-hand side, the contribution of the pressure, the force due to gravity and the viscous stresses in form of wall friction force. Typically, in 1D formulation the gravity term considers the angle of the X axis to the horizon. However, this is not the case, since in this CFD platform the angle is considered during the geometry building, and the distance between cells in each direction (in this case, the x direction) is already calculated. Therefore, the influence of term ρg is calculated in each direction. In particular, for X axis the impact of this term is calculated as $g d_x \frac{\partial}{\partial x} \rho$, where d_x is the distance between two cell centers in x direction.

The term F_w represents the wall friction model, which is calculated as a value proportional to the velocity.

Lastly, the initial energy equation can be written, in terms of sensible enthalpy, as

$$\frac{\partial (\rho h)}{\partial t} + \nabla \cdot (\rho U h) + \frac{\partial (\rho K)}{\partial t} + \nabla \cdot (\rho U K) = \frac{\partial p}{\partial t} + \nabla \cdot (\alpha_{eff} \nabla h) + \rho U g. \quad (5)$$

Eq. (5) presents on the left-hand side of the equation the rate of change in the fluid. In this side, K represents the kinetic energy. The right-hand side represents the net flux and the rate of work applied to the fluid due to the surface and body forces. The term α_{eff} is the effective thermal diffusivity and it accounts for both laminar and turbulent diffusivities.

In the one-dimensional system, the kinetic energy is given per unit of volume. Therefore $K = \frac{1}{2} \rho U^2$. Besides, the new one-dimensional equation can be obtained assuming again the preferential direction and neglecting turbulence tangential term because only axial divergence is considered. The influence of the axial viscous term is included in a convective heat transfer model. Rearranging, Eq. (5) can be simplified as

$$\frac{\partial \left[\rho \left(h + \frac{U^2}{2} \right) \right]}{\partial t} + \frac{\partial \left[\rho U \left(h + \frac{U^2}{2} \right) \right]}{\partial x} = \frac{\partial p}{\partial t} + \rho U g - q'' \frac{a_w}{V} + q'''. \quad (6)$$

The final terms in Eq. (6) represent, starting from the left-hand side, the increase of energy per unit of time and volume. On the right-hand side, $\rho U g$ states for the work required to overcome gravity, q'' represents the thermal convective heat flux and q''' is the volumetric source, in case there is any. The term a_w symbolizes the heated wall surface, whereas the term V represents the volume of each cell. The thermal convective heat flux accounts for the axial term of the turbulent diffusivity. The implemented equation does not consider radiation heat within the fluid, it is necessary to define it as external sources to take them into account.

The new formulation requires two closure models in order to solve all unknowns. In this case, since one-dimensional simulations are performed, a wall friction model to solve momentum equation and a correlation for heat transfer coefficient were implemented so that energy equation can be solved. These correlations do not replace any old model, but they are simply added as new models. As mentioned before, these both correlations are also used in TRACE, the reference code, so they are used here to make a rigorous comparison.

Regarding to the wall friction model (F_w in Eq. (4)), it is modeled as a term proportional to the velocity squared. In TRACE system code, this is represented as $F_w = C_w \cdot U^2$. The proportional term, C_w , also called wall drag coefficient, is defined as the following

$$C_w = f_w \frac{2\rho}{D_h}, \quad (7)$$

The term f_w is the Fanning friction factor, and it is calculated using the Churchill correlation. This formula has been chosen due to its suitability with any kind of regime (United State Nuclear Regulatory Commission, 2013; Churchill, 1977). The correlation proposed by Churchill is given by

$$f_w = 2 \left[\left(\frac{8}{Re} \right)^{12} + \frac{1}{(a+b)^{3/2}} \right]^{1/12} \quad (8)$$

In Eq. (8), Re is the Reynolds number a and b are the following expressions,

$$a = \left\{ 2.457 \cdot \ln \left[\frac{1}{\left(\frac{7}{Re} \right)^{0.9} + 0.27 \cdot \left(\frac{\epsilon}{D_h} \right)} \right] \right\}^{16} \quad (9)$$

and

$$b = \left(\frac{3.753e^4}{Re} \right)^{16} \quad (10)$$

In Eq. (9), ϵ represents the roughness of the solid structure and D_h the hydraulic diameter.

On the other hand, the heat transfer rate from the wall to the liquid is function of a wall heat transfer coefficient (λ), which considers the amount of heat that the fluid is able to dissipate. This term is necessary to calculate the convective heat flux, $q'' = \lambda(T_w - T_{fluid})$. From now on, the convective heat flux term q'' is called q''_{conv} for the sake of clarity. For convective single-phase fluids, the wall heat transfer is taken as the maximum value of the heat transfer coefficients for the three possible different regimes, which are $h = \max(h_{lam}, h_{turb}, h_{NC})$, where the subscripts *lam*, *turb* and *NC* accounts for the laminar, turbulent and natural convection regime, respectively. The general equation of the wall heat transfer coefficient is

$$\lambda = Nu \frac{\kappa_l}{D_h} \quad (11)$$

Therefore, since the hydraulic diameter D_h is constant within a particular region and the fluid conductivity κ_l is also constant for a particular pressure and temperature, the calculation is reduced to the Nusselt number for each regime. For laminar flows, the Nusselt has been fixed to $Nu_{lam} = 4.36$ (Holman, 1997), which is the analytical value for a fully developed flow in this regime. In natural convection flows, the solver selects the maximum value between laminar and turbulent natural convection, $Nu_{NC} = \max(Nu_{NC,lam}, Nu_{NC,turb})$. For laminar natural convection flows, the value of the Nusselt number depends on the Grashof and Prandtl numbers, $Nu_{NC,lam} = c(GrPr)^{1/4}$ (Holman, 1997) and (Rohsenow et al., 1998), where c is a constant dependent on the geometry. The Grashof number represents the influence of the buoyancy with respect to the viscous force, and it is given by

$$Gr_l = \frac{g\beta_l \Delta T L^3}{(\mu_l/\rho_l)^2} \quad (12)$$

In Eq. (12), the numerator shows buoyancy terms, which are gravity (g), liquid thermal expansion (β_l), difference of temperatures (ΔT) and characteristic length (L). The latter is replaced by the hydraulic diameter, so $L = D_h$.

The turbulent natural convection Nusselt number is stated as $Nu_{NC,turb} = C_t(GrPr)^{1/3}$ (Holman, 1997) and (Rohsenow et al., 1998). It is worthy to note that both laminar and turbulent formulas were developed for vertical plates (Holman, 1997), and in this case, the constants c and C_t are averaged. The former value is 0.59 and the latter corresponds to 0.1 (Rohsenow et al., 1998).

Lastly, in order to avoid disagreement due to the use of hydraulic diameter instead of characteristic length, Eq. (11) for turbulent natural convection is readjusted in the following way

$$\lambda_{NC,turb} = 0.1 \frac{\kappa_l}{D_h} \left[\frac{g\beta_l \Delta T D_h^3}{(\mu_l/\rho_l)^2} \right]^{1/3} Pr^{1/3}, \quad (13)$$

which leads to

$$\lambda_{NC,turb} = 0.1 \kappa_l \left[\frac{g\beta_l \Delta T}{(\mu_l/\rho_l)^2} \right]^{1/3} Pr^{1/3}, \quad (14)$$

so that the characteristic length is avoided.

For NC laminar flows same arrangement is applied, but in this case, the characteristic length cannot be removed from the equation because their power index do not match, but since the influence of this term is one fourth ($Nu \approx Dh^{-1/4}$), the deviations can be accepted.

Regarding turbulent flows, the Nusselt number is calculated using Gnielinski correlation (Gnielinski, 1976). This model is given by the following

$$Nu_{turb} = \frac{(f/2)(Re - 1000)Pr}{1 + 12.7(f/2)^{1/2}(Pr^{2/3} - 1)}. \quad (15)$$

In this equation, the parameter f is a friction factor calculated by Filonenko formula (Filonenko, 1954)

$$f = [1.58 \ln(Re) - 3.28]^{-2}. \quad (16)$$

2.2. Two-phase flow governing equations

Once the single-phase solver was verified, the work was moved forward to the final baseline solver, which was twoPhaseEulerFoam. This solver is defined as a solver for a system of two compressible fluid phases with one dispersed phase (OpenFOAM, 2017c). The standard twoPhaseEulerFoam solves a two-phase fluid system using the TFM model. However, it does not consider boiling or condensation and therefore, it is either not able to take subcooled boiling phenomena into account.

First step in this second stage of the work consisted in replacing the 3D equations by the 1D system of equations considered for this purpose. Furthermore, the built-in solver was meant to simulate two-phase flows without mass exchange, but the final solver must be able to consider mass transfer between phases and also events where subcooled boiling happens. Therefore, this section starts introducing the differences between the original and the final set of equations and the necessary arrangements to obtain the latter and then, it presents the different models that were necessary to implement in the solver to give it these capabilities.

The mass conservation equation for a phase in a two-phase flow system in twoPhaseEulerFoam is given as follows:

$$\frac{\partial(\alpha_k \rho_k)}{\partial t} + \nabla \cdot (\alpha_k \rho_k U_k) = 0. \quad (17)$$

where k represents the phase, which can be dispersed ($k = v$) or continuous ($k = l$). Term α_k represents the void fraction, U_k the velocity of the phase k and ρ is the density of the phase. In this case, it is necessary to consider the contribution of the mass transfer between phases, in case it exists.

Let be Γ_k the mass gained by phase k . It is important to note that the increase of mass in one phase is equivalent to the loss of mass in the other phase ($\Gamma_v = -\Gamma_l$), so that the conservation is satisfied in the system.

As in Section 2.1, assuming X axis as the preferential direction, the final one-dimensional two-phase equation implemented for a general phase k stays as follows

$$\frac{\partial(\alpha_k \rho_k)}{\partial t} + \frac{\partial(\alpha_k \rho_k U_k)}{\partial x} = \Gamma_k. \quad (18)$$

Regarding to momentum equation, the formulation used in twoPhaseEulerFoam is

$$\frac{\partial(\alpha_k \rho_k U_k)}{\partial t} + \nabla(\alpha_k \rho_k U_k U_k) = -\alpha_k \nabla p + \nabla[\alpha_k(\tau_k + \tau'_k)] + \alpha_k \rho_k g + M_k, \quad (19)$$

where the subscript k represents the phase (e.g. continuous or dispersed phase). Departing from the beginning of the left-hand side of the

Eq. (19), the different terms represent the change of momentum in the fluid with time and the rate of change of momentum through the faces of the volume. On the right-hand side, following the same order, one can find the change in pressure due to the different forces, terms $\tau_k + \tau'_k$ are the combined Reynolds viscous and turbulent stress, the gravity term and the averaged interfacial momentum transfer term.

