NASA/TP—2014–216659

Physics Based Model for Cryogenic Chilldown and Loading.

Part I: Algorithm

Author Name: Dmitry, G, Luchinsky
Mission Critical Technologies Inc.
Ames Research Center, Moffett Field, California

Vadim, N, Smelyanskiy
Ames Research Center, Moffett Field, California

Barbara Brown
Kennedy Space Center, Florida

February 2014

NASA STI Program ... in Profile

Since its founding, NASA has been dedicated to the advancement of aeronautics and space science. The NASA scientific and technical information (STI) program plays a key part in helping NASA maintain this important role.

The NASA STI program operates under the auspices of the Agency Chief Information Officer. It collects, organizes, provides for archiving, and disseminates NASA's STI. The NASA STI program provides access to the NASA Aeronautics and Space Database and its public interface, the NASA Technical Reports Server, thus providing one of the largest collections of aeronautical and space science STI in the world. Results are published in both non-NASA channels and by NASA in the NASA STI Report Series, which includes the following report types:

- TECHNICAL PUBLICATION. Reports of completed research or a major significant phase of research that present the results of NASA Programs and include extensive data or theoretical analysis. Includes compilations of significant scientific and technical data and information deemed to be of continuing reference value. NASA counterpart of peer-reviewed formal professional papers but has less stringent limitations on manuscript length and extent of graphic presentations.

- TECHNICAL MEMORANDUM. Scientific and technical findings that are preliminary or of specialized interest, e.g., quick release reports, working papers, and bibliographies that contain minimal annotation. Does not contain extensive analysis.

- CONTRACTOR REPORT. Scientific and technical findings by NASA-sponsored contractors and grantees.

- CONFERENCE PUBLICATION. Collected papers from scientific and technical conferences, symposia, seminars, or other meetings sponsored or co-sponsored by NASA.

- SPECIAL PUBLICATION. Scientific, technical, or historical information from NASA programs, projects, and missions, often concerned with subjects having substantial public interest.

- TECHNICAL TRANSLATION. English-language translations of foreign scientific and technical material pertinent to NASA's mission.

Specialized services also include organizing and publishing research results, distributing specialized research announcements and feeds, providing information desk and personal search support, and enabling data exchange services.

For more information about the NASA STI program, see the following:

- Access the NASA STI program home page at http://www.sti.nasa.gov

- E-mail your question to help@sti.nasa.gov

- Fax your question to the NASA STI Information Desk at 443-757-5803

- Phone the NASA STI Information Desk at 443-757-5802

- Write to:
  STI Information Desk
  NASA Center for AeroSpace Information
  7115 Standard Drive
  Hanover, MD 21076-1320

NASA/TP—2014–216659

Physics Based Model for Cryogenic Chilldown and Loading
Part I: Algorithm

Author Name: Dmitry, G, Luchinsky
Mission Critical Technologies Inc.
Ames Research Center, Moffett Field, California

Vadim, N, Smelyanskiy
Ames Research Center, Moffett Field, California

Barbara Brown
Kennedy Space Center, Florida

National Aeronautics and
Space Administration

Ames Research Center
Moffett Field, CA 94035-1000

February 2014

Acknowledgments

We would like to acknowledge valuable comments and stimulating discussions by Dr. Cetin Kiris (NASA-Ames Research Center) and Dr. Charlie Goodrich (NASA-Kennedy Space Center).

This report is also available in electronic form at

http://

Introduction

This work is motivated by NASA plans to develop and maturate technology for autonomous cryogenic propellants management (CPM) on the ground and in space. The latter can be defined as the capability to accomplish launch vehicle propellant load and drain without human interaction. Importantly, the system and subsystems should be able to change their behavior in response to un-anticipated events. The solution of this problem requires a deployment of an intelligent control system that can optimize loading protocol, predict various deviations from the nominal regime, detect, localize, and mitigate faults online.

The difficulties in solving this problem stem from the complexity of the transient analysis of the two-phase flow (TPF) under strongly non-equilibrium conditions. The application of such analysis to cryogenic propellant management poses additional complications due to uncertainties of flow regime and heat transfer characteristics of the cryogenic fluids. These uncertainties are even larger under microgravity condition imposing severe design concern for the development of the autonomous CPM in space.

To solve this problem and to enable online autonomous control of cryogenic loading we have developed a concept of physics based automated loading operation in space and on the ground that combines hierarchy of models with best industrial practice and deep analytical insight into the nature of flow patterns, frictional losses and heat transfer correlations. The model hierarchy is ranging from the fastest online fault detection, isolation, and mitigation (FDI&M) model to the full 3D supercomputer model on the ground and incorporates (worth of 50 years R&D) best practice of nuclear reactor and refrigerator industrial modeling frameworks for the automated control of the two-phase flow and heat transfer.

Here we report the progress in development of one of the key component of the model hierarchy, namely one-dimensional model of fully separated non-equilibrium two-phase cryogenic flow that allows for real time control of transient response of the system during chilldown and loading.

In developing this model, we took into account additional limitations imposed on the program by constraint resources and time. We recognized that the development of multiphase cryogenic flow models capable of predicting flow regimes and heat transfer at low gravity is one of the top priority recommendations made by NRC Committee [1] for successful NASA Space Program. In these recommendations, it was also pointed out [1] that such models should incorporate best practice of autonomous thermal-hydraulic management developed by the Department of Energy.

Indeed, cryogenic chilldown and loading shares many common transients with other industrial processes [2] such as loss of coolant and other anticipated transients without scram[1]. Many transient characteristics of system failure during cryogenic loading are also common with industrial processes including, e.g. loss of coolant, loss of feeding fluid, loss of offsite power, station blackout etc. We note in addition that the latest in the series of nuclear reactor codes are highly generic and can be used for simulation and control of a wide variety of hydraulic and thermal transients in both nuclear and nonnuclear systems involving mixtures of vapor, liquid, non-condensable gases, and nonvolatile solute.

In what follows, we briefly formulate a general transient two-phase flow problem and list the key difficulties in its solution. Next, we briefly discuss the hierarchy of models that can be used to accomplish online control of this flow in various approximations.

[1] the failure of the emergency shutdown system

Two-phase flow problem

At present, the predictive capability and physical understanding of the two-phase flow in industrial systems must rely heavily on theoretical and computational models [3]. Major theoretical efforts here are required to find functional form of correlations that cover the diversity of multiphase flows patterns, friction and heat transfer characteristics.
For autonomous control the model has to be efficient enough to detect and localize faults in real time. We therefore restrict our discussion to one-dimensional models that satisfy this requirement. We note though that the models discussed below can be extended to 3D version if required (see e.g. [2], [4], [5]).

We note that the hierarchy of one-dimensional models of two-phase flow is diverse. More importantly, an efficient online fault detection, isolation, and mitigation must involve optimal utilization of a few members of this hierarchy for every particular fault. For example, the most accurate model (separated non-equilibrium two-phase flow) at the top of this hierarchy should be responsible for real time FDI and providing recommendations for the most likely cause of the fault and possible mitigation strategies, therefore restricting search space for the fault and control parameters. The fastest model at the bottom of this hierarchy (incompressible isothermal flow network) should be responsible for providing guess for the initial conditions and fast estimations of pressure losses and flow rates. The intermediate level model (e.g. moving front model for homogeneous flow) performs evaluation of relatively slow thermal fault transients and search for the optimal mitigation strategies.

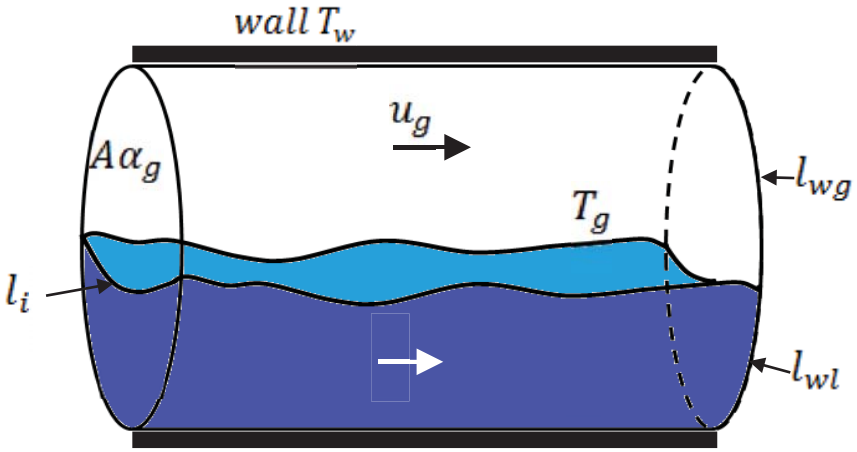

Figure 1. Control volume of the two-phase flow. $\alpha_{g,l}$ void fractions for gas (g) and liquid (l). $T_{g,l}$ phasic temperatures. $u_{g,l}$ phasic velocities. $l_{g,l}$ wetted perimeter for each phase. l_i perimeter of the interface. A is the cross-sectional area. T_w is the wall temperature.

The relation between the models in the hierarchy their capabilities and the conceptual architecture of the intelligent online control module should become more transparent in the following sections after we describe generic two-phase flow models and basic solution algorithms.

Generic two-phase flow model

The governing equations for effective-field formulation of multi-fluid/multi-phase flows were developed in late 60s [4], [6]-[9][6], and are widely used in nuclear reactor safety and thermal-hydraulics analysis, as well as for other multiphase-flow applications in oil and aerospace industries, geoscience, advanced weaponry, etc..
Consider two-phase flow equations averaged over control volume [6], [10], [11] shown in the

Figure 1. Each phase of the fluid is characterized by its own void fraction α_k, density ρ_k, temperature T_k (energy e_k), and velocity u_k, where index k takes values g for the gas/vapor phase and l for liquid phase. For non-homogeneous ($u_g \neq u_l$) and non-equilibrium ($T_g \neq T_l$) flow conditions a closed system of equations can be obtained assuming equal local pressure values for the two phases ($p_g = p_l = p$) that the source terms on the right-hand sides of the balance equations are exclusive algebraic functions of state and flow parameters [6]-[8], [10], [11]. The corresponding model often referred as “Wallis model” [7] has the following form

$$
\begin{aligned}
&\left(A\alpha_g\rho_g\right)_{,t} + \left(A\alpha_g\rho_g u_g\right)_{,x} = A\Gamma_g \\
&\left(A\alpha_g\rho_g u_g\right)_{,t} + \left(A\alpha_g\rho_g u_g^2\right)_{,x} + A\alpha_g p_{,x} = -A\alpha_g\rho_g z_{,x} - \tau_{gw}l_{wg} - \tau_{gi}l_i + A\Gamma_g u_{ig} \\
&\left(A\alpha_g\rho_g e_g\right)_{,t} + \left(A\alpha_g\rho_g e_g u_g\right)_{,x} = -Ap\alpha_{,t} - p\left(A\alpha u_g\right)_{,x} + \dot{q}_{gw}l_{wg} + \dot{q}_{gi}l_i + A\Gamma_g h_{ig} \\
&\left(A\alpha_l\rho_g\right)_{,t} + \left(A\alpha_l\rho_l u_l\right)_{,x} = -A\Gamma_g \\
&\left(A\alpha_l\rho_l u_l\right)_{,t} + \left(A\alpha_l\rho_l u_l^2\right)_{,x} + A\alpha_l p_{,x} = -A\alpha_l\rho_l z_{,x} - \tau_{lw}l_{wl} - \tau_{li}l_i - A\Gamma_g u_{il} \\
&\left(A\alpha_l e_l\rho_l\right)_{,t} + \left(A\alpha_l e_l\rho_l u_l\right)_{,x} = -Ap\left(1-\alpha\right)q_{,t} - p\left(A\left(1-\alpha\right)u_l\right)_{,x} + \dot{q}_{lw}l_{wl} + \dot{q}_{li}l_i - A\Gamma_g h_{il}
\end{aligned}
\tag{1}
$$

where the first three equations represent mass, momentum and energy conservation laws for gas/vapor and the last three equations represent the same laws for the liquid.