In order to implement the momentum equation considering the effects due to mass transfer between phases, the momentum transfer during the phase change, $\Gamma_k U_k$, is added to the equation. Similar to Section 2.1, turbulence (Reynolds stresses) in the new system is simplified and replaced by the axial viscous term, which is considered in the wall friction model. Additionally, taking x axis as the axial direction, the equation can be reduced in the following way

$$\frac{\partial(\alpha_k \rho_k U_k)}{\partial t} + \frac{\partial(\alpha_k \rho_k U_k U_k)}{\partial x} = -\alpha_k \frac{\partial p}{\partial x} + \alpha_k \rho_k g + F_{wk} + \Gamma_k U_k + M_k. \quad (20)$$

Eq. (20) is the final momentum equation for a general phase k implemented in the new one-dimensional solver presented in this work. There are two closure models involved in this equation. On the one hand, the wall friction F_{wk} , which has a different influence on each phase. This term is regime dependent, and for this work, focused on bubbly/slug regime. M_k is the generalized drag force (Ishii and Hibiki, 2011), which involves the momentum applied to the phase due to the different force interactions.

Lastly, the original two-phase energy equation is given by

$$\begin{aligned} \frac{\partial(\alpha_k \rho_k h_k)}{\partial t} + \nabla(\alpha_k \rho_k h_k U_k) + \frac{\partial(\alpha_k \rho_k K)}{\partial t} + \nabla \cdot (\alpha_k \rho_k U_k K) = \\ -\nabla[\alpha_k(q_k + q'_k)] + \frac{\partial(\alpha_k p)}{\partial t} + \nabla \cdot (\alpha_k T_k U_k) + \alpha_k \rho_k U_k g + S_k. \end{aligned} \quad (21)$$

where K states for the kinetic energy, $q_k + q'_k$ account for both the thermal and the turbulent convection heat fluxes, T_k represents the work due to shear forces and S_k is the interfacial supply energy.

In solving problems, it is often useful to separate the mechanical and thermal effects in the total energy equation. In these cases, the mechanical equation is subtracted from the thermal energy equation. This method is applied here, focusing on enthalpy equation (Todreas and Kazimi, 2011). The final system proposed in this work must also consider the possible interfacial energy due to the phase change, and this is done by adding to the equation the term $\Gamma_k h_{k,sat}$, which accounts for the heat transferred from phase k to the interface. Assuming x axis as the preferential direction, replacing turbulence work by simplified models that consider the axial viscosity in terms of wall friction and heat transfer and rearranging, the final one-dimensional energy equation for two-phase flow for a general phase k used in this work is given as

$$\begin{aligned} \frac{\partial(\alpha_k \rho_k h_k)}{\partial t} + \frac{\partial(\alpha_k \rho_k h_k U_k)}{\partial x} = \frac{\partial(\alpha_k p)}{\partial t} + \alpha_k \rho_k U_k g + \\ \Gamma_k h_{k,sat} + q''_{ik} a_i - \alpha_k q''_{ik} a_w + q''_k. \end{aligned} \quad (22)$$

Eq. (22), includes new terms that are worth pointing out. The term $\alpha_k q''_k$ states for the thermal heat flux, which encompasses the wall heat flux and the heat flux due to turbulence in those cases when it is considered. Moreover, the term Γ_k represents the mass transfer between phases, and $q''_{ik} a_i$ accounts for the interfacial heat, which is the heat exchanged from each phase to the interface. Term $\alpha_k q''_{ik} a_w$ is the thermal heat flux of phase k transferred through the wall and q''_k is the interfacial supply energy.

Regarding to the closure equations for the one-dimensional system, as already seen in Section 2.1, were implemented in the code as new models, apart from the already built-in models, so they did not replace any model. In this case, in addition to the wall friction and heat transfer model, it was necessary to implement different correlations that worked in one-dimensional calculations for the interfacial forces that influence the total phase momentum, such as the interfacial drag

and lift forces. On the other hand, in one-dimensional simulation, the influence of some interfacial forces can be neglected (Brooks et al., 2012). Only those related with the preferential direction or the heat transfer are included, so in this work, only drag force and interfacial heat transfer are considered. Furthermore, it is worthwhile to remark that the closure equations are regime dependent. In this work, only bubbly/slug regime was considered, so the selected models are suitable only for this two-phase flow regime.

The interfacial drag correlation is based on the drift flux approach (Zuber and Findlay, 1965; Ishii and Mishima, 1984). In a drift flux model the mixture of the phases is solved as a whole.

The general area averaged interphase drag term is given as

$$M_d = C_i |U_r| U_r. \quad (23)$$

In Eq. (23), the drag momentum is a function of the drag coefficient C_i and the relative velocity U_r . However, when the drift velocity approach is applied, the term M_d is given as a balance of the forces in the direction of the flow. It considers the interfacial drag, buoyancy and pressure drop and applies the assumptions that both phases have equivalent pressure and the action–reaction principle for the interfacial momentum terms. As a result, the interfacial drag is

$$M_d = \alpha_k (1 - \alpha_k) \Delta \rho g. \quad (24)$$

The term α_k stands for the dispersed phase void fraction. The drag coefficient can be obtained from the definitions of Eqs. (23) and (24), obtaining the following interfacial drag coefficient

$$C_i = \frac{\alpha_k (1 - \alpha_k) g \Delta \rho}{U_r^2}. \quad (25)$$

Finally, in order to solve Eq. (25), the relative velocity (U_r) will be replaced by the drift flux correlation, rewriting the equation in terms of the weighted drift flux velocity, v_{gj} , and the distribution parameter $C0$. The complete development can be found in Zuber and Findlay (1965) and in United State Nuclear Regulatory Commission (2013). The final implemented equation is the following

$$C_i = \frac{\alpha_k (1 - \alpha_k) g \Delta \rho}{U_r^2} = \frac{\alpha_k (1 - \alpha_k)^3 g \Delta \rho}{v_{gj}^2} \frac{(\frac{1-C0(\alpha_k)}{1-\alpha_k} \bar{U}_g - C0 \bar{U}_l)^2}{U_r^2}. \quad (26)$$

Once the drag coefficient model is presented, it remains to define a correlation to calculate the drift flux velocity and the distribution parameter.

In order to calculate the drift flux velocity, a model was chosen that covers from a dispersed regime with small uniform bubbles to a slug flow, where larger bubbles flow through the pipe.

The following formula for this regime was introduced by Ishii and Hibiki (2011)

$$v_{gj}^- = \sqrt{2} \left(\frac{\sigma g \Delta \rho}{\rho_l^2} \right)^{1/4}, \quad (27)$$

where σ represents the superficial tension. On the other hand, the distribution parameter, which is used for all regimes, is given by

$$C0 = 1.2 - 0.2 \sqrt{\frac{\rho_g}{\rho_l}}. \quad (28)$$

However, as United State Nuclear Regulatory Commission (2013) mentioned, as the void fraction increases, the bubbles grow up towards a Taylor cap regime and the flow might progressively reach slug situation. In these cases, where larger bubbles are presented and they might be agglomerated in some zones of the pipe, the previous model, which is devoted to uniform dispersed flows, is no longer accurate.

A different model was also implemented for slug regimes, based on Kataoka and Ishii (1987). Now the drift flux velocity depends on a non-constant term

$$v_{gj}^- = v_{gj}^- + \left(\frac{\sigma g \Delta \rho}{\rho_l^2} \right)^{1/4}, \quad (29)$$

where v_{gj}^{-+} is known as the weighted drift velocity. This term is function of the pipe diameter, which is written in non-dimensional form, and is given by the following

$$v_{gj}^{-+} = 0.0019 \cdot \text{Min}[30, D_h^*]^{0.809} \left(\frac{\rho_g}{\rho_l}\right)^{-0.157} (N_{\mu l})^{-0.562} \quad (30)$$

where dimensionless diameter is defined as

$$D_h^* = \frac{D_h}{\sqrt{\sigma/g\Delta\rho}} \quad (31)$$

and the continuous phase viscosity number is given as

$$N_{\mu l} = \frac{\mu_l}{(\rho_l \sigma \sqrt{\sigma/g\Delta\rho})^{1/2}} \quad (32)$$

The value of 30 in Eq. (30) for the dimensionless diameter establishes a maximum limit between what is considered small pipes and large pipe diameters. As the pipe diameter increases, the bubbles could lead to cap bubble regime due to potential surface imbalances that might disintegrate them.

The final drift flux velocity is the maximum of both Eqs. (27) and (29).

In order to take into account the heat transfer between the liquid and the bubble interface ($q''_{i,l}$), it is necessary to determine the heat transfer coefficient ($h_{i,l}$), which is calculated by means of Ranz–Marshall correlation (Ranz et al., 1952). The interfacial heat transfer between the vapor and the interface ($q''_{i,v}$) works similar and its heat transfer coefficient ($h_{i,v}$) is considered constant and equal to 1000 W/m² K for bubbly regime. This value was chosen due to its implication also in the reference system code model. Once the verification is assessed, a more accurate model will be searched. Thus, the interfacial heat transfer term for liquid phase (heat transferred from the liquid to the interface) in Eq. (22) is calculated by the following

$$q''_{i,l} = h_{i,l}(T_l - T_{sat}). \quad (33)$$

Therefore, term $q''_{i,l}$ will become positive when the liquid is superheated. On the other hand, the vapor interfacial heat transfer term (from vapor phase to the interface) is equivalent

$$q''_{i,v} = h_{i,v}(T_v - T_{sat}). \quad (34)$$

According to Eq. (22), this term completed by considering the interfacial area. The surface of the dispersed bubbles is calculated as

$$a_i = \frac{6\alpha_k}{d_b}. \quad (35)$$

The term d_b represents the bubble diameter. This value is calculated considering the bubbles in this regime as an approximation to distorted bubbles, since it is assumed that the spherical ones rarely appear. Taking into account this approximation, the bubble diameter is given by

$$d_b = 2\sqrt{\sigma/g\Delta\rho}, \quad (36)$$

where σ is the superficial tension, g represents the gravity and $\Delta\rho$ is the density difference between phases.

2.3. Subcooled boiling model

In systems where the wall temperature overcomes the fluid saturation temperature, small bubbles may be generated close the wall due to an increase of the liquid temperature in this region with respect to the bulk temperature. Therefore, even though the fluid is not in saturation conditions, there are local areas where vapor begins to occur. This phenomena is included in the mass transfer between phases term

$$\Gamma_k = \Gamma_i + \Gamma_{sub}. \quad (37)$$

In Eq. (37), Γ_i represents the mass transferred between phases due to the interfacial heat

$$\Gamma_i = \frac{q''_{i,l}a_i + q''_{i,v}a_i}{h_{v,sat} - h_l}. \quad (38)$$

In Eq. (38), the terms $h_{v,sat}$ and h_l are the enthalpy of the vapor at saturation conditions and the liquid enthalpy. Conversely, Γ_{sub} represents the mass transferred to vapor phase due to the local boiling near the wall occasioned by the high temperature of this. However, not all the generated vapor finally account as vapor phase, but only a fraction continues as vapor. The rest of the generated bubbles may collapse due to condensation or may not detach from wall because they do not evolve sufficiently. Thus, only a fraction of this boiling nucleation should be considered in the mass transfer term:

$$\Gamma_{sub} = \frac{f_{sub}(q''_{SB} - q''_{l,conv})a_w}{h_{v,sat} - h_l}. \quad (39)$$

In Eq. (39), f_{sub} is the fraction of vapor that is considered to remain as part of the vapor bulk. The terms q''_{SB} and $q''_{l,conv}$ are the subcooled nucleate boiling and the liquid forced convection heat fluxes, and a_w is the surface area of the heat transfer surface per unit volume. The fraction of vapor f_{sub} is taken from the Lahey mechanistic subcooled boiling model (Lahey Jr., 1978) but the formula is readjusted to work with temperatures rather than enthalpies,

$$f_{sub} = \max\left(0, \frac{T_l - T_{ld}}{T_{sat} - T_{ld}}\right), \quad (40)$$

where T_{ld} is the temperature of detachment of the bubbles from the wall. The terms T_l and T_{sat} are the temperature of the liquid phase and the liquid temperature at saturation conditions.

In order to take into account the subcooled nucleate boiling heat flux, a pool boiling model was implemented in the solver since the former is considered as a linear superposition of the forced convection and pool boiling heat fluxes (Chen, 1966), this is

$$q''_{SB} = q''_{l,conv} + (q''_{PB} - q''_{BI}). \quad (41)$$

The term q''_{PB} in Eq. (41) is the pool boiling heat flux at wall temperature. This term is calculated using Gorenflo model (Gorenflo et al., 2014), which its explicit formulation is given by the following

$$q''_{PB} = \left(h_0 \frac{F_p}{q_0^n}\right)^{\left(\frac{1}{1-n}\right)} (T_w - T_{sat})^{\left(\frac{1}{1-n}\right)}. \quad (42)$$

The terms h_0 and q_0^n in Eq. (42) are constant empirical values, while the terms F_p and n are dependent on the critical pressure. Apart from the empirical terms of the formula, the pool boiling is a function of the wall temperature and the saturation temperature of the continuous phase.

Back to Eq. (41), the term q''_{BI} is the heat flux at the point of boiling initiation, this is $q''_{BI} = q''_{PB}(T_{ONB})$. It is assumed that the onset nucleate boiling temperature, T_{ONB} , is the wall temperature at which bubble generation will start, since this effect does not necessarily start just after reaching saturation temperature, but a little higher, so that saturation temperature is reached locally by the fluid. Considering the definition of T_{ONB} , the term q''_{BI} helps to smooth the increasing curve of vapor represented by q''_{SB} , so it avoids large discontinuities at the point where the subcooled boiling starts.

The onset nucleate boiling temperature, T_{ONB} , is calculated by the following (Basu et al., 2002):

$$T_{ONB} = T_L + \frac{1}{4} \left(\sqrt{\Delta T_{ONB,sat}} + \sqrt{\Delta T_{ONB,sat} + 4\Delta T_{sub}} \right)^2 \quad (43)$$

The Eq. (43) is a rearrangement that can be found in United State Nuclear Regulatory Commission (2013) based on the model presented in Basu et al. (2002). ΔT_{sub} is the difference between liquid and saturation temperatures (subcooling temperature), $\Delta T_{sub} = T_{sat} - T_{liq}$.

$\Delta T_{ONB,sat}$ is the wall superheated temperature necessary for the onset of nucleate boiling when the liquid is at saturation and it is stated as the following

$$\Delta T_{ONB,sat} = \frac{2h_l \sigma T_{sat}}{\phi^2 \rho_v h_{fg} \kappa_l} \quad (44)$$

In Eq. (44), κ_l is the liquid conductivity, while the term h_{fg} represents the latent heat of the fluid, which is the energy transferred between phases in conditions of evaporation or condensation. On the other hand, ϕ accounts for a correction factor that depends on contact angle. This factor is determined to be

$$\phi = 1 - \exp(-\theta^3 - 0.5\theta), \quad (45)$$

where θ is the contact angle and it is measured in radians. It depends on the solid material, and for stainless steel, which is the solid selected in this work, it is constant and equal to 0.663225 rad (Basu et al., 2002).

2.4. Wall conduction equation

Up here, the system of one-dimensional conservation equations implemented in the new solver is presented. Therefore, the solver must be able to simulate the fluid behavior of single- and two-phase flows where heat transfer may happen and also boiling phenomena. There are fields where the heat transfer among fluid and solid is specially important because the dissipation of heat must be maximum in order to avoid structural problems. In this solver a solid domain was included in order to consider the heat transfer between fluid and solid. In order to consider the conduction heat through the solid and how it influences the local fluid temperature near the wall, the following conduction equation was implemented

$$\rho C_p \frac{\partial T}{\partial t} = \frac{\partial}{\partial y} \left(\kappa \frac{\partial T}{\partial y} \right) + q''' \quad (46)$$

Eq. (46) is calculated in only one axis. The preferential direction taken is that in which the heat flows through the wall from the external side to the fluid. This direction is perpendicular to the direction of the fluid, so Y axis has been taken as reference.

In order to find the proper surface temperature at the interface liquid–solid, a heat balance is imposed. Therefore, the convective heat flux in the fluid side (introduced as $q_{l,conv}$ in Section 2.3) must be equal to the conduction heat flux from the solid side, this means $q_{SB} = q_{cond}$ at the interface. The term q_{cond} states for the conductive heat flux transferred from the solid to the fluid. This term is calculated as

$$q_{cond} = \kappa \frac{T_w - T_{cell}}{\delta y} \quad (47)$$

Term κ is the solid conductivity, T_w is the wall temperature at the solid–fluid interface, while T_{cell} is the temperature at the adjacent solid cell. The denominator considers the distance from the center of the solid to the interface. In Section 3, this will be shown in detail.

3. Numerical implementation

The mathematical formulation exposed in Section 2 was implemented within the OpenFOAM® framework (OpenFOAM, 2017c). OpenFOAM-v1712 version was used in this work. The methodology departed from a standard OpenFOAM® solver which then was modified to adapt it for the conditions of this work.

3.1. OpenFOAM framework

OpenFOAM® is a collection of C++ libraries designed to solve complex problems in fluid mechanics. The toolkit implements operator-based implicit and explicit Finite Volume (FV) discretization in the three dimensional space. The structure of OpenFOAM® allows users to adapt the standard solvers and libraries with moderately effort for different problems in several fields. Consequently, a wide variety

of solvers can be found in the scientific community that provides flexibility to the software and capability to perform calculations at different scales and among diverse domains. Therefore, this program can be considered as a multi-physics multi-scale tool.

Another of its main strengths is its open-sourced development, allowing the user to modify the code at any level, not only the standard solvers, but also the hierarchical design. Therefore, the user can extend or modify any of its capabilities.

3.2. Methodology

As mentioned in Section 2, this work was undertaken in two stages. First, the standard solver buoyantPimpleFoam, which is meant to solve single-phase and compressible fluids, was chosen as the base solver and adapted to be one-dimensional. During this part, in addition to the fluid equations implementation, the mesh to represent the solid structure, as well as the conduction equation, were added to the solver.

In OpenFOAM® built-in applications, there are two phenomena in the system that always need to be considered. These are the thermophysical and the turbulence model. Since this solver calculates 1D models in which the radial velocity profile is not considered, turbulence model is not included, but the turbulence effect is taken into account within the selected wall friction and the heat transfer models (see Section 2). Therefore, only the thermophysical model is developed here. In OpenFOAM®, these models are based on a pressure–temperature system, where these are the independent variables. If the ideal gas equation is not suitable for the system, the most accurate model for the evaluation of the fluid density is the Peng–Robinson (PR) equation of state (OpenFOAM, 2017c). The rest of built-in models are limited to be constant or to calculate the properties as a polynomial dependent of one variable, either pressure or temperature.

In order to get more accurate properties, this solver was extended by including a routine that allows accessing the steam tables. The steam tables are a data set that contains the thermodynamic properties of water and steam. This data set is tabulated in terms of pressure and temperature, using a wide range of values. Using the steam tables, the standard international values for the different water properties is applied. The standard used in this work was the one adopted by the International Association for the Properties of Water and Steam (IAPWS) in 1997 (IAPWS, 2008). This organism is responsible for the international standards for thermophysical properties. The routine implemented that permits the access to these tables was based on a previous work (IAPWS-IF97, 1997). This work included the necessary commands to call the steam tables from the solver and makes the definition available for OpenFOAM® solvers of the compressible family. This conversion is necessary because the steam tables are stored in a library written in C language and called *freeSteam* (Pye, 2004). Therefore, the main purpose of this extension is to translate the information so that the new 1D solver can use it. Section 3.4 explain the conversion in detail.

Once the one-dimensional single-phase solver was able to simulate single-phase flows where heat is involved and water properties were obtained from the steam tables, the work was moved to a second stage. In this case, the solver twoPhaseEulerFoam is the base solver, but the equations were replaced so that it worked as a one-dimensional solver and the mass transfer between phases was implemented, along with the rest of closure models and solid conduction equation. As mentioned before, last stage was the verification of this solver so that it was able to work for single-phase or two-phase simulations.