This set of equations can be closed by adding volume conservation equation

$$\alpha_g + \alpha_l = 1$$

and phasic state relationships

$$\rho_{g(l)} = \rho_{g(l)}\left(p, e_{g(l)}\right).$$

In addition, the constitutive equations for the interfacial mass flux Γ_g, friction and heat transfer at the wall and at the interface have to be given for various flow regimes. The functional form and parameters of these constitutive relations are the major unknowns of the two-phase flow problem in any specific environment as will be discussed in more details later.

The main applications of this model are real-time fault detection and isolation, parameter estimation, off-line analysis of the loading regimes, analysis and prioritization of the faults.

Two-phase flow model hierarchy in one dimension

Homogenous non-equilibrium model

To obtain the next model in the hierarchy we add together momenta equations for each phase and assuming equal phase velocities ($u_g = u_l = u$). Using volume conservation equation and introducing mixture density

$$\rho_m = \alpha_g \rho_g + \alpha_l \rho_l$$

we obtain the following mixture momentum equation

$$(A\rho_m u)_{,t} + (A\rho_m u^2)_{,x} + Ap_{,x} = -A\rho_m g z_{,x} - (\tau_w l_w)_{2\phi} \tag{2}$$

In deriving this equation, it was assumed that all the interfacial terms cancel each other giving rise to so-called jump condition

$$-\tau_{gi} l_i + A\Gamma_g u_{ig} + \tau_{li} l_i - A\Gamma_g u_{il} = 0.$$

In addition, it is usually assumed without derivation that the interface momentum transfer due to friction and due to mass transfer independently sum to zero, that is,

$$\tau_{gi} = \tau_{li} = \tau_i, \qquad u_{ig} = u_{il} = u_i.$$

The remaining equations can be left unchanged leading to the so-called five equations non-equilibrium homogeneous model (see e.g. [8]). The eigenvalue analysis [8] shows that this system has three distinct eigenvalues

$$\lambda_{1,2} = u \pm a, \qquad \lambda_3 = u \tag{3}$$

where the first two eigenvalues correspond to pressure waves with sound velocity

$$a = 1 \Big/ \sqrt{\frac{\alpha_g \rho_m}{a_g^2 \rho_g} + \frac{\alpha_l \rho_m}{a_l^2 \rho_l}}$$

and the last eigenvalue corresponds to void fraction wave. Two more eigenvalues are equal λ_3 and correspond to the entropy waves.

Homogeneous equilibrium model

The homogeneous equilibrium model can be obtained from the homogenous non-equilibrium model by assuming that two phases have the same temperature ($T_g = T_l = T$), velocities ($u_a = u_l = u$) and pressure ($p_a = p_l = p$) and by adding pairwise the equations of conservation for the mass and energy. The resulting set of equations has the form

$$\begin{aligned} &A\rho_{m,t} + (A\rho_m u)_{,x} = 0, \\ &A(\rho_m u)_{,t} + (A\rho_m u^2)_{,x} = -Ap_{,x} - (\tau_w l_w)_{2\phi} - \rho_m A g \sin\theta, \\ &A(\rho_m E)_{,t} + (A\rho_m u H)_{,x} = \dot{q}_w l_w, \end{aligned} \tag{4}$$

where $E = e + u^2/2$ is total energy, $H = E + p/\rho = h + u^2/2$ is the total enthalpy, θ is the angle of pipe axis with horizontal line, x is the length along the pipe, and the mixture properties are defined as follows

$$\rho = \alpha_g \rho_g + \alpha_l \rho_l, \quad e = \chi_g e_g + \chi_l e_l, \quad h = \chi_g h_g + \chi_l h_l \tag{5}$$

Here $\chi_{g(l)}$ is the corresponding mass fraction

$$\chi_{g(l)} = \frac{\alpha_{g(l)} \rho_{g(l)}}{\alpha_g \rho_g + \alpha_l \rho_l},$$

$e_{g(l)}$ and $h_{g(l)}$ are specific energy and enthalpy of gas (liquid).

In one-component two-phase media undergoing phase transitions (evaporation or condensation), the thermal equilibrium assumption results [8] in a strict coupling of pressure and temperature as given by the saturation condition $p = p_{sat}(T)$ as long as $0 \le \chi \le 1$. The structure of three

eigenvalues remains the same as in eq. (3). The primarily application of this model is a fast and reasonably accurate solution of the online optimization problem for thermal transients in non-isothermal compressible TPF networks.

Steady flow network

The last member of the considered hierarchy is the isothermal flow network model. It is assumed that the flow in the network has single phase and is incompressible and isothermal. Then the last equation in (4) is reduced to the hollowing statement that the fluid temperature in the network remains constant

$$T = const,$$

while the fist equation in (4) is reduced to the conservation of the total mass flow rate and volumetric flow rate at every cross-section and junction

$$A\rho_m u = const, \qquad \rho_m = const, \quad and \quad Au = const\,.$$

Finally, under assumption of steady flow the momentum equation in (4) integrated over section of the pipe with length L and diameter D becomes

$$\Delta p = -\left(\frac{f\,L}{D}\right)\frac{\rho u^2}{2} - \rho g \Delta h$$

This case is important for a few reasons. Firstly, it allows for the fast non-linear optimization during the search of the fault and control parameters. Secondly, this is the simplest possible approximation of the problem that yields useful results. In addition, this model is currently used by the KATE for the online FDI&M at KSC cryogenic testbed (CTB).

Intelligent autonomous loading control

An analysis of the current state of art modeling capabilities of the two-phase flow suggests that an efficient online autonomous control of the TPF can be accomplished in an optimal way using hierarchical approach to the control architecture. Within this architecture the most accurate of the available models (separated non-equilibrium TPF model) follows in real time the loading operation, detects the deviations from the nominal regime, generate a table of suspected faults and corresponding sensitivity matrixes, suggests the most probable mitigation strategies and the efficient method of using module capabilities for each specific fault. The most important use of this model is the detection of the critical faults at the earliest possible time.

The primarily use of the low level models at the bottom of the hierarchy (steady compressible and incompressible flow networks) is to provide estimations for the flow rates and pressure distribution along the system conditions in two limiting cases (compressible and incompressible flow). The corresponding estimations and mitigation strategies will be used as initial conditions by more accurate models.

The key application of the middle level models (e.g. homogeneous moving front model) is to provide the fastest possible solution of the online of optimization problem for the fault detection, evaluation, and mitigation within specified parameter' and control' subspaces. Essentially, for the cryogenic loading the model has to provide analysis of the temperature transients both for the wall and for the fluid and to optimize mitigation.

A very important application of the control module is off-line analysis and prioritization of each deviation of the loading operation from the nominal regime. The module will collect and analyze severity of the detected faults, continue to learn model parameters (physics of the two-phase flow) in autonomous regime. The module will also determine if further analysis of the loading regimes and/or system/components performance is required using ground facilities including high-fidelity supercomputer code and/or experimental testbeds.

In this report, we focus on the development of ine of the key components in this hierarchy -- fully separated non-equilibrium two-phase flow – which will be used for the real-time fault detection, isolation and parameter estimation. In the next section, we review briefly the current state of art in the methods of numerical solution of this model. Then we will describe briefly one of the solution methods, which is currently used for modeling cryogenic loading at KC CTB.

Solution methods (in a few words)

In our brief discussion below, we follow closely recent review [4]. The reader should consult this comprehensive text for further details. All current work-horse reactor thermal-hydraulics codes (RELAP5 [2], TRAC [12], TRACE [5], CATHARE [13] and RETRAN [14]) originate from Liles and Reed [15] extension of Harlow and Asden [16], [17] all-speed *"Implicit Continuous-Fluid Eulerian (ICE)"* algorithm. The whole family of these algorithms is often referred to as *"semi-implicit".*

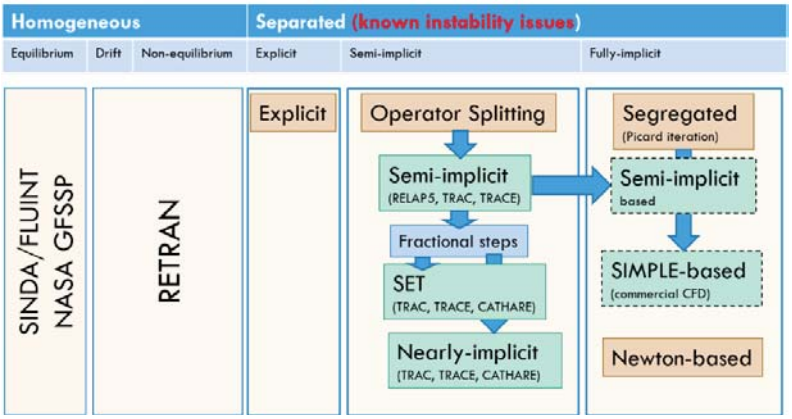

Figure 2 Hierarchy of the 1D TPF models and relation of the algorithms (see [4] for further details).

A couple of extensions of the "semi-implicit" algorithm were introduced, based on the fractional step method, to enhance stability of the method and to eliminate material CFL restrictions. These are the SETS algorithm (implemented in TRAC and TRACE, [5], [12]), and Nearly-Implicit algorithm (implemented in RELAP5-3D, [2]).

We refer to ICE-based algorithms as "weakly-compressible", even though the compressibility effects are fully accounted for in numerical discretization. Almost all reactor thermal-hydraulics codes belong to this class and utilize first-order finite-difference donor-cell/upwinding based schemes, implemented on structured staggered meshes.

The schematic presentation of the model hierarchy and relation between semi-implicit algorithms is show in the Figure 2. One of the current state of art solution of the problem of the real-time control of the two-phase flow in industry is based on the semi-implicit and nearly-implicit algorithms. Importantly, the corresponding models were verified and validated for multiple flow regimes and incorporate a few decades of extensive numerical experience.

In this work following the best industrial practice we focus on the development and implementation of the semi-implicit and nearly-implicit algorithms for the solution of the fully separated non-equilibrium two-phase model of the cryogenic flow.

Non-hyperbolicity of the Wallis model

Eigenvalues of the Wallis model

Before we proceed to the analysis of the solution we note that the set of equations of the Wallis model (1) is non-hyperbolic. Two eigenvalues of the system (1) are complex. To see this and to

understand the idea behind various techniques introduced in the literature to correct this problem we first write system (1) in the form

$$\cdot\frac{\partial F}{\partial t}+\frac{\partial U}{\partial x}=C\,. \tag{6}$$

where

$$F=\begin{bmatrix}\alpha\rho_g\\(1-\alpha)\rho_l\\\alpha\rho_g u_g\\(1-\alpha)\rho_l u_l\\\alpha\rho_g e_g\\(1-\alpha)\rho_l e_l\end{bmatrix};U=\begin{bmatrix}\alpha\rho_g u_g\\(1-\alpha)\rho_l u_l\\\alpha\rho_g u_g^2\\(1-\alpha)\rho_l u_l^2\\\alpha\rho_g e_g u_g\\(1-\alpha)\rho_l e_l u_l\end{bmatrix};C=\begin{bmatrix}0\\0\\0\\0\\-p\alpha_{,t}-p\left(\alpha u_g\right)_{,x}\\p\alpha_{,t}-p\left((1-\alpha)u_g\right)_{,x}\end{bmatrix};C_0=\begin{bmatrix}\Gamma_g\\-\Gamma_g\\\alpha\rho_g z_{,x}-\tau_{wg}l_{wg}-\tau_{gi}l_i+\Gamma_g u_{ig}\\(1-\alpha)\rho_g z_{,x}-\tau_{wg}l_{wg}-\tau_{gi}l_i+\Gamma_g u_{ig}\\\dot{q}_{gw}\frac{l_{wg}}{A}+\dot{q}_{gi}\frac{l_i}{A}+\Gamma_g h_{ig}\\\dot{q}_{wl}\frac{l_{wl}}{A}+\dot{q}_{li}\frac{l_i}{A}-\Gamma_g h_{il}\end{bmatrix}. \tag{7}$$

The next step is to linearize this system of equations and write it in the form

$$\left(\frac{\partial F}{\partial Q}-C_t\right)\frac{\partial Q}{\partial t}+\left(\frac{\partial U}{\partial Q}-C_x\right)\frac{\partial Q}{\partial x}=C_o, \tag{8}$$

where Q is the vector of primitive variables

$$Q=\begin{bmatrix}\alpha & p & u_g & u_l & e_g & e_l\end{bmatrix}^T. \tag{9}$$

The eigenvalues of the system (6) are found by solving generalized eigenvalue problem

$$f(\lambda)=\det(A\lambda-B)=0, \tag{10}$$

where $A=\partial F/\partial Q-C_t$ and $B=\partial U/\partial Q-C_x$. A Maple code that calculates required Jacobians and eigenvalues can be found in Appendix D. The Wallis model has the following set of distinct eigenvalues

$$\lambda=\left\{u,v,\frac{\alpha\rho_l u_l+(1-\alpha)\rho_g u_g}{\alpha\rho_l+(1-\alpha)\rho_g}\pm i\sqrt{\alpha(1-\alpha)}\frac{\left(u_g-u_l\right)}{\alpha\rho_l+(1-\alpha)\rho_g}\right\}. \tag{11}$$

It is clear that a pair of eigenvalues is complex at any parameters of the model. According to Staedke [8]: This indicates that the assumption of exclusively algebraic terms for the interfacial drag results in an incomplete formulation for the interfacial momentum coupling. A variety of corrective terms is suggested in the literature to mitigate this problem. Below we consider one of such terms, which used in the present model.

Virtual mass term

There is a common agreement that corrective terms that contain derivatives of the velocity or/and pressure are required to complete the formulation of the interfacial momentum transfer. However, at present there is no way to deduce these terms completely from the first principles [8]. A common correction (which is also used in RELAP5 code) is to add so-called the “virtual mass” term in the phasic momentum equations accounting for the effect of the local mass displacement in the case of a relative acceleration between the two phases. This term has the form (cf Ref [22])

$$M_V=C\alpha(1-\alpha)\rho_m\left[\frac{\partial\left(u_g-u_l\right)}{\partial t}+u_l\frac{\partial u_g}{\partial x}-u_g\frac{\partial u_l}{\partial x}\right]. \tag{12}$$

With an addition of this term the non-hyperbolicity of the Wallis system can be alleviated and the eigenvalues of the model take the form

$$\lambda = \left\{ u, v, \frac{2\left(\alpha\rho_l u_l + (1-\alpha)\rho_g u_g\right) - \rho_m C\left((1-\alpha)v - u\right)}{2\left(\alpha\rho_l + (1-\alpha)\rho_g - \rho_m C(1-\alpha)\right)} \pm \frac{\sqrt{\rho_m^2 C^2\left(u-(1-\alpha)v\right)^2 - 4\alpha(1-\alpha)\rho(u-v)\left(\rho_m C u v + r(u-v)\right)}}{2\left(\alpha\rho_l + (1-\alpha)\rho_g - \rho_m C(1-\alpha)\right)} \right\} \quad (13)$$

We note that non-hyperbolicity does not necessarily results in the instability, see e.g. [10] for a counter-example. However, numerical experience suggests that the introduction of the corrective terms is essential for obtaining stable solution.

A set of equations as it was coded

In this section, we briefly outline the details of the semi-implicit algorithm as it was coded for the chilldown applications. Further details can be found e.g. in [2], [4], [10]. A detailed outline of a similar nearly-implicit algorithm is provided in “Part IV: Code Structure” of this report.
As a first step in deriving this algorithm, we obtain a numerically convenient set of equations for the two-phase flow starting from the full set in the form

$$\left(A\alpha_g\rho_g\right)_{,t} + \left(A\alpha_g\rho_g u_g\right)_{,x} = A\Gamma_g,$$

$$\begin{aligned}
&\left(A\alpha_a\rho_a u_a\right)_{,t} + \left(A\alpha_a\rho_a u_a^2\right)_{,x} + A\alpha_a p_{,x} = -A\alpha_a\rho_a z_{,x} - \tau_{aw} l_{wa} - \tau_{ai} l_i + A\Gamma_a u_{ia},\\
&\left(A\alpha_g\rho_g E_g\right)_{,t} + \left(A\alpha_g\rho_g E_g u_g\right)_{,x} = A\alpha\, p_{,t} - \left(pA\alpha u_g\right)_{,x} + \dot{q}_{gw} l_{wg} + \dot{q}_{gi} l_i + A\Gamma_g h_{ig},
\end{aligned} \quad (14)$$

$$\begin{aligned}
&\left(A\beta\rho_a\right)_{,t} + \left(A\beta\rho_l u_l\right)_{,x} = -A\Gamma_a,\\
&\left(A\beta\rho_l u_l\right)_{,t} + \left(A\beta\rho_l u_l^2\right)_{,x} + A\beta p_{,x} = -A\beta\rho_l z_{,x} - \tau_{lw} l_{wl} - \tau_{li} l_i - A\Gamma_a u_{il},\\
&\left(A\beta E_l\rho_l\right)_{,t} + \left(A\beta E_l\rho_l u_l\right)_{,x} = A\beta p_{,t} - \left(pA\beta u_l\right)_{,x} + \dot{q}_{lw} l_{wl} + \dot{q}_{li} l_i - A\Gamma_a h_{il}.
\end{aligned}$$

Note that these equations are written for total the energies $E_{g,l} = e_{g,l} + u_{g,l}^2/2$ $E_{a,l} = e_{a,l} + \frac{u_{g,l}}{2}$. We now consider each pair of equations in turn

Continuity equations

The continuity equations have the form

$$\begin{aligned}
A\left(\alpha\rho_g\right)_{,t} + \left(A\alpha\rho_g u_g\right)_{,x} &= A\Gamma_g,\\
A\left(\beta\rho_l\right)_{,t} + \left(A\beta\rho_l u_l\right)_{,x} &= -A\Gamma_g.
\end{aligned} \quad (15)$$

It is convenient to write (15) in the form of the sum and difference of the mass conservation equations

$$\begin{aligned}
A\left(\alpha\rho_g + \beta\rho_l\right)_{,t} + \left(A\left(\alpha\rho_g u_g + \beta\rho_l u_l\right)\right)_{,x} &= 0,\\
A\left(\alpha\rho_g - \beta\rho_l\right)_{,t} + \left(A\left(\alpha\rho_g u_g - \beta\rho_l u_l\right)\right)_{,x} &= 2A\Gamma_g.
\end{aligned} \quad (16)$$

In the derivation of these equations the following jump conditions were used

$$\Gamma_l = -\Gamma_g,$$

corresponding to the conservation of total mass.

Momenta equations

The next step is to write sum and difference of the momenta equations in a non-conservative form. Expanding momenta equations and subtracting density equations multiplied by corresponding velocities we have (in this section β is used inter-changeably with (1-α))

$$\begin{aligned} A\alpha\rho_g u_{g,x} + \frac{1}{2}A\alpha\rho_g\left(u_g^2\right)_{,x} + A\alpha\, p_{,x} &= -A\alpha\rho_g z_{,x} - \left(\frac{f_g l_{wg}}{4}\right)\frac{\rho_g u_g\left|u_g\right|}{2} - c_i u_R\left|u_R\right| + A\Gamma_g\left(u_{gi}-u_g\right) + \tilde{M}_V, \\ A\beta\rho_l u_{l,x} + \frac{1}{2}A\beta\rho_l\left(u_l^2\right)_{,x} + A\beta\, p_{,x} &= -A\beta\rho_l z_{,x} - \left(\frac{f_l l_{wl}}{4}\right)\frac{\rho_l u_l\left|u_l\right|}{2} - c_i u_R\left|u_R\right| - A\Gamma_g\left(u_{li}-u_l\right) - \tilde{M}_V, \end{aligned} \tag{17}$$

where we have introduced velocity difference $u_R = u_a - u_l\, u_R = u_g - u_l$ term and virtual mass term $\tilde{M}_V = C\alpha\beta\rho_m\left(u_a - u_l\right)$. $\tilde{M}_V = \alpha\beta\rho_m(u_g - u_l)_{,t}$. The latter is a simplified version of the full expression (12).