3.3. Solid domain implementation

Besides single- and two-phase flows simulations, the new one-dimensional solver was required to be able to simulate the radial temperature distribution within a solid in contact with the fluid in those cases where heat transfer plays an important role. There are components that needs to know the solid behavior to get the whole

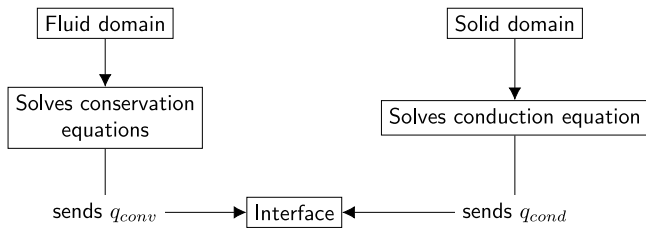


Fig. 1. Mesh structure in the new solver.

perspective of the heat transfer phenomena. This is the case of heat exchangers or fuel rods in nuclear field, but also in chemical field, heat sinks or in combustion engines simulations one can find other examples.

Even though the wall temperature can be calculated by using a wall heat flux model as boundary condition without the necessity of simulating a solid, such as the models proposed by Steiner et al. (2005) or Kurul and Podowski (1990), in these cases the feedback between domains is lost. Furthermore, the temperature and heat flux distributions on the boundary depend not only on the thermal properties and the flow characteristics of the fluid, but also on the properties of the wall Barozzi and Pagliarini (1985), and in most physically realistic situations, however, the boundary conditions at the interface are not known a priori, but depend on the coupled conduction–convection mechanism.

Therefore, an extra domain was included in the solver to consider the conduction phenomena in a wall. Thus, if a solid exists, its geometry and mesh are added to the fluid model as an independent structure which exchanges information with the fluid. The solid domain is also solved for one dimension, but in this case the heat conduction is calculated in the direction perpendicular to the surface.

Hence, the solid domain is considered by using an extra mesh that allows calculating the conjugate heat transfer in the solver (CHT). This method consists in coupling the solid conduction equation to the fluid energy equation. This feature gives the solver powerful capabilities to simulate the heat transfer, providing a complete perspective of the fluid–solid interaction, so a better understanding of the process can be acknowledged. Different authors applied the conjugate heat transfer to the two-phase flow, like (Welch and Rachidi, 2002), which applied the conjugate heat transfer for film boilings in horizontal pipes, obtaining good agreements. Also the authors in Yapıcı and Albayrak (2004) calculated the temperature and stress ratio distributions inside the pipe wall for uniform and non-uniform heat flux boundary condition. This method has also been applied in PWR pressurizer surge line (Jo and Kang, 2010), to realistically simulate the thermally stratified flow in the pipe, and in Ateş et al. (2010) a parametric study was done to demonstrate the strong influence effects of four defining parameters namely, wall thickness ratio, wall-to-fluid thermal conductivity ratio, wall-to-fluid thermal diffusivity ratio and the Peclet number in the final wall temperature.

The solid mesh implemented consists in a hexahedral mesh which dimensions are defined by the user in the input. It was implemented as a different domain, so two separate domains must be modeled in this case. However, the surface of the solid domain in contact with the fluid must be particularly named so that the code can identify it. Conservation equations are solved in the fluid region while the conduction equation is calculated in the solid domain. It remains to link both systems so that they exchange the required information to solve them considering the influence of the other system. Fig. 1 shows the relation between meshes.

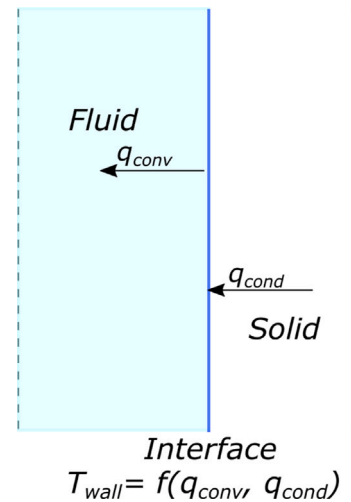


Fig. 2. Fluid–solid feedback.

3.3.1. Interface boundary condition

One of the critical points in this implementation was the calculation of the surface temperature at the interface between fluid and solid. This condition was developed to the single-phase solver and then it was adapted to the two-phase flow application. Numerical methods for the solution of the conjugate problem with single-phase fluids are well established. However, analysis of CHT in fluids involving boiling are much more challenging. The wall surface in contact to the fluid, which is the interface, acts as a boundary for coupled Navier–Stokes and energy equations in both phases. Moreover, the point where boiling starts is not known a priori and must be found as part of the solution procedure. The discontinuity across the phase interface of many flow variables is an additional disadvantage.

In order to consider both fluid convective and solid conduction heats, a heat balance was stated at the interface, so $q''_{conv} = q''_{cond}$ each time step. The term q''_{conv} represents the convective heat flux, equivalent to q'' for single-phase flows and to q''_{SB} for two-phase flows. On the other hand, q''_{cond} is the conduction heat flux and it was introduced in Eq. (47). Fig. 2 shows the information provided for each system to the interface.

Recalling this both formulations, the balance can be written as

$$\lambda \cdot (T_{wall} - T_l) + \frac{\kappa(T_{wall} - T_{cell})}{\Delta y} = 0, \quad (48)$$

where T_{wall} is the temperature at the interface and T_{cell} is the temperature at the center of the solid cell adjacent to the interface. Developing and rewriting Eq. (48), the wall temperature at the surface (T_{wall}) is given by the following

$$T_{wall} = \left(\frac{1}{1+C} T_l + \left(1 - \frac{1}{1+C}\right) T_{cell} \right), \quad (49)$$

where $C = \kappa/(\lambda\Delta y)$. T_l is the liquid temperature of the adjacent fluid cell at the time n , according to Fig. 3. T_{cell} is the adjacent solid cell for this time step $n + 1$, so this boundary condition is solved in a semi-implicit mode. Including this information in the conduction equation, the wall temperature is solved. Therefore, once the balance is achieved at the interface, the wall temperature reaches its converged value. It is worth noting that in simulations where void bubbles are concentrating in the wall due to subcooled boiling, the heat transfer coefficient term in Eq. (48), λ should account for the convective heat and also for the fraction of subcooled heat that is used to evaporation. Being λ_{sub} the heat transfer coefficient that accounts for the evaporation heat, then the term C will change to $C = \kappa/[(\lambda + \lambda_{sub})\Delta y]$. The heat transfer coefficient due to subcooled heat is calculated as

$$\lambda_{sub} = \frac{q''_{SB}}{T_w - T_{l,sat}}. \quad (50)$$

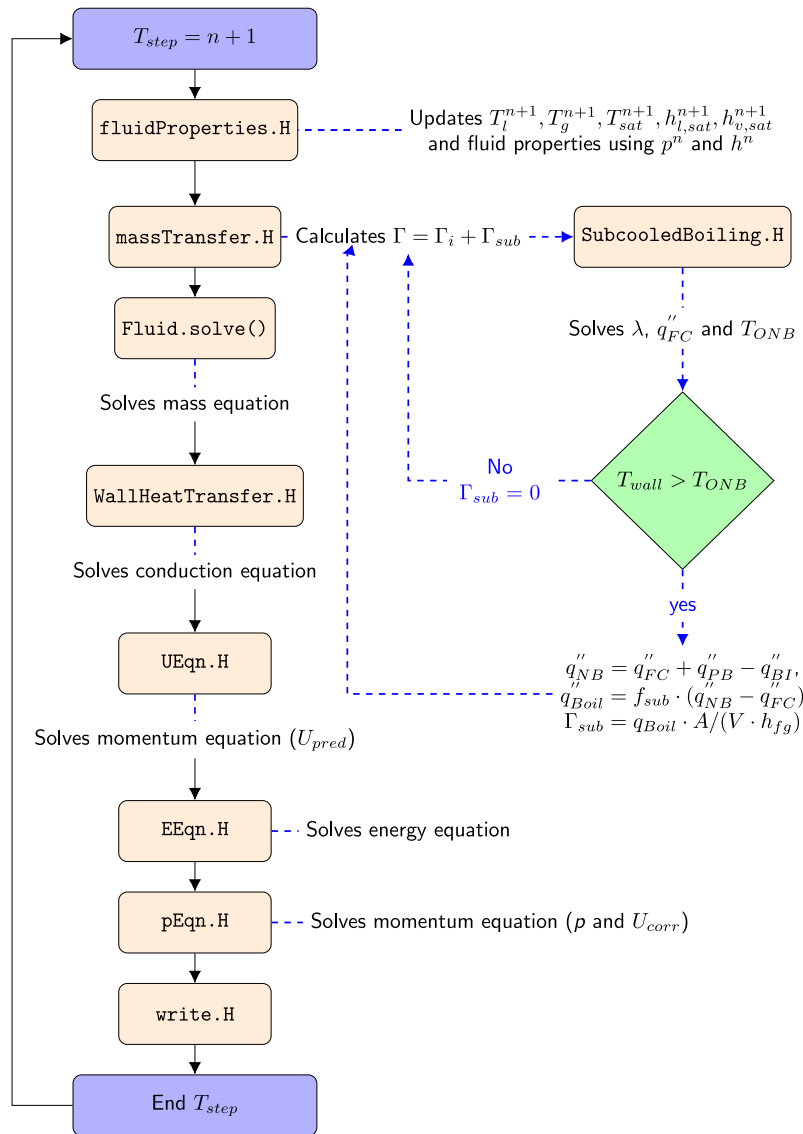


Fig. 3. New solver loop.

3.4. Steam tables IAPWS-IF97

The standard thermophysical models in OpenFOAM[®] are used for defining how the energy, heat and physical properties of the fluid are calculated. These variables are determined using models that are relations of a pressure–temperature equation system and then adding an extra equation to complete the information, such as the equation of ideal gases to calculate density or the janaf thermodynamic table (temperature dependent thermochemical tables that are a compilation of properties of different substances, J.T. Tables, 1974) to calculate properties as a function of temperature.

System codes get fluid properties from steam tables, either from polynomial functions or tabulated values, in both cases using two thermodynamic variables as inputs parameters, pressure and temperature for instance. Using this methodology, properties at saturation conditions can also be obtained.

In this work, an external application which provides IAPWS-IF97 steam tables was linked to the new solver, creating a new thermophysical model. In order to link the external tool, which was written in C-language, it was necessary to write the commands that allows the conversion between C++ and C languages. A previous work assented the basis of this work (IAPWS-IF97, 1997), but it was developed only

for single-phase contribution. In this work, these commands were extended that it could provide fluid and vapor properties at saturation conditions.