Next, we notice that it is convenient to cancel cross-sectional area A *A* whenever possible and introduce (see RELAP5-3D [2] for further details) the following definitions of the frictional losses terms

$$\begin{aligned} A\alpha\rho_g u_g F_{wg} &= \left(\frac{f_g l_{wg}}{4}\right)\frac{\rho_g u_g\left|u_g\right|}{2}, \qquad A(1-\alpha)\rho_l u_l F_{wl} = \left(\frac{f_l l_{wl}}{4}\right)\frac{\rho_l u_l\left|u_l\right|}{2}, \\ A\alpha\rho_g F_{ig}\left(u_g - u_l\right) &= c_{gi} u_R\left|u_R\right|, \qquad A(1-\alpha)\rho_l u_l F_{il}\left(u_l - u_g\right) = -c_{li} u_R\left|u_R\right|. \end{aligned} \tag{18}$$

Using these definitions and adding and subtracting two momenta equations we obtain

$$\begin{aligned} &\alpha\rho_g u_{g,t} + \beta\rho_l u_{l,t} + \frac{\alpha\rho_g}{2}\left(u_g^2\right)_{,x} + \frac{\beta\rho_l}{2}\left(u_l^2\right)_{,x} + p_{,x} = -\rho_m z_{,x} - \alpha\rho_g u_g F_{wg} - \beta\rho_l u_l F_{wl} - \Gamma_g\left(u_g - u_l\right), \\ &u_{g,t} - u_{l,t} + \frac{\left(u_g^2\right)_{,x}}{2} + \frac{\left(u_l^2\right)_{,x}}{2} + \left(\frac{1}{\rho_g} + \frac{1}{\rho_l}\right)p_{,x} = u_l F_{wl} - u_g F_{wg} + \\ &\qquad \frac{\Gamma_g}{\alpha\rho_g}\left(u_i - u_g\right) + \frac{\Gamma_g}{\beta\rho_l}\left(u_i - u_l\right) + \rho_m F_i\left(u_g - u_l\right) + \frac{\rho_m}{\alpha\rho_g\beta\rho_l}\tilde{M}_V \end{aligned} \tag{19}$$

The following jump conditions were used in derivation of Eqs. (19)

$$-A\alpha\rho_g F_{ig}\left(u_g - u_l\right) + A\Gamma_g\left(u_{gi} - u_g\right) + \tilde{M}_V - A\beta\rho_l u_l F_{il}\left(u_l - u_g\right) - A\Gamma_g\left(u_{li} - u_l\right) - \tilde{M}_V = 0$$

corresponding to the fact that all internal interactions must cancel each other according to the 1st Newton's law. It is further assumed without proof that interfacial drags related to the surface friction, mass transfer, and virtual mass cancel each other independently resulting in the following simplified set of the jump conditions

$$\alpha\rho_g u_g F_{ig} = \beta\rho_l u_l F_{il}; \qquad u_{gi} = u_{li} = u_i.$$

Energy equations

To simplify and speed up calculations it is convenient [2] to rewrite equations of energy in the form that contains only internal energy of liquid and gas. To derive the equations for internal energy we first derive equations for the kinetic energy and subtract the kinetic energy equations from the equations for the total energy. The equations for kinetic energy are obtained by multiplying conservative and non-conservative forms of the momenta equations by $1/2\, u\, \frac{1}{2} u$ and adding them together. For example for the kinetic energy of the gas/vapor we obtain

$$\left(A\alpha\rho_g\frac{u_g^2}{2}\right)_{,t}+\left(A\alpha\rho_g u_g\frac{u_g^2}{2}\right)_{,x}=-A\Gamma_g\frac{u_g^2}{2}+\left(-\alpha Ap_{,x}-\tau_{wg}l_{wg}-\tau_i l_i+A\Gamma_g u_i\right)u_g$$

Subtracting this equation from the equation for the total energy and neglecting the interfacial transfer of kinetic energy and frictional heating we have

$$\begin{aligned}&\left(A\alpha\rho_g e_g\right)_{,t}+\left(A\alpha\rho_g e_g u_g\right)_{,x}=-Ap\alpha_{,t}-p\left(A\alpha u_g\right)_{,x}+\dot{q}_{gw}l_{wg}+\dot{q}_{gi}l_i+A\Gamma_g h_{ig},\\&\left(A\beta\rho_l e_l\right)_{,t}+\left(A\beta\rho_l e_l u_l\right)_{,x}=-Ap\beta_{,t}-p\left(A\beta u_l\right)_{,x}+\dot{q}_{lw}l_{wl}+\dot{q}_{li}l_i-A\Gamma_g h_{il}.\end{aligned}\qquad(20)$$

The jump conditions for the energy conservation are

$$\dot{q}_{gi}l_i+A\Gamma_g h_{ig}+\dot{q}_{li}l_i-A\Gamma_g h_{il}=0.$$

Expanded form of the density and energy equations

In addition to the set of equations (16), (19), and (20) the first step of both semi-implicit and nearly-implicit algorithms requires solution of the expanded form of the coupled sum and difference density and energy equations. This set of equations is obtained by expanding time derivatives of equations (16) and (20) as follows

$$\begin{aligned}&\alpha\rho_{g,t}+\beta\rho_{l,t}+\alpha_{,t}\left(\rho_g-\rho_l\right)+\frac{1}{A}\left(A\left(\alpha\rho_g u_g+\beta\rho_l u_l\right)\right)_{,x}=0,\\&\alpha\rho_{g,t}-\beta\rho_{l,t}+\alpha_{,t}\left(\rho_g+\rho_l\right)+\frac{1}{A}\left(A\left(\alpha\rho_g u_g-\beta\rho_l u_l\right)\right)_{,x}=2\Gamma_g,\\&\left(\rho_g e_g+p\right)\alpha_{,t}+\alpha\rho_g e_{g,t}+\alpha e_g\rho_{g,t}+\frac{1}{A}\left[\left(A\alpha\rho_g e_g u_g\right)_{,x}+p\left(A\alpha u_g\right)_{,x}\right]=\dot{q}_{gw}\frac{l_{wg}}{A}+\dot{q}_{gi}\frac{l_i}{A}+\Gamma_g h_{ig},\\&-\left(\rho_l e_l+p\right)\alpha_{,t}+\beta\rho_l e_{l,t}+\beta e_l\rho_{l,t}+\frac{1}{A}\left[\left(A\beta\rho_l e_l u_l\right)_{,x}+p\left(A\beta u_l\right)_{,x}\right]=\dot{q}_{lw}\frac{l_{wl}}{A}+\dot{q}_{li}\frac{l_i}{A}-\Gamma_g h_{il}.\end{aligned}\qquad(21)$$

Equation for the wall temperature

In the present research the following simplifications are adopted for the analysis of the wall temperature. We neglect axial conduction in the wall as compared to the heat transfer through the wall on the ambient side and internal flow sides. We also assume that the thickness of the pipe wall d_w is small enough to estimate pipe wall volume as $V_w=\pi Dd_w$, where D $D$is the pipe diameter. The resulting equation for the wall temperature becomes

$$\rho_w c_w d_w\frac{\partial T_w}{\partial t}=h_{gw}\left(T_g-T_w\right)+h_{lw}\left(T_l-T_w\right)+h_{amb}\left(T_{amb}-T_w\right)\qquad(22)$$

A preview of the algorithm

The efficiency of the solution in semi-implicit and nearly-implicit algorithms of RELAP5 is archived by splitting the operators for conduction in the wall (22) and two-phase flow (16), (19), (20) into practically independent problems (see also [4], [23]). In turn, the solution of the latter problem is constructed in a two-stage “predictor-corrector” fashion [4].
In the first step the expanded mass and energy conservation equations (21) are solved together with momenta equations (19). At the second step of the algorithm the unexpanded conservation equations (15) and (20) are solved to enforce conservation of the mass and energy.

At each step of the algorithm the discretized equations are carefully linearized with respect values at the new time step to break CFL limitations related to the pressure waves (semi-implicit) and material waves (nearby-implicit).

An outline of the semi-implicit algorithm

In this section we describe the solution of the Wallis model using semi-implicit algorithm. First, we introduce a set of control volumes, and staggered grid. Next, we describe approximation of the scalar variables at the interfaces of control volumes and of the velocities at the centers of the control volumes required for the integration. Finally, we describe two steps of the solution. The key goal of this method is to lift limitations on the CFL number related to the speed of sound. Note, however, that the material CFL limitation

$$\Delta t < \frac{\Delta x}{\max\left(\left|u_{g(l)}\right|\right)} \tag{23}$$

still holds in the semi implicit scheme.
The advantage of the approach proposed in [2] and adopted for the current chilldown model is its efficiency and linear scaling of the algorithm with the number of control volumes. These advantages are particularly important for the applications to the real time control of cryogenic systems.

Grid

To discretize model equations we introduce N control volumes (CV). The scalar values including pressure, temperatures, densities, and energies are stored at the centers of the CVs indexed $L = 1, \ldots, N$ as shown in the Fig. 3

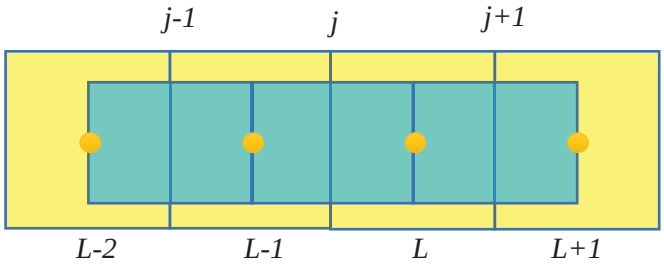

Figure 3 Control volumes $\left(L = 1, \ldots, N\right)$ *and staggered grid* $\left(j = 1, \ldots, N+1\right)$ *of the model.*

Upwind values

The values of the velocities and mass, energy, and would fluxes are calculated at the interfaces centered at staggered grid with indexes $j = 1, \ldots, N+1$. The values of scalar variables are calculated using upwind scheme as follows

$$\widehat{\varphi}_j = \frac{1}{2}\left(\varphi_{L-1}+\varphi_L\right)+\frac{1}{2}\left(s_{u,j}+z_{u,j}\cdot s_{p,j}\right)\left(\varphi_{L-1}-\varphi_L\right)=\left(1+\left(s_{u,j}+z_{u,j}\cdot s_{p,j}\right)\right)\frac{\varphi_{L-1}}{2}+\left(1-\left(s_{u,j}+z_{u,j}\cdot s_{p,j}\right)\right)\frac{\varphi_L}{2},$$

$$\widehat{\psi}_j = \frac{1}{2}\left(\psi_{L-1}+\psi_L\right)+\frac{1}{2}\left(s_{u,j}+z_{u,j}\cdot s_{p,j}\right)\left(\psi_{L-1}-\psi_L\right)+z_{u,j}\cdot z_{p,j}\left(-\frac{1}{2}\left(\psi_{L-1}+\psi_L\right)+\frac{\left(\psi_{L-1}\rho_{g(l),L-1}+\psi_L\rho_{g(l),L}\right)}{\rho_{g(l),L-1}+\rho_{g(l),L}}\right)= \quad (24)$$

$$\left(1+\left(s_{u,j}+z_{u,j}\cdot s_{p,j}\right)-z_{u,j}\cdot z_{p,j}+\frac{2\rho_{g(l),L-1}}{\rho_{g(l),L-1}+\rho_{g(l),L}}\right)\frac{\psi_{L-1}}{2}+\left(1-\left(s_{u,j}+z_{u,j}\cdot s_{p,j}\right)-z_{u,j}\cdot z_{p,j}+\frac{2\rho_{g(l),L-1}}{\rho_{g(l),L-1}+\rho_{g(l),L}}\right)\frac{\psi_L}{2}.$$

Here φ corresponds to the densities $\rho_{g(l)}$, while ψ corresponds to the void fractions $a_{g(l)}$ or energies $e_{g(l)}$. The values *s* and *zz* are defined as follows

$$s_{u,j} = sign(u_j);\quad s_{p,j} = sign(p_j);\quad z_{u,j} = \left\{1\ if\ u_j=0,\ 0\ otherwise\right\};\quad z_{p,j} = \left\{1\ if\ p_j=0,\ 0\ otherwise\right\}.$$

An additional smoothing is introduced for small velocities for scalar values centered at the junctions

$$\varphi_j = \zeta_j\varphi_{L-1}+\left(1-\zeta_j\right)\varphi_L;\quad \zeta_j = \begin{cases} 1 & when \quad j_m > j_{\lim} \\ \xi^2\left(3-2\xi\right) & when \quad -j_{\lim} < j_m < j_{\lim} \\ 0 & when \quad j_m < -j_{\lim} \end{cases}$$

The following notations are used in this smoother $\xi = \left(j_m + j_{lim}\right)/\left(2\,j_{lim}\right)$, $j_m = \hat{\alpha}_{g,j}u_{g,j}+\hat{\alpha}_{l,j}u_{l,j}$, and $j_{lim} = 0.46\ m/s$.

Face centered interpolated values

In the discretization of the model equations the upwind values introduced above are used in the flux terms. For the scalar values entering time derivatives and source terms in momenta equations are interpolated between values in neighboring control volumes as follows

$$\overline{\psi}_{g(l),j} = \frac{1}{2}\left(\overline{\psi}_{g(l),L-1}l_{L-1}+\overline{\psi}_{g(l),L}l_L\right)\Big/\left(l_{L-1}+l_L\right). \quad (25)$$

Volume-centered velocities

The volume-centered velocities are required to calculate the kinetic energy in the momenta equations. For a simple pipe flow volume-centered velocities can be found using mass conservation law

$$A_L\alpha_{g(l),L}\rho_{g(l),L}u_{g(l),L} = \frac{1}{2}\left[A_j\widehat{\alpha}_{g(l),j}\widehat{\rho}_{g(l),j}u_{g(l),j}+A_{j+1}\widehat{\alpha}_{g(l),j+1}\widehat{\rho}_{g(l),j+1}u_{g(l),j+1}\right]$$

resulting in the following expression

$$u_{g(l),L} = \frac{A_j\widehat{\alpha}_{g(l),j}\widehat{\rho}_{g(l),j}}{2A_L\alpha_{g(l),L}\rho_{g(l),L}}u_{g(l),j}+\frac{A_{j+1}\widehat{\alpha}_{g(l),j+1}\widehat{\rho}_{g(l),j+1}}{2A_L\alpha_{g(l),L}\rho_{g(l),L}}u_{g(l),j+1} \quad (26)$$

Artificial velocity diffusion

In the semi-implicit version of the algorithm [2] an additional stabilizing factor is introduced in the form of the artificial velocity diffusion

$$\frac{1}{2}\alpha_g\rho_g\Delta x\frac{\partial^2\left(A\left|u_g\right|u_g\right)}{\partial x^2}.$$

The integration of this term on a staggered grid results in the hollowing expression

$$\frac{1}{2}\left(\alpha\rho\Delta x\right)_{g,j}^{n}\left[\partial_x\left(A\left|u_g\right|u_g\right)\Big|_{L}^{n}-\partial_x\left(A\left|u_g\right|u_g\right)\Big|_{L-1}^{n}\right],$$

which can be further simplified by writing a finite difference approximation

$$F_{VG(L),j}^{n}=\frac{1}{2}\left(\alpha\rho\right)_{g(l),j}^{n}\left[\left|u_{g(l),L}\right|\left(\left|\left(Au_{g(l)}\right)_{j+1}^{n}\right|-\left(Au_{g(l)}\right)\Big|_{j}^{n}\right)-\left|u_{g(l),L-1}\right|\left(\left|\left(Au_{g(l)}\right)_{j}^{n}\right|-\left(Au_{g(l)}\right)\Big|_{j-1}^{n}\right)\right] \quad (27)$$

First step of the semi-implicit algorithm

In this section we describe the first step of the algorithm as it is coded. It involves the following sub-steps

- Solve momenta equations for new time velocities in terms of new time pressure
- Solve expanded equations for the new pressure
- Find new velocities using new pressure
- Solve expanded equations for the provisional values of void fractions and energies
- Find provisional values of the temperatures and interphase mass fluxes

The main difference with the original version [2] is the use of the incremental values for the new scalar variables and velocities, which eliminates the necessity to use special techniques for the solution of resulting matrix equations. The notions new and old values refer to the time steps $n+1$ and n correspondingly.

Discretizing and solving momenta equations

Integrating the momenta equations (19) over volume of staggered grid

$$\begin{aligned}&\left[\left(\overline{\alpha\rho}\right)_{g,j}^{n}du_{g,j}^{n+1}+\left(\overline{\alpha\rho}\right)_{l,j}^{n}du_{l,j}^{n+1}\right]\Delta x_j+\left(\frac{1}{2}\left(\widehat{\alpha\rho}\right)_{g,j}^{n}\left(\left(u_g^2\right)_{L}^{n}-\left(u_g^2\right)_{L-1}^{n}-F_{VG\,j}^{\;n}\right)+\frac{1}{2}\left(\widehat{\alpha\rho}\right)_{l,j}^{n}\left(\left(u_l^2\right)_{L}^{n}-\left(u_l^2\right)_{L-1}^{n}-F_{VL\,j}^{\;n}\right)\right)\Delta t-\\&\quad=-\left(dp_L^{n+1}-dp_{L-1}^{n+1}\right)\Delta t-\left(p_L^n-p_{L-1}^n\right)\Delta t+\\&\left[-\rho_{m,j}^{n}g\Delta z_j-\left(\overline{\alpha\rho}\right)_{g,j}^{n}F_{wg,j}^{n}\left(du_{g,j}^{n+1}+u_{g,j}^{n}\right)-\left(\overline{\alpha\rho}\right)_{l,j}^{n}F_{wl,j}^{n}\left(du_{l,j}^{n+1}+u_{l,j}^{n}\right)-\Gamma_{g,j}^{n}\left(du_{g,j}^{n+1}+u_{g,j}^{n}-du_{l,j}^{n+1}-u_{l,j}^{n}\right)\right]\Delta t\Delta x_j\end{aligned} \quad (28)$$

$$\begin{aligned}&\left(1+\frac{c\left(\rho_m\right)^2}{\overline{\rho}_g\overline{\rho}_l}\right)_{j}^{n}\left[du_{g,j}^{n+1}-du_{l,j}^{n+1}\right]\Delta x_j+\frac{\Delta t}{2}\left(\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{g,j}^{n}\left(\left(u_g^2\right)_{L}^{n}-\left(u_g^2\right)_{L-1}^{n}-F_{VG,j}^{n}\right)-\right.\\&\left.\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{l,j}^{n}\left(\left(u_l^2\right)_{L}^{n}-\left(u_l^2\right)_{L-1}^{n}-F_{VL,j}^{n}\right)\right)=-\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)\left(p_L^{n+1}-p_{L-1}^{n+1}\right)\Delta t-\left\{F_{wg,j}^{n}\left(du_{g,j}^{n+1}+u_{g,j}^{n}\right)-F_{wl,j}^{n}\left(du_{l,j}^{n+1}+u_{l,j}^{n}\right)-\right.\\&\Gamma_{g,j}^{n}\frac{\rho_{m,j}^{n}du_{l,j}^{n+1}-\left(\overline{\alpha\rho}\right)_{g,j}^{n}du_{l,j}^{n+1}-\left(\overline{\alpha\rho}\right)_{l,j}^{n}du_{g,j}^{n+1}}{\left(\overline{\alpha\rho}\right)_{g,j}^{n}\left(\overline{\alpha\rho}\right)_{l,j}^{n}}-\Gamma_{g,j}^{n}\frac{\rho_{m,j}^{n}u_{l,j}^{n}-\left(\overline{\alpha\rho}\right)_{g,j}^{n}u_{l,j}^{n}-\left(\overline{\alpha\rho}\right)_{l,j}^{n}u_{g,j}^{n}}{\left(\overline{\alpha\rho}\right)_{g,j}^{n}\left(\overline{\alpha\rho}\right)_{l,j}^{n}}\\&\left.\left(F_I\rho\right)_j^{n}\left(du_{g,j}^{n+1}+u_{g,j}^{n}-du_{l,j}^{n+1}-u_{l,j}^{n}\right)\right\}\Delta t\Delta x_j\end{aligned} \quad (29)$$

We note that of the new time step these equations involve only $du_{g,j}^{n+1}$ $du_{g,j}^{n+1}$, $du_{l,j}^{n+1}$ $du_{l,j}^{n+1}$, and $p_{L(L-1)}^{n+1}$ $p_{L(L-1)}^{n+1}$. Therefore, at j-th interface the two equations can be presented in the form

$$A_u^n\begin{bmatrix}du_{g,j}^{n+1}\\du_{l,j}^{n+1}\end{bmatrix}=a_u^n p_L^{n+1}+b_u^n p_{L-1}^{n+1}+c_u^n.$$

This equation is solved to express new velocities in terms of new pressure

$$du_{g,j}^{n+1} = \left(A_u^n\right)_1^{-1}\left(a_u^n p_L^{n+1} + b_u^n p_{L-1}^{n+1} + c_u^n\right), \qquad du_{l,j}^{n+1} = \left(A_u^n\right)_2^{-1}\left(a_u^n p_L^{n+1} + b_u^n p_{L-1}^{n+1} + c_u^n\right). \tag{30}$$

Note, that in the equations above matrix A_u^n and vectors a_u^n, b_u^n, and c_u^n are functions of variables at the previous time step. The solution (30) can be substituted into the set of expanded equations to find new pressures.