Therefore, at the beginning of each time step, the IAPWS-IF97 application is accessed with the final pressure and temperature values obtained at the previous time step. In return, the required fluid properties are provided to the solver. Depending on the simulation, properties only for fluid phase or those for the continuous and dispersed phase are provided, along with those at saturation conditions (T_{sat} , ρ , κ or μ).

3.5. Numerical methods

The standard solution algorithm used by both the buoyantPimpleFoam and the twoPhaseEulerFoam built-in solvers is PIMPLE algorithm. This solution method is defined in OpenFOAM[®] documentation as *Large time-step transient solver*. This algorithm is basically divided into two loops, one outer loop, called pimple.loop(), that involves the second loop, called pimple.correct().

The loop pimple.loop() (outer iteration) allows getting a convergence within the time step. This means that velocity and pressure are calculated as many times as necessary to satisfy the tolerance (o residual, which is the difference between the values of a parameter in two

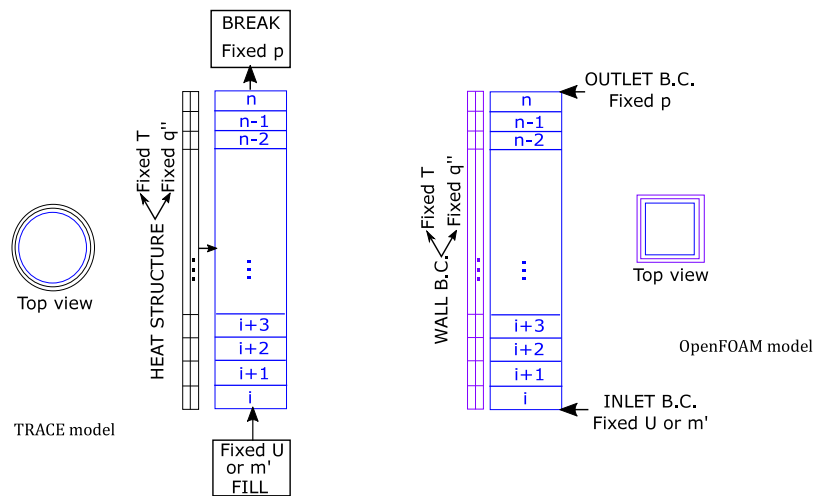


Fig. 4. Models and nodalization used in TRACE (left) and OpenFOAM (right).

consecutive time steps) criteria defined by the user during a time step. When the convergence is accomplished, `pimple.loop()` becomes *true* and the time moves on. This algorithm is specially useful in complex geometries or in cases with rapid velocity changes, which causes stiff challenges. If this option is not used, then this algorithm works as a usual transient solver, solving once the loop. This second way of working corresponds to the PISO (Pressure Implicit with Splitting of Operators) Issa (1986) algorithm and it is the mode used in this work.

Hence, the final new one-dimensional solver keeps the original algorithm and it works as typical transient, which is equivalent to perform the calculations using a typical system code, and therefore, the Courant number must be considered and kept below one. PISO algorithm applies the predictor–corrector philosophy, in which first predicts velocity and then this and pressure are corrected and energy is solved each time step.

Fig. 3 shows the final loop for a timestep. This new solver was called *my1DTPFoam*.

4. Testcases results

The solver presented in this work was verified by means of inter comparison of the results obtained for different cases. The system code TRACE was selected to carry out the comparison. Then, to show the adequacy of the results and assess the accuracy of the implemented models, a validation for both single- and two-phase flow simulations is presented in this section. In particular, the dataset of the experiment performed by Bartolomej and Chanturiya (1967) was selected for this task. All computations presented in this work were performed with the new two-phase flow solver.

Geometry for all cases follows the same methodology. In case of TRACE, the model is composed of a pipe that simulates the fluid, and two especial components used to define boundary conditions. In particular, there is a FILL, which defines fixed velocity (or mass flow) at the inlet, and a BREAK, which determines a fixed pressure at the outlet. Besides, there is a component called heat structure that plays the role of solid and allows to set a wall boundary condition at the external surface, either fixed temperature or fixed heat flux. Both the pipe and the heat structure are considered cylindrical. Final TRACE model can be seen in Fig. 4. In OpenFOAM® simulations, the geometry consists of a domain to simulate the fluid and its boundary conditions, which are defined at the surfaces. These are also a fixed velocity or mass flow at the inlet and a fixed pressure at the outlet of the pipe. There is an extra domain to simulate the solid, and the external wall boundary condition, which is defined at the external surface of the solid. However, the cells in OpenFOAM® are considered hexahedrals, so the general model for

Table 1

Initial conditions.		
Initial condition	Value	Description
Outlet pressure (Pa)	2.0725e5	–
Inlet temperature (K)	294.15	–
Inlet velocity (m/s)	0.2	–
External heat (W)	1.0e5	Wall B.C Case 1
External wall temperature (K)	375.0	Wall B.C Case 2

one dimensional simulations is the rectangular pipe, even for cylinders. In order to get the cylinder form, it should be necessary to model it in 3D. Therefore, there is always a transformation between cylinder area and tetrahedral in order to keep volumes and areas equivalent between codes. Final OpenFOAM® model for the presented simulations is shown in Fig. 4.

4.1. Single-phase flow test case

The simulation performed for the single-phase flow verification consisted in a fluid (water) running upwards through a vertical pipe and heated by an external source. The inner pipe diameter is 0.104 m, total height is 4 m and the thickness is 0.003 m. The geometry is divided into 99 uniform axial nodes for the fluid, and the solid mesh is also represented surrounding the fluid and split in 3 regular radial nodes. First, a steady state simulation has been executed in order to ensure initial conditions and then a null transient of 100 s has been run. Table 1 summarizes the initial conditions for each simulation.

This first part aims at the verification of the solid–liquid heat transfer model and the use of the steam tables application, as well as the single-phase closure equations, so the objective is focused on the verification of the fluid properties and the wall temperatures for a single-phase flow computation. Two different cases are presented so that the heat flux supply can be analyzed, one case with constant external heat flux boundary condition and a second case with a Dirichlet boundary condition (fixed outer wall temperature). First, the latter case is presented.

4.1.1. Fixed wall temperature boundary condition

Focusing on temperature, Fig. 5 shows the axial evolution of the fluid temperature. Fig. 6 displays the axial evolution of the temperature at the surface of the solid which is in contact to the fluid. Both images show good agreement when they are compared to TRACE results. Wall temperature presents an average difference of 0.05 K which is constant along the pipe, getting a maximum difference of 0.07 K at the outlet

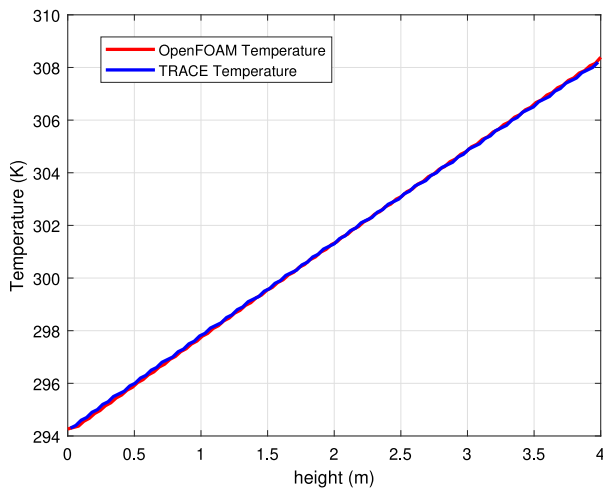


Fig. 5. Liquid temperature axial evolution.

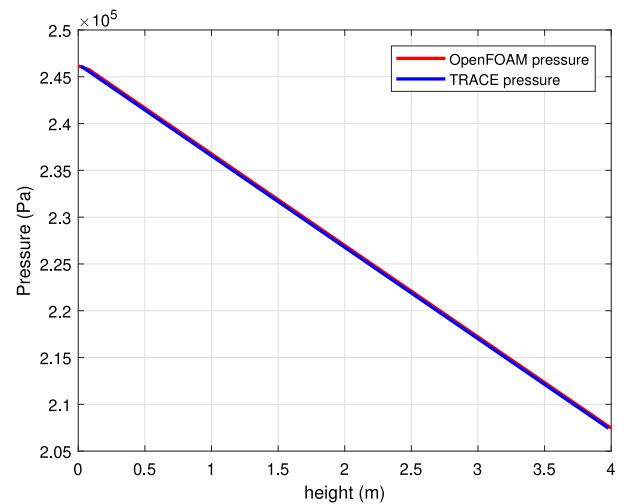


Fig. 7. Pressure axial evolution.

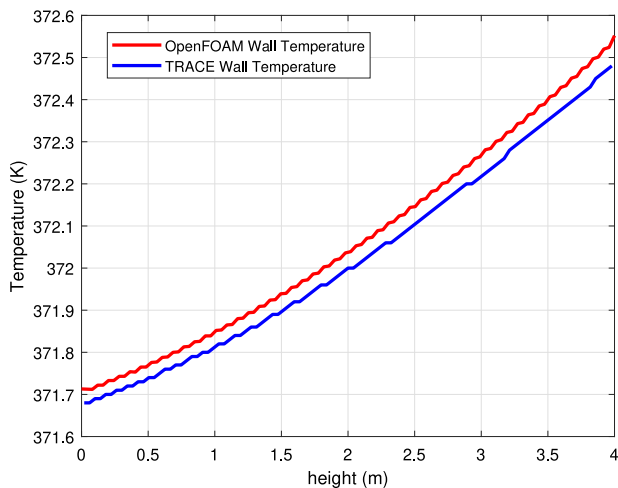


Fig. 6. Innerwall temperature axial evolution.

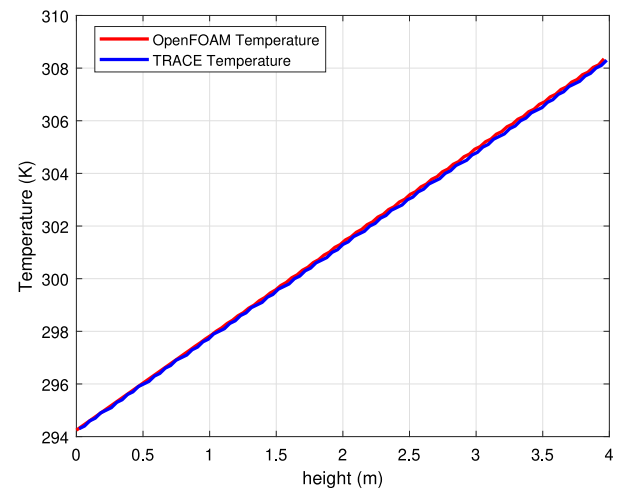


Fig. 8. Liquid temperature axial evolution.

section. Fig. 7 presents pressure axial evolution, which also exhibits a good agreement between codes, getting a maximum difference of 13 Pa at the inlet region.