Solving expanded equations for new pressure

Integrating expanded equations (21) over the length of the control volume we obtain for the density

$$\begin{aligned}
&\alpha_{g,L}^n d\rho_{g,L}^{n+1} + \beta_{l,L}^n d\rho_{l,L}^{n+1} + d\alpha_{g,L}^{n+1}\left(\rho_{g,L}^n - \rho_{l,L}^n\right) + \frac{\Delta t}{V}\left(\left(\overline{\alpha\rho}A\right)_{g,j+1}^n u_{g,j+1}^{n+1} - \left(\overline{\alpha\rho}A\right)_{g,j}^n u_{g,j}^{n+1}\right) + \\
&\quad \frac{\Delta t}{V}\left(\left(\overline{\alpha\rho}A\right)_{l,j+1}^n u_{l,j+1}^{n+1} - \left(\overline{\alpha\rho}A\right)_{l,j}^n u_{l,j}^{n+1}\right) = 0, \\
&\alpha_{g,L}^n d\rho_{g,L}^{n+1} - \beta_{l,L}^n d\rho_{l,L}^{n+1} + d\alpha_{g,L}^{n+1}\left(\rho_{g,L}^n + \rho_{l,L}^n\right) + \frac{\Delta t}{V}\left(\left(\overline{\alpha\rho}A\right)_{g,j+1}^n u_{g,j+1}^{n+1} - \left(\overline{\alpha\rho}A\right)_{g,j}^n u_{g,j}^{n+1}\right) - \\
&\quad - \frac{\Delta t}{V}\left(\left(\overline{\alpha\rho}A\right)_{l,j+1}^n u_{l,j+1}^{n+1} - \left(\overline{\alpha\rho}A\right)_{l,j}^n u_{l,j}^{n+1}\right) = 2\Gamma_{g,L}^n,
\end{aligned} \tag{31}$$

and for the energy

$$\begin{aligned}
&\left(\rho_{g,L}^n e_{g,L}^n + p_L^n\right) d\alpha_{g,L}^{n+1} + \left(\alpha\rho\right)_{g,L}^n de_{g,L}^{n+1} + \left(\alpha e\right)_{g,L}^n d\rho_{g,L}^{n+1} + \frac{\Delta t}{V}\left[\left(A\overline{\alpha}\left(\overline{\rho e} + p\right)\right)_{g,j+1}^n u_{g,j+1}^{n+1} + \left(A\overline{\alpha}\left(\overline{\rho e} + p\right)\right)_{g,j}^n u_{g,j}^{n+1}\right] = \\
&\quad \left[\tilde{h}_{gw,L}^n\left(T_{w,L}^n - \tilde{T}_{g,L}^{n+1}\right) S_{wg} + \tilde{h}_{ig,L}^n\left(\tilde{T}_{l,L}^{s,n} - \tilde{T}_{g,L}^{n+1}\right) S_{ig} + \tilde{\Gamma}_{g,L}^n h_{ig,L}^n\right]\frac{\Delta t}{V}, \\
&-\left(\rho_{l,L}^n e_{l,L}^n + p_L^n\right) d\alpha_{g,L}^{n+1} + \left(\alpha\rho\right)_{l,L}^n de_{l,L}^{n+1} + \left(\alpha e\right)_{l,L}^n d\rho_{l,L}^{n+1} + \frac{\Delta t}{V}\left[\left(A\overline{\alpha}\left(\overline{\rho e} + p\right)\right)_{l,j+1}^n u_{l,j+1}^{n+1} + \left(A\overline{\alpha}\left(\overline{\rho e} + p\right)\right)_{l,j}^n u_{l,j}^{n+1}\right] = \\
&\quad \left[\tilde{h}_{lw,L}^n\left(T_{w,L}^n - \tilde{T}_{l,L}^{n+1}\right) S_{wl} + \tilde{h}_{igl,L}^n\left(\tilde{T}_{l,L}^{s,n+1} - \tilde{T}_{l,L}^{n+1}\right) S_{il} - \tilde{\Gamma}_{g,L}^n h_{il,L}^n\right]\frac{\Delta t}{V}.
\end{aligned} \tag{32}$$

To solve these equations we write velocities in the form $u_{g(l),j}^{n+1} = u_{g(l),j}^n + du_{g(l),j}^{n+1}$ and express $du_{g(l),j}^{n+1}$ in terms of new pressure using equation (30). Next, we express densities and temperatures at the new time step in terms of energies and pressure using Taylor series expansions as follows

$$\begin{aligned}
&d\rho_{g,L}^{n+1} = \rho_{g,L}^{n+1} - \rho_{g,L}^n \approx \left(\frac{\partial\rho}{\partial p}\right)^n dp_L^{n+1} + \left(\frac{\partial\rho}{\partial e}\right)^n de_{g,L}^{n+1}; \qquad d\rho_{l,L}^{n+1} = \rho_{l,L}^{n+1} - \rho_{l,L}^n \approx \left(\frac{\partial\rho}{\partial p}\right)^n dp_L^{n+1} + \left(\frac{\partial\rho}{\partial e}\right)^n de_{l,L}^{n+1}; \\
&dT_{g,L}^{n+1} = \tilde{T}_{g,L}^{n+1} - T_{g,L}^n \approx \left(\frac{\partial T}{\partial p}\right)^n dp_L^{n+1} + \left(\frac{\partial T}{\partial e}\right)^n de_{g,L}^{n+1}; \qquad dT_{l,L}^{n+1} = \tilde{T}_{l,L}^{n+1} - T_{l,L}^n \approx \left(\frac{\partial T}{\partial p}\right)^n dp_L^{n+1} + \left(\frac{\partial T}{\partial e}\right)^n de_{l,L}^{n+1}.
\end{aligned} \tag{33}$$

The derivatives in these expansions are taken from the NIST tables as will be discussed in more details in the Part III of this report that discusses correlations. After these substitutions the system of equations can be written as a matrix equation

$$A_x^n dx_{g,L}^{n+1} = a_x^n du_{g,j+1}^{n+1} + b_x^n du_{g,j}^{n+1} + c_x^n du_{l,j+1}^{n+1} + d_x^n du_{l,j}^{n+1} + e_x^n \tag{34}$$

for the following unknowns

$$dx_L^n = \left\{d\alpha_{g,L}^{n+1}, de_{g,L}^{n+1}, de_{l,L}^{n+1}, dp_L^{n+1}\right\}.$$

Below we provide the explicit form of matrixes and vectors in (34) under additional simplifying assumption that all the temperatures in the source terms are taken at the old time. The rationale behind this assumption is twofold. Firstly, in practical applications the time step for semi-implicit algorithm is of the order of 5 ms, which is much longer than the characteristic time of the temperature relaxation, since the temperature is the slowest variable of the model. This

approximation can be violated for very small void fractions for one of the phases. However, in the limit of small void fraction, the whole algorithm breaks down and limiters and smoothers are employed to stabilize calculations as will be discussed in the Section related to the phase appearance/disappearance. Secondly, the 2nd step of the algorithm is used to enforce mass and energy conservation and the temperature is corrected during this 2nd step. Ultimately, the final approximation will be established after validation of the algorithm and extensive numerical experimentation.

Under these assumptions, the matrix A_x has the following form

$$A_x = \begin{bmatrix} \alpha^n_{g,L}\left(\partial_e\rho\right)^n_{g,L} & -\left(1-\alpha^n_{g,L}\right)\left(\partial_e\rho\right)^n_{l,L} & \rho^n_{g,L}+\rho^n_{l,L} & \alpha^n_{g,L}\left(\partial_p\rho\right)^n_{g,L}-\left(1-\alpha^n_{g,L}\right)\left(\partial_p\rho\right)^n_{l,L} \\ \alpha^n_{g,L}\left(\left(e\partial_e\rho\right)^n_{g,L}+\rho^n_{g,L}\right) & 0 & \left(e\rho\right)^n_{g,L}+p^n_L & \left(e\alpha\partial_p\rho\right)^n_{g,L} \\ 0 & \left(1-\alpha^n_{g,L}\right)\left(\rho^n_{l,L}+\left(e\partial_e\rho\right)^n_{l,L}\right) & -\left(e\rho\right)^n_{l,L}-p^n_L & \left(\left(1-\alpha\right)e\partial_p\rho\right)^n_{l,L} \\ \alpha^n_{g,L}\left(\partial_e\rho\right)^n_{g,L} & \left(1-\alpha^n_{g,L}\right)\left(\partial_e\rho\right)^n_{l,L} & \rho^n_{g,L}-\rho^n_{l,L} & \alpha^n_{g,L}\left(\partial_p\rho\right)^n_{g,L}+\left(1-\alpha^n_{g,L}\right)\left(\partial_p\rho\right)^n_{l,L} \end{bmatrix}$$

The vectors a^n_x and b^n_x have the form

$$a^n_x = -\begin{bmatrix} \widehat{\rho}^n_{g,j+1} \\ \left(\left(\widehat{e}\widehat{\rho}\right)^n_{g,j+1}+p^n_L\right) \\ 0 \\ \widehat{\rho}^n_{g,j+1} \end{bmatrix}\cdot\widehat{\alpha}^n_{g,j+1}\frac{\Delta tA_{j+1}}{V_L}; \qquad b^n_x = \begin{bmatrix} \widehat{\rho}^n_{g,j} \\ \left(\left(\widehat{e}\widehat{\rho}\right)^n_{g,j}+p^n_L\right) \\ 0 \\ \widehat{\rho}^n_{g,j} \end{bmatrix}\cdot\widehat{\alpha}^n_{g,j}\frac{\Delta tA_j}{V_L};$$

and the vectors c^n_x and d^n_x have the form

$$c^n_x = \begin{bmatrix} \widehat{\rho}^n_{l,j+1} \\ 0 \\ -\left(\left(\widehat{e}\widehat{\rho}\right)^n_{l,j+1}+p^n_L\right) \\ -\widehat{\rho}^n_{l,j+1} \end{bmatrix}\cdot\widehat{\alpha}^n_{l,j+1}\frac{\Delta t\cdot A_{j+1}}{V_L}; \qquad d^n_x = \begin{bmatrix} -\widehat{\rho}^n_{l,j} \\ 0 \\ \left(\left(\widehat{e}\widehat{\rho}\right)^n_{l,j}+p^n_L\right) \\ \widehat{\rho}^n_{l,j} \end{bmatrix}\cdot\widehat{\alpha}^n_{l,j}\frac{\Delta t\cdot A_j}{V_L}.$$

Finally, the free vector in the equation (34) is written as follows

$$e^n_x = \begin{bmatrix} 2\Gamma^n_{g,L}V_L \\ \Gamma^n_{g,L}h^n_{wg}V_L-Q_{wg} \\ -\Gamma^n_{g,L}h^n_{wl}V_L-Q_{wl} \\ 0 \end{bmatrix}\cdot\frac{\Delta t}{V_L}+a^n_x u^n_{g,j+1}+b^n_x u^n_{g,j}+c^n_x u^n_{l,j+1}+d^n_x u^n_{l,j}$$

To solve equation (34) we notice that on substituting (30) the last row of this equation depends only on the new pressure increments dp^{n+1}_{L-1}, dp^{n+1}_L dp^{n+1}_L, and dp^{n+1}_{L+1} dp^{n+1}_{L+1}. Multiplying the last row of eq. (34) by inverted matrix A^n_x we obtain the matrix equation for the pressure in the form

$$dp^{n+1}_L = a^n_p dp^{n+1}_{L-1}+b^n_p dp^{n+1}_L+c^n_p dp^{n+1}_{L+1}+d^n_p\ . \tag{35}$$

This tridiagonal equation can be readily solved using standard algorithm (see Appendix C).

New velocities and provisional values of densities, void fractions, and energies

Once new pressure is known the new velocities are obtain by substituting result of solution of (35) into (30). Next, using new velocities and pressures we obtain provisional values for densities, void fractions, and energies by solving the remaining three equations of (34). Next, we update the upwind values of scalar variables using new velocities and pressures. At this stage we also update the heat transfer coefficients and interphase mass transfer rate using provisional values for the

temperatures and new velocities. This completes the first step of the algorithm that solves momenta equations and expanded (non-conservative) form of equations for densities and energies. Since, the densities and energies at this step were obtained using solution of the non-conservative form of the equations to corrects the errors a second step is required.
In the following section we describe the calculation of corrected the mass and energies by solving unexpanded conservative form of the density and energy equations (15) and (20).

Second step of the algorithm

To update densities and energies we integrate equations (15) over the control volume and obtain the following equations

$$\begin{aligned}
\left(\alpha\rho\right)_{g,L}^{n+1} &= \left(\alpha\rho\right)_{g,L}^{n} - \left(\left(\widehat{\alpha\rho}\right)_{g,\mathrm{j}+1}^{n} u_{g,\mathrm{j}+1}^{n+1} A_{j+1} - \left(\widehat{\alpha\rho}\right)_{g,\mathrm{j}}^{n} u_{g,\mathrm{j}}^{n+1} A_{j} \right) \frac{\Delta t}{V_L} + \Gamma_{g,L}^{n} \Delta t, \\
\left(\alpha\rho\right)_{l,L}^{n+1} &= \left(\alpha\rho\right)_{l,L}^{n} - \left(\left(\widehat{\alpha\rho}\right)_{l,\mathrm{j}+1}^{n} u_{l,\mathrm{j}+1}^{n+1} A_{j+1} - \left(\widehat{\alpha\rho}\right)_{l,\mathrm{j}}^{n} u_{l,\mathrm{j}}^{n+1} A_{j} \right) \frac{\Delta t}{V_L} - \Gamma_{g,L}^{n} \Delta t.
\end{aligned} \tag{36}$$

Integrating the energy equations (20) we obtain

$$\begin{aligned}
\left(\alpha\rho e\right)_{g,L}^{n+1} &= \left(\alpha\rho e\right)_{g,L}^{n} - \left[\widehat{\alpha}_{g,j+1}^{n} \left(\left(\widehat{e\rho}\right)_{g,\mathrm{j}+1}^{n} + p_L^{n} \right) u_{g,\mathrm{j}+1}^{n+1} A_{j+1} - \widehat{\alpha}_{g,j}^{n} \left(\left(\widehat{e\rho}\right)_{g,j}^{n} + p_L^{n} \right) u_{g,j}^{n+1} A_{j} \right] \frac{\Delta t}{V_L} - \\
& p_L^{n} d\tilde{\alpha}_{g,L}^{n} + \left(\tilde{h}_{wg,L}^{n} \left(\tilde{T}_{w,L}^{n} - \tilde{T}_{\mathrm{g},L}^{n} \right) S_{wg,L}^{n} + V_L \tilde{\Gamma}_{g,L}^{n} \tilde{h}_{g,L}^{n} \right) \frac{\Delta t}{V_L}, \\
\left(\alpha\rho e\right)_{l,L}^{n+1} &= \left(\alpha\rho e\right)_{l,L}^{n} - \left[\widehat{\alpha}_{l,j+1}^{n} \left(\left(\widehat{e\rho}\right)_{l,\mathrm{j}+1}^{n} + p_L^{n} \right) u_{l,\mathrm{j}+1}^{n+1} A_{j+1} - \widehat{\alpha}_{l,j}^{n} \left(\left(\widehat{e\rho}\right)_{l,j}^{n} + p_L^{n} \right) u_{l,j}^{n+1} A_{j} \right] \frac{\Delta t}{V_L} - \\
& p_L^{n} d\tilde{\alpha}_{l,L}^{n} + \left(\tilde{h}_{wl,L}^{n} \left(\tilde{T}_{w,L}^{n} - \tilde{T}_{l,L}^{n} \right) S_{wl,L}^{n} - V_L \tilde{\Gamma}_{g,L}^{n} \tilde{h}_{l,L}^{n} \right) \frac{\Delta t}{V_L}.
\end{aligned} \tag{37}$$

In the semi-implicit algorithm these equations are solved explicitly for every control volume. Note that the overall implicitness of the solution can be slightly elevated at no cost by taking pressure values at the new time step. Note also that various modifications can be justified only after extensive numerical experimentation.
The outlined algorithm is perhaps one of the most efficient approaches to the solution of the two-phase Wallis model. It involves inversion of N 4×4 matrices, inversion of N 2×2 matrices, solution of tridiagonal matrix equation with $N\times N$ matrix, and $N\times m$ $N\times m$ explicit computations. The algorithm scales linearly with the number of control volumes.
In practice, however, we were not able to exceed time step limit 5 ms. This poses some limitations for the real time applications to the autonomous control of cryogenic loading. That is why nearly-implicit approach that allows to increase time step several times is considered in the following sections. But before we do, let us briefly summarize the steps involved in the semi-implicit algorithm.

Summary of the semi-implicit algorithm

FIRST STEP (predictions)

1. Solve sum and difference momenta equations in non-conservative form with respect to pressure corrections
2. Solve last in the set of expanded equations in with respect to new velocities corrections in terms of pressure.
3. Substitute new velocities from 2 into solution 1 and solve resulting tridiagonal equation for the pressure
4. Find new velocities using new pressure obtained at the previous step
5. Find provisional values for energies, void fractions, and temperatures
6. Find provisional values of mass exchange rate and heat transfer coefficients using provisional values of temperatures obtained in step 5

SECOND STEP (corrections): Using conservative forms of the mass and energy conservation equations find new values of the

- densities,
- void fractions, and
- energies

.

Nearly-implicit algorithm

The key idea behind the nearly-implicit algorithm is to elevate the implicitness of the scheme to the level required to break material CFL number limitations (23). In this respect two main corrections were introduced. The kinetic energy in the momenta equations and the conservative forms of the density and energy equations are solved implicitly. These corrections require some significant changes in the solution strategy (see [2]), which we describe below.

First step of the nearly-implicit algorithm

In the first step of the nearly–implicit algorithm we solve the coupled pressure-density-energy equations. Although this step is similar to the first step of the semi-implicit algorithm, the non-local coupling between pressure and velocity in the momenta equations requires a different order of the solution and a different integration scheme for the solution of velocity equations.

Approximation of the momenta equations for nearly-implicit scheme

The most significant changes are introduced into the discretized momenta equations (28) and (29) and as a result into the solution of the velocity-density-energy coupling equations. The artificial diffusion terms (27) are no longer required for stabilization. But the kinetic energy terms are treated implicitly. However, in the spirit of maximum computational efficiency these terms are linearized with respect to new velocities

$$\left(u_g^2\right)_L^{n+1} = \left(u_{g,L}^{n+1} - u_{g,L}^n\right)^2 + 2u_{g,L}^{n+1}u_{g,L}^n - \left(u_{g,L}^n\right)^2 \approx 2u_{g,L}^{n+1}u_{g,L}^n - 2\left(u_{g,L}^n\right)^2 + \left(u_{g,L}^n\right)^2 = 2du_{g,L}^{n+1}u_{g,L}^n + \left(u_{g,L}^n\right)^2. \quad (38)$$