4.1.2. Fixed heat flux boundary condition

Regarding to the case with fixed external heat flux, which was defined as an uniform axial distribution, Figs. 8 and 9 presents the comparison of the axial evolution of the liquid and wall temperatures. In this case, the maximum temperature difference at the fluid–solid interface between codes is 0.21 K and it is found at the inlet, but this difference decreases as the fluid reaches the outlet. The general RMS error (Root Mean Square error), which is the quadratic mean of the deviation respect from the measured value, for this parameter is 0.1 K. Regarding to the fluid temperature, both models agree and follow same evolution along the pipe, having an RMS less than 0.1 K.

Considering both fixed temperature boundary condition and fixed heat flux boundary condition cases, it can be seen that closure models related to fluid works fine, as well as the acquisition of fluid properties from tables. However, there is a slightly difference in the solid or interface between solid and liquid. In this point, three variables are contributing, the heat transfer coefficient, the solid properties and the geometry of the pipe. The properties of solid can be discarded, since they are constant and equal in both codes. Therefore, the temperature

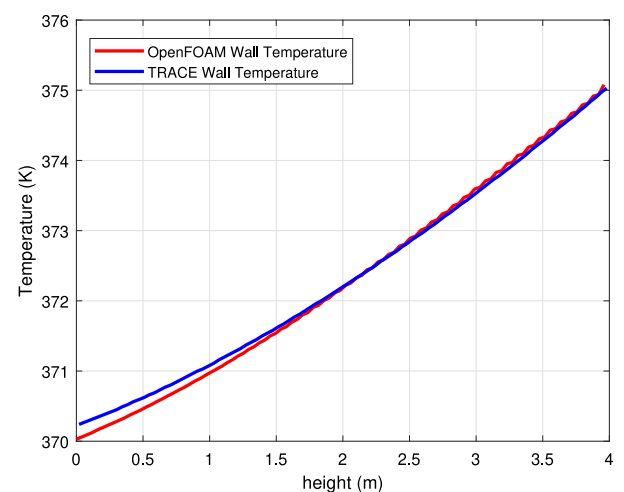


Fig. 9. Wall temperature axial evolution.

difference may be caused due to a variation in the heat transfer coefficient or due to discrepancy in the transformation performed between geometries. However, energy conservation is accomplished, since the

Table 2
Conservation of energy in fixed heat flux case.

Parameter	Value
Fluid inlet enthalpy (kJ/kg)	1.48499e5
Fluid outlet enthalpy (kJ/kg)	0.887363e5
Inlet mass flow (kg/s)	1.69698
Outlet mass flow (kg/s)	1.69558
Total heat provided to the system (W)	1.0032e5

Table 3
Fluid boundary conditions.

Variable	Inlet	Outlet	Wall
v_r	Dirichlet	Neumann	No-slip
v_θ	Dirichlet	Neumann	No-slip
p	Neumann	Dirichlet	Neumann
α	Dirichlet	Neumann	Neumann
T_r	Dirichlet	Neumann	Mixed

change in enthalpy in the fluid meets the amount of heat provided to the fluid, as shown in Table 2. Therefore, the variation in the wall temperature at the inlet comes from a variation in the wall heat transfer coefficient model. This model shall be analyzed to find the origin of this difference.

4.2. Two-phase flow test case

For the validation, the Bartolomej experiment was selected. This test was performed in a 2 m long heated tube, whose inner diameter was 15.4 mm. The heat flux provided was $5.7 \times 10^5 \text{ W/m}^2$. Regarding to the fluid conditions, the water massflow of the system was $900.0 \text{ kg/(s m}^2)$ and the inlet subcooling was set to 58.2 K. The pressure at the outlet section was 4.5 MPa. The data set of this experiment, in particular, includes also the axial wall temperature, so this comparison is also presented.

The tube was modeled in 1D geometry with 50 grid cells in axial direction. This grid is a good compromise between the numerical accuracy and the computational effort, since only axial dimension is involved and Courant number is limited to 0.95 without having discrepancies while running the case. Table 3 shows the boundary conditions selected for each parameter.

All the closure models presented in Section 2 were used in this simulations. Since Bartolomej experiment does not include local two-phase flow parameter measurements, the closure models were selected according to those used by TRACE, so that the comparison was accurate. This particular case was also compared to a 3D solver developed within OpenFOAM® framework, called *BoilEulerFOAM*, which allows considering boiling phenomena (Rollins, 2018). This 3D solver was meant as a testing platform for different scenarios, encompassing from critical heat flux situations, DNB calculations and multi-phase flow simulations where boiling occurs. It includes a wide range of interfacial models, allowing the analysis of the influence of these in the final result. It considers the RPI wall heat flux model (Kurul and Podowski, 1990) to calculate wall temperature and the fluid properties used are considered constant. The model in this solver consists of an axisymmetric geometry of 200 axial uniform cells, 25 divisions in radial direction and 1 azimuthal cells. This comparison let evaluate the amount of information lost due to the 1D simplification.

Fig. 10 shows the comparison of the axial evolution of the liquid temperature. The different trends represent the following: red line is the new 1D solver developed in OpenFOAM®, blue line represents TRACE results, magenta dashed line is *BoilEulerFOAM* simulations and the green dots shows the experimental results. Starting from the inlet, the values for the lower part of the geometry, which corresponds to the single-phase part, meets for all cases, although the results for the 3D code evolves with lower results until it reaches the subcooled portion

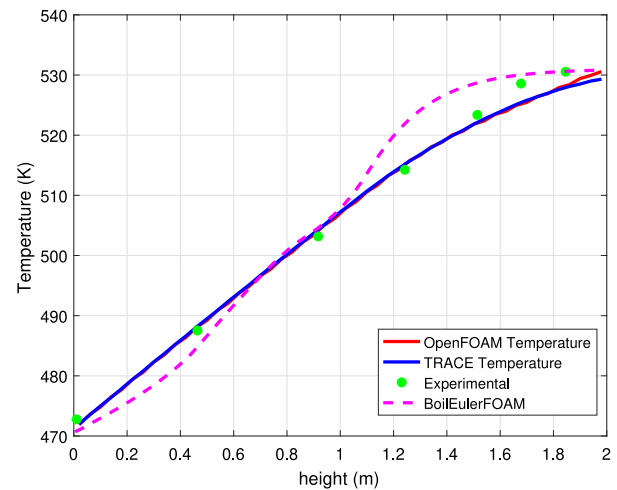


Fig. 10. Liquid temperature axial evolution in Bartolomej case.

of the pipe. From that moment, the temperature increases faster and it reaches saturation condition before the rest of cases. Regarding to *my1DTPFoam*, the evolution matches the prediction of TRACE until the last two cells, where it reaches saturation conditions while TRACE remains lower. In general, the agreement between these both curves is high except for the outlet section, where the maximum difference is reached with a value of 1.3 K. In this case, TRACE presents a similar difference with the rest of codes. It is clear that the amount of heat transferred to the fluid in the 1D solver is slightly higher than in TRACE. On the other hand, neither *my1DTPFoam* nor TRACE agree with the experimental result at the top part of the pipe, since this case gets higher temperatures during the upper quarter of the pipe until it reaches saturation. In this case, *my1DTPFoam* gets a better agreement than TRACE.

Regarding to the void fraction, which is shown in Fig. 11, one can see that the new 1D OpenFOAM® solver under predicts the void fraction during the subcooled nucleate boiling zone in comparison to TRACE. The experimental results are also in agreement with the system code, specially from 1.1 m to 1.3 m. There is a constant delay of a 2.5% in the value of void fraction of the new solver concerning TRACE which is only overcome in the upper cells, when the fluid reaches saturation conditions.

However, this reduction in the vapor generation is not reflected in the wall temperature (Fig. 12). This parameter shows a good agreement between the new 1D solver and TRACE results, both trends below experimental outcome. In this case, the 3D CFD solver presents better agreement, although slightly over-predicted. The difference at the inlet between BoilEulerFoam code and the one dimensional and experimental results is due to the radial distribution of the geometry. The author presents a sensitivity grid analysis in Rollins (2018) showing that a decrease in the number of radial nodes leads to a reduction in the difference of the wall temperature at the lower nodes. This allows to accept the verification of the new solver, but in terms of validation, a lower wall temperature should result in a higher amount of heat transferred to the water, which would result in a greater value of void fraction. Therefore, the subcooled boiling model should be revised to ensure that it is not under predicting the void fraction.

In general, 3D solver numerical solution has a better agreement with experimental data, especially in temperature parameter. However, the mesh is 50 times bigger, consequently it consumes more resources. On the other hand, the results obtained by the 1D solver agrees in general with the experimental data and the system code, consuming less resources. Therefore, the new code presents an alternative to reduce time and obtain correct results in those geometries or portion of

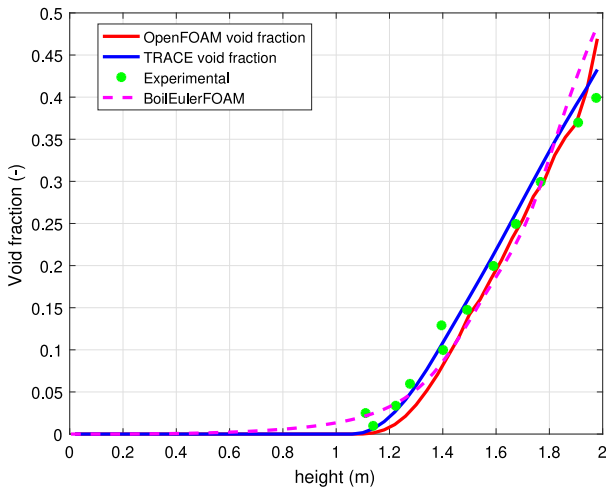


Fig. 11. Void fraction axial evolution in Bartolomej case.

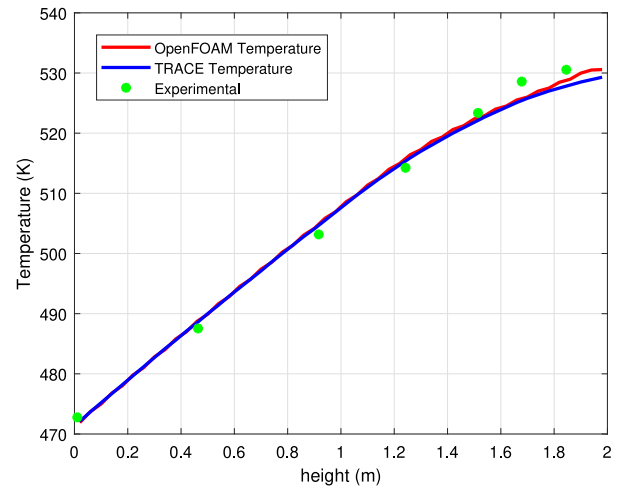


Fig. 13. Liquid temperature axial evolution with Dirichlet B.C.

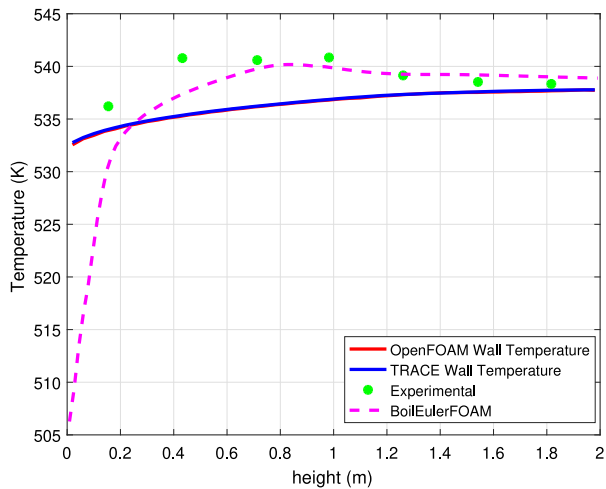


Fig. 12. Wall temperature axial evolution at the fluid-solid interface in Bartolomej case.

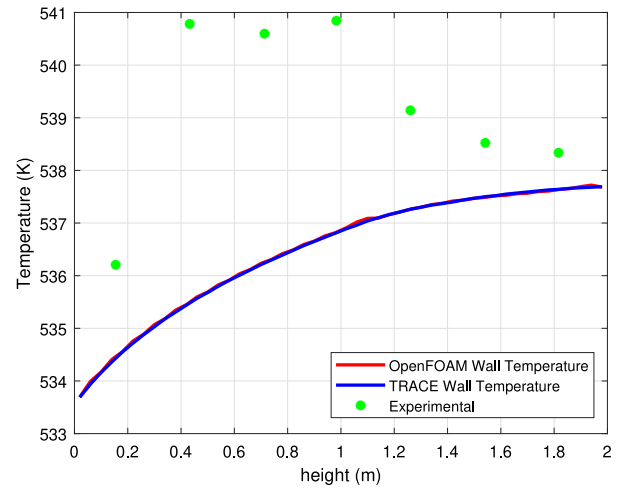


Fig. 14. Wall temperature axial evolution with Dirichlet B.C.

geometries where the accuracy on the results is not essential. Hence, this new solver meets the conditions to be coupled to a 3D solver optimizing the computational resources.

4.2.1. Two-phase flow with fixed temperature boundary condition

Besides the validation case, a different case using the Dirichlet boundary condition at the external wall was also simulated. The value of the temperature fixed at the boundary was chosen so that the heat flux provided to the fluid was kept equivalent to the experimental case. The rest of the conditions remained the same as in the previous case. This performance allows disregarding different potential parameters of failure in order to revise the solver and find the causes of the difference in the results respect TRACE and the experimental values, specially in the void fraction variable.

Fig. 13 shows the axial evolution of the temperature for the last time step. The simulated results are compared to the experimental values, since the equivalent heat flux provided to the fluid remains the same as in the previous case. The evolution between computer codes agrees in a high level, up to a height of 1.8 m, where the fluid in the new solver is overheated, reaching a maximum difference of 1.05 K. This overheating is perceived along the whole pipe, which implies that the solid is dissipating more heat. However, focusing on the outlet section, one can see that the fluid, according to the experiment, reaches saturation temperature, as happens in the new 1D solver.

Regarding to the behavior found at the surface between liquid and solid, which is shown in Fig. 14, the results calculated by both codes agree in a high level, showing a RMS of 0.045 K. However, when they are compared to the experimental wall temperature, the behavior shown is different, since the wall temperature at the device reaches higher values. This means a larger dissipation of heat in the codes, which do not correspond to the experiment, so the wall heat flux model should be analyzed and compared to different models to look for the most accurate correlation. Measurement error in this variable might provide a better understanding of the deviations and the degree of acceptance of these results. However, due to the lack of knowledge of these errors, in this work the reference code has been set as the proof to accept the obtained results.

Lastly, Fig. 15 shows the void fraction axial evolution. It can be seen that the new 1D solver rather under predicts this variable, which somewhat differs with the rest of parameters, where the variables are properly adjusted or rather over-predicted. The higher the temperature in the fluid, the higher the vapor generation should be. In this case, the over-prediction can be found at the outlet section of the pipe, when the fluid reaches saturation temperature. Similar behavior can be found when the new 1D solver is compared to the experimental values, which rather approaches to TRACE calculations. The computer calculations are under-predicted except for the outlet section, where the simulation changes the pattern and increases significantly, becoming higher than the experimental value.

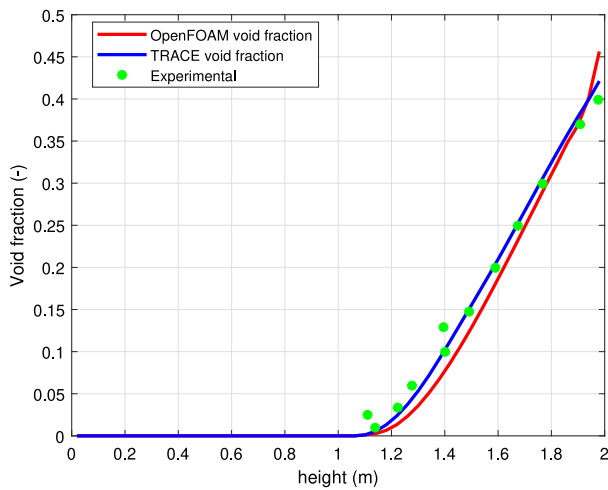


Fig. 15. Void fraction axial evolution with Dirichlet B.C.

Overall, it can be concluded that the case with a fixed temperature boundary condition provides good results in general in comparison to TRACE. This means that closure models are suitable for this simulations and the solver in general can be verified. Subcooled boiling model and mass transfer term in continuity equation should be deeply analyzed, since the differences in the case with fixed heat flux boundary condition could be a consequence of a not adjusted balance.

Both cases have a correct trend all variables, but when the fluid reaches the subcooled nucleate boiling state, there are discrepancies between codes that result in over-predictions in the new solver.

5. Conclusions

A new solver for one-dimensional simulations was implemented within the framework of OpenFOAM[®], an open-sourced CFD library. This solver departed from a standard solver of OpenFOAM library, called twoPhaseEulerFoam, which was implemented for solving two-phase flow simulations with an Eulerian–Eulerian model. This solver was modified so that only one direction was considered and different closure models for one-dimensional simulations were implemented. Furthermore, a subcooled boiling model was added to undertake accurate computations of the mass transfer contribution between phases and its influence in the heat transfer. In addition to the fluid calculations, the solver is able to solve conduction equation in the radial direction of a solid structure that surrounds the fluid.

In order to verify the solver, first single-phase flow simulation testcases were presented and their results were compared to those obtained with the system code TRACE for the same testcases. These cases consisted in the same problem were each one had a different wall boundary condition, so the heat transfer between solid and liquid was verified regardless the external condition. These results showed a good agreement for the fluid properties, so there is a good adequacy in the closure models of both codes. Wall temperature was also compared, having slightly differences between code results, specially at the inlet.

Regarding to the two-phase flow simulations, the testcase performed was also verified against TRACE and it was validated using an international experimental dataset. In particular, Bartolomej dataset was used. This experiment allows the validation of the subcooled boiling for averaged values, which in this case the Gorenflo model was selected to be implemented in the new solver. According to the obtained results, the void fraction is rather under-predicted, whereas the fluid and wall temperature presents a good agreement. Since the void fraction is mainly influenced by the subcooled boiling model, this should be reviewed and adjusted to avoid disagreements. The mass transfer term

in continuity equation should be also assessed, since it is also strongly related to the void fraction calculation.

Overall, it can be concluded that the solver provides results that are numerically and physically realistic. Once this solver is validated for one-dimensional simulations in both single-phase and two-phase flows, the following step is the coupling with a 3D solver within the same CFD platform. This coupling will save external communication interfaces making the methodology simpler and easier than the coupling with system codes.

CRedit authorship contribution statement

C. Gómez-Zarzuela: Conceptualization, Investigation, Software, Methodology, Writing – original draft preparation, Visualization. **C. Peña-Monferrer:** Conceptualization, Investigation, Methodology, Reviewing and Editing. **S. Chiva:** Methodology, Investigation, Writing - reviewing and editing. **R. Miró:** Investigation, Methodology, Writing - reviewing and editing, Supervision, Funding acquisition, Resources – Computing resources.

Declaration of competing interest

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

Acknowledgments

This work has been partially supported by the Spanish Agencia Estatal de Investigación [grant number BES-2013-064783], and the Spanish Ministerio de Economía Industria y Competitividad [project NUC-MULTPHYS ENE2012- 34585].

References

- Anglart, H., Nylund, O., 1996. CFD application to prediction of void distribution in two-phase bubbly flows in rod bundles. *Nucl. Eng. Des.* 163 (1–2), 81–98.
- Ateş, A., Darıcı, S., Şefik Bilir, 2010. Unsteady conjugated heat transfer in thick walled pipes involving two-dimensional wall and axial fluid conduction with uniform heat flux boundary condition. *Int. J. Heat Mass Transfer* (ISSN: 0017-9310) 53 (23), 5058–5064. <http://dx.doi.org/10.1016/j.ijheatmasstransfer.2010.07.059>, URL <http://www.sciencedirect.com/science/article/pii/S0017931010004400>.
- Barozzi, G.S., Pagliarini, G., 1985. A method to solve conjugate heat transfer problems: The case of fully developed laminar flow in a pipe. *J. Heat Transfer* 107 (1), 77–83. <http://dx.doi.org/10.1115/1.3247406>.
- Bartolomej, G., Chanturiya, V., 1967. Experimental study of true void fraction when boiling subcooled water in vertical tubes. *Therm. Eng.* 14 (2), 123–128.
- Basu, N., Warriar, G.R., Dhir, V.K., 2002. Onset of nucleate boiling and active nucleation site density during subcooled flow boiling. *J. Heat Transfer* 124 (4), 717–728. <http://dx.doi.org/10.1115/1.1471522>.
- Bertolotto, D., Manera, A., Frey, S., Prasser, H.-M., Chawla, R., 2009. Single-phase mixing studies by means of a directly coupled CFD/system-code tool. *Ann. Nucl. Energy* 36 (3), 310–316. <http://dx.doi.org/10.1016/j.anucene.2008.11.027>, PHYSOR 2008.
- Brooks, C.S., Hibiki, T., Ishii, M., 2012. Interfacial drag force in one-dimensional two-fluid model. *Prog. Nucl. Energy* 61, 57–68.
- Chen, J.C., 1966. Correlation for boiling heat transfer to saturated fluids in convective flow. *Ind. Eng. Chem. Process Des. Dev.* 5 (3), 322–329.
- Churchill, S., 1977. Friction-factor equation spans all fluid-flow regimes. *Chem. Eng.* 91–92.
- Colombo, M., Fairweather, M., 2016. Accuracy of Eulerian Eulerian two fluid CFD boiling models of subcooled boiling flows. *Int. J. Heat Mass Transfer* 103, 28–44.
- Corzo, S., Damian, S.M., Ramajo, D., Nigro, N., 2012. Numerical simulation of bubbly two-phase flow using eulerian-eulerian model. *Asoc. Argent. Mec. Comput.* 31, 85–112.
- Drzewiecki, T.J., Asher, I.M., Grunloh, T.P., Petrov, V.E., Fidkowski, K.J., Manera, A., Downar, T.J., 2012. Parameter sensitivity study of boiling and two-phase flow models in CFD. *J. Comput. Multiph. Flows* (ISSN: 1757-482X) 4 (4), 411–425. <http://dx.doi.org/10.1260/1757-482x.4.4.411>.
- Filonenko, G., 1954. Hydraulic resistance in pipes. *Teploenergetika*.
- Gnielinski, V., 1976. New equations for heat and mass transfer in turbulent pipe and channel flow. *Int. Chem. Eng.* 16 (2), 359–368.

- Godino, D.M., Corzo, S.F., Nigro, N.M., Ramajo, D.E., 2018. Cfd simulation of the pre-heater of a nuclear facility steam generator using a thermal coupled model. *Nucl. Eng. Des.* 335, 265–278.
- Gorenflo, D., Baumhögger, E., Herres, G., Kotthoff, S., 2014. Prediction methods for pool boiling heat transfer: A state-of-the-art review. *Int. J. Refrig.* 43, 203–226.
- Holman, J., 1997. McGrawHill, New York, USA.
2008. IAPWS Industrial formulation 1997 for the thermodynamic properties of water and steam. In: *International Steam Tables: Properties of Water and Steam Based on the Industrial Formulation IAPWS-IF97*. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 7–150. http://dx.doi.org/10.1007/978-3-540-74234-0_3.
- IAPWS-IF97 water properties for OpenFOAM, URL <https://github.com/romansCode/IAPWS-IF97-OF>.
- Ishii, M., Hibiki, T., 2011. *Thermo-Fluid Dynamics of Two-Phase Flow*. Springer.
- Ishii, M., Mishima, K., 1984. Two-fluid model and hydrodynamic constitutive relations. *Nucl. Eng. Des.* 82 (2–3), 107–126.
- Issa, R.I., 1986. Solution of the implicitly discretised fluid flow equations by operator-splitting. *J. Comput. Phys.* 62 (1), 40–65.
- Jeong, J.H., Han, B.-S., 2008. Coolant flow field in a real geometry of PWR downcomer and lower plenum. *Ann. Nucl. Energy* 35 (4), 610–619.
- Jo, J.C., Kang, D.G., 2010. CFD analysis of thermally stratified flow and conjugate heat transfer in a PWR pressurizer surge line. *J. Press. Vessel Technol.* (ISSN: 0094-9930) 132 (2), <http://dx.doi.org/10.1115/1.4000727>, <https://doi.org/10.1115/1.4000727>.
- J.T. Tables, 1974. (Nsrdsnbs 37, 1971). *J. Chem. Phys. Ref. Data* 3 (311), 4.
- Kataoka, I., Ishii, M., 1987. Drift flux model for large diameter pipe and new correlation for pool void fraction. *Int. J. Heat Mass Transfer* 30 (9), 1927–1939.
- Krepper, E., Konar, B., Egorov, Y., 2007. CFD modelling of subcooled boiling. Concept, validation and application to fuel assembly design. *Nucl. Eng. Des.* (ISSN: 0029-5493) 237 (7), 716–731. <http://dx.doi.org/10.1016/j.nucengdes.2006.10.023>, URL <http://www.sciencedirect.com/science/article/pii/S0029549306006248>.
- Kurul, N., 1991. *Multidimensional Effects in Two-Phase Flow Including Phase Change* (Ph.D. thesis). Rensselaer Polytechnic Institute.
- Kurul, N., Podowski, M.Z., 1990. Multidimensional effects in forced convection subcooled boiling. In: *International Heat Transfer Conference Digital Library*. Begel House Inc..
- Lahey Jr., R.T., 1978. A mechanistic subcooled boiling model. In: *International Heat Transfer Conference Digital Library*. Begel House Inc..
- Lee, G.H., Bang, Y.S., Woo, S.W., Cheong, A.J., 2014. Comparative study on the effect of reactor internal structure geometry modeling methods on the prediction accuracy for PWR internal flow distribution. *Ann. Nucl. Energy* 70, 208–215.
- Li, W., Wu, X., Zhang, D., Su, G., Tian, W., Qiu, S., 2014. Preliminary study of coupling CFD code FLUENT and system code RELAP5. *Ann. Nucl. Energy* 73, 96–107.
- Mahaffy, J., Chung, B., Dubois, F., Ducros, F., 2007. *Best Practice Guidelines for the use of CFD in Nuclear Reactor Safety Applications*. Tech. Rep., NEA/CSNI.
- OpenFOAM: User Guide v1712. The open source CFD toolbox, URL <https://www.openfoam.com/documentation/guides/latest/man/twoPhaseEulerFoam.html>.
- OpenFOAM: User Guide v1712. The open source CFD toolbox, URL <https://www.openfoam.com/documentation/guides/latest/doc/guide-applications-solvers-heat-transfer-buoyantPimpleFoam.html>.
- OpenFOAM®: Open source CFD: Documentation, URL <https://www.openfoam.com/documentation/guides/latest/man/twoPhaseEulerFoam.html>.
- Pye, J., Freesteam, URL <http://freesteam.sourceforge.net/>.
- Ranz, W., Marshall, W.R., et al., 1952. Evaporation from drops. *Chem. Eng. Prog.* 48 (3), 141–146.
- Rohsenow, W.M., Hartnett, J.P., Cho, Y.I., et al., 1998. *Handbook of Heat Transfer*, Vol. 3. McGraw-Hill New York.
- Rollins, C., 2018. *Development of Multiphase Computational Fluid Dynamics Solver in OpenFOAM* (Ph.D. thesis). North Carolina State University.
- Smith, B., Andreani, M., Bieder, U., Ducros, F., 2015. Assessment of CFD Codes for Nuclear Reactor Safety Problems - Revision 2. Tech. Rep., NEA/CSNI.
- Steiner, H., Kobar, A., Gebhard, L., 2005. A wall heat transfer model for subcooled boiling flow. *Int. J. Heat Mass Transfer* (ISSN: 0017-9310) 48, 4161–4173.
- Todreas, N.E., Kazimi, M.S., 2011. *Nuclear Systems Volume I: Thermal Hydraulic Fundamentals*. CRC press.
- Toti, A., Vierendeels, J., Belloni, F., 2017. Improved numerical algorithm and experimental validation of a system thermal-hydraulic/CFD coupling method for multi-scale transient simulations of pool-type reactors. *Ann. Nucl. Energy* 103.
- Tu, J., Yeoh, G., 2002. On numerical modelling of low-pressure subcooled boiling flows. *Int. J. Heat Mass Transfer* (ISSN: 0017-9310) 45 (6), 1197–1209. [http://dx.doi.org/10.1016/S0017-9310\(01\)00230-7](http://dx.doi.org/10.1016/S0017-9310(01)00230-7), URL <http://www.sciencedirect.com/science/article/pii/S0017931001002307>.
- United State Nuclear Regulatory Commission, 2013. TRACE V5. 840 Theory Manual. Division of Safety Analysis, US NRC.
- Welch, S.W.J., Rachidi, T., 2002. Numerical computation of film boiling including conjugate heat transfer. *Numer. Heat Transfer B* 42 (1), 35–53. <http://dx.doi.org/10.1080/10407790190053824>.
- Weller, H.G., Tabor, G., Jasak, H., Fureby, C., 1998. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Comput. Phys.* 12 (6), 620–631.
- Yapıcı, H., Albayrak, B., 2004. Numerical solutions of conjugate heat transfer and thermal stresses in a circular pipe externally heated with non-uniform heat flux. *Energy Convers. Manage.* (ISSN: 0196-8904) 45 (6), 927–937. [http://dx.doi.org/10.1016/S0196-8904\(03\)00195-X](http://dx.doi.org/10.1016/S0196-8904(03)00195-X), URL <http://www.sciencedirect.com/science/article/pii/S019689040300195X>.
- Yeoh, G., Tu, J., 2006. Numerical modelling of bubbly flows with and without heat and mass transfer. *Appl. Math. Model.* 30 (10), 1067–1095. <http://dx.doi.org/10.1016/j.apm.2005.06.012>, Special issue of the 12th Biennial Computational Techniques and Applications Conference and Workshops (CTAC-2004) held at The University of Melbourne, Australia, from 27th September to 1st October 2004. URL <http://www.sciencedirect.com/science/article/pii/S0307904X05001678>.
- Zuber, N., Findlay, J.A., 1965. Average volumetric concentration in two-phase flow systems. *J. Heat Transfer* 87 (4), 453. <http://dx.doi.org/10.1115/1.3689137>, URL <http://heattransfer.asmedigitalcollection.asme.org/article.aspx?articleid=1433615>.