With these modifications the momenta equations read

$$\begin{aligned}
&\left[\left(\overline{\alpha\rho}\right)_{g,j}^n du_{g,j}^{n+1} + \left(\overline{\alpha\rho}\right)_{l,j}^n du_{l,j}^{n+1}\right]\Delta x_j + \left(\frac{1}{2}\left(\widehat{\alpha\rho}\right)_{g,j}^n \left(2du_{g,L}^{n+1}u_{g,L}^n + \left(u_{g,L}^n\right)^2 - 2du_{g,L-1}^{n+1}u_{g,L}^n - \left(u_{g,L-1}^n\right)^2\right) + \right. \\
&\left. \frac{1}{2}\left(\widehat{\alpha\rho}\right)_{l,j}^n \left(2du_{l,L}^{n+1}u_{l,L}^n + \left(u_{l,L}^n\right)^2 - 2du_{l,L-1}^{n+1}u_{l,L}^n - \left(u_{l,L-1}^n\right)^2\right)\right)\Delta t = -\left(dp_L^{n+1} - dp_{L-1}^{n+1}\right)\Delta t - \left(p_L^n - p_{L-1}^n\right)\Delta t + \\
&\left[-\rho_{m,j}^n g\Delta z_j - \left(\overline{\alpha\rho}\right)_{g,j}^n F_{wg,j}^n \left(du_{g,j}^{n+1} + u_{g,j}^n\right) - \left(\overline{\alpha\rho}\right)_{l,j}^n F_{wl,j}^n \left(du_{l,j}^{n+1} + u_{l,j}^n\right) - \Gamma_{g,j}^n \left(du_{g,j}^{n+1} + u_{g,j}^n - du_{l,j}^{n+1} - u_{l,j}^n\right)\right]\Delta t\Delta x_j
\end{aligned} \quad (39)$$

$$\begin{aligned}
&\left(1 + \frac{c\left(\rho_m\right)^2}{\overline{\rho}_g\overline{\rho}_l}\right)_j^n \left[du_{g,j}^{n+1} - du_{l,j}^{n+1}\right]\Delta x_j + \frac{\Delta t}{2}\left(\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{g,j}^n \left(2du_{g,L}^{n+1}u_{g,L}^n + \left(u_{g,L}^n\right)^2 - 2du_{g,L-1}^{n+1}u_{g,L}^n - \left(u_{g,L-1}^n\right)^2\right) - \right. \\
&\left. \left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{l,j}^n \left(2du_{l,L}^{n+1}u_{l,L}^n + \left(u_{l,L}^n\right)^2 - 2du_{l,L-1}^{n+1}u_{l,L}^n - \left(u_{l,L-1}^n\right)^2\right)\right) = -\left(\frac{\overline{\rho}_l - \overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)\left(p_L^{n+1} - p_{L-1}^{n+1}\right)\Delta t - \left\{F_{wg,j}^n \left(du_{g,j}^{n+1} + u_{g,j}^n\right) - \right. \\
&F_{wl,j}^n \left(du_{l,j}^{n+1} + u_{l,j}^n\right) - \Gamma_{g,j}^n \frac{\rho_{m,j}^n du_{l,j}^{n+1} - \left(\overline{\alpha\rho}\right)_{g,j}^n du_{l,j}^{n+1} - \left(\overline{\alpha\rho}\right)_{l,j}^n du_{g,j}^{n+1}}{\left(\overline{\alpha\rho}\right)_{g,j}^n \left(\overline{\alpha\rho}\right)_{l,j}^n} - \Gamma_{g,j}^n \frac{\rho_{m,j}^n u_{l,j}^n - \left(\overline{\alpha\rho}\right)_{g,j}^n u_{l,j}^n - \left(\overline{\alpha\rho}\right)_{l,j}^n u_{g,j}^n}{\left(\overline{\alpha\rho}\right)_{g,j}^n \left(\overline{\alpha\rho}\right)_{l,j}^n} \\
&\left.\left(F_I\rho\right)_j^n \left(du_{g,j}^{n+1} + u_{g,j}^n - du_{l,j}^{n+1} - u_{l,j}^n\right)\right\}\Delta t\Delta x_j
\end{aligned} \quad (40)$$

Note that coupling between velocities and pressure has become nonlocal. Before writing these equations in matrix form we express volume centered increments for the velocities $du^{n+1}_{g(l),L}$ are approximated using eqs. (26). Accordingly, the new velocities in equations (39), (40) are located at nodes $j-1$, j and $j+1$. Because of the nonlocal character of this coupling the solution of eqs. (39), (40) with respect to pressure in terms of velocities becomes impossible. Instead these equations are solved as follows.

Solution of the coupled momenta and expanded density and energy equations

To solve coupled pressure-density-energy equation in the nearly-implicit scheme we notice that the expanded set of equations (34) does not change. Accordingly, the last row of the matrix equation

$$A^n_x \begin{bmatrix} d\alpha^{n+1}_{g,L} \\ de^{n+1}_{g,L} \\ de^{n+1}_{l,L} \\ dp^{n+1}_{L} \end{bmatrix} = a^n_x du^{n+1}_{g,j+1} + b^n_x du^{n+1}_{g,j} + c^n_x du^{n+1}_{l,j+1} + d^n_x du^{n+1}_{l,j} + e^n_x$$

can be used to express new pressure in terms of new velocities

$$dp^{n+1}_L = \left(A^n_x\right)^{-1}_4 \left(a^n_x du^{n+1}_{g,j+1} + b^n_x du^{n+1}_{g,j} + c^n_x du^{n+1}_{l,j+1} + d^n_x du^{n+1}_{l,j} + e^n_x\right). \qquad (41)$$

This solution can now be substituted into the momenta equations

$$B^n_u \begin{bmatrix} du \\ dv \end{bmatrix}^{n+1}_{j-1} + C^n_u \begin{bmatrix} du \\ dv \end{bmatrix}^{n+1}_{j} + D^n_u \begin{bmatrix} du \\ dv \end{bmatrix}^{n+1}_{j+1} = b_0 + b_1 dp^{n+1}_L + b_2 dp^{n+1}_{L-1}$$

resulting in the matrix equation for the velocity with *N*+1 blocks of size (2x6) of the form (the following example is given for *N* = 3 control volumes)

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} & 0 & 0 & 0 & 0 \\ m_{21} & m_{22} & m_{23} & m_{24} & 0 & 0 & 0 & 0 \\ m_{31} & m_{32} & m_{33} & m_{34} & m_{35} & m_{36} & 0 & 0 \\ m_{41} & m_{42} & m_{43} & m_{44} & m_{45} & m_{46} & 0 & 0 \\ 0 & 0 & m_{51} & m_{52} & m_{53} & m_{54} & m_{55} & m_{56} \\ 0 & 0 & m_{61} & m_{62} & m_{63} & m_{64} & m_{65} & m_{66} \\ 0 & 0 & 0 & 0 & m_{71} & m_{72} & m_{73} & m_{74} \\ 0 & 0 & 0 & 0 & m_{81} & m_{82} & m_{83} & m_{84} \end{bmatrix} \begin{bmatrix} du^{n+1}_{g,1} \\ du^{n+1}_{l,1} \\ du^{n+1}_{g,2} \\ du^{n+1}_{l,2} \\ du^{n+1}_{g,3} \\ du^{n+1}_{l,3} \\ du^{n+1}_{g,4} \\ du^{n+1}_{l,4} \end{bmatrix} = \begin{bmatrix} n^n_1 \\ n^n_3 \\ n^n_3 \\ n^n_4 \\ n^n_5 \\ n^n_6 \\ n^n_7 \\ n^n_8 \end{bmatrix} \qquad (42)$$

The explicit form of the coefficients of this matrix equation is the following. For the 1st row in (2x6) block we have 6 coefficients in the form

$$m_{2k+1} = \left\{ \left(\left(\widehat{\alpha\rho}\right)^n_{g,j} b_{g,L-1} u^n_{g,L-1} - a_{p2,L-1} \right)\Delta t, \left(\left(\widehat{\alpha\rho}\right)^2_{l,j} b_{l,L-1} u^n_{l,L-1} - a_{p4,L-1} \right)\Delta t, \right.$$
$$\left(\left(\overline{\alpha\rho}\right)^n_{g,j} \Delta x_j + \left(\left(\overline{\alpha\rho}\right)^n_{g,j} F^n_{wg,j} + \Gamma^n_{g,j} \right)\Delta x_j + \left(\widehat{\alpha\rho}\right)^n_{g,j} \left(b_{g,L} u^n_{g,L} - a_{g,L-1} u^n_{g,L-1} \right) + \left(a_{p2,L} - a_{p1,L-1} \right) \right)\Delta t,$$
$$\left(\left(\overline{\alpha\rho}\right)^n_{l,j} \Delta x_j + \left(\left(\overline{\alpha\rho}\right)^n_{l,j} F^n_{wl,j} - \Gamma^n_{g,j} \right)\Delta x_j + \left(\widehat{\alpha\rho}\right)^n_{l,j} \left(b_{lg,L} u^n_{l,L} - a_{l,L-1} u^n_{l,L-1} \right) + \left(a_{p4,L} - a_{p3,L-1} \right) \right)\Delta t,$$
$$\left. \left(\left(\widehat{\alpha\rho}\right)^n_{g,j} a_{g,L} u^n_{g,L} + a_{p1,L} \right)\Delta t, \left(\left(\widehat{\alpha\rho}\right)^n_{l,j} a_{l,L} u^n_{l,L} + a_{p3,L} \right)\Delta t \right\};$$

For the 2nd row in this block 6 coefficients corresponding to the difference of momenta equations are

$$m_{2k,1}=\left(\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{g,j}^{n}b_{g,L-1}u_{g,L-1}^{n}-\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)a_{p2,L-1}\right)\Delta t;$$

$$m_{2k,2}=\left(\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{l,j}^{n}b_{l,L-1}u_{l,L-1}^{n}-\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)a_{p4,L-1}\right)\Delta t;$$

$$m_{2k,3}=\left(1+\frac{c\left(\overline{\rho}_m^n\right)^2}{\overline{\rho}_g^n\overline{\rho}_l^n}\right)_j\Delta x_j+\Delta t\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{g,j}^{n}\left(b_{g,L}u_{g,L}^{n}-a_{l,L-1}u_{l,L-1}^{n}\right)+F_{wg,j}^{n}\Delta x_j\Delta t-$$

$$\frac{\Gamma_{g,j}^{n}}{\left(\overline{\alpha\rho}\right)_{g,j}^{n}}\left(\frac{\lambda_j^n\overline{\rho}_{m,j}^{n}}{\left(\overline{\alpha\rho}\right)_{l,j}^{n}}-1\right)\Delta x_j\Delta t+\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)\left(a_{p2,L}-a_{p1,L-1}\right)\Delta t+\overline{\rho}_{m,j}^{n}F_{i,j}^{n}\Delta x_j\Delta t;$$

$$m_{2k,4}=-\left(1+\frac{c\left(\overline{\rho}_m^n\right)^2}{\overline{\rho}_g^n\overline{\rho}_l^n}\right)_j\Delta x_j-\Delta t\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{l,j}^{n}\left(b_{l,L}u_{l,L}^{n}-a_{l,L-1}u_{l,L-1}^{n}\right)-F_{wl,j}^{n}\Delta x_j\Delta t-$$

$$\frac{\Gamma_{g,j}^{n}}{\left(\overline{\alpha\rho}\right)_{l,j}^{n}}\left(\frac{\left(1-\lambda_j^n\right)\overline{\rho}_{m,j}^{n}}{\left(\overline{\alpha\rho}\right)_{g,j}^{n}}-1\right)\Delta x_j\Delta t+\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)\left(a_{p4,L}-a_{p3,L-1}\right)\Delta t-\overline{\rho}_{m,j}^{n}F_{i,j}^{n}\Delta x_j\Delta t;$$

$$m_{2k,5}=\left(\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{g,j}^{n}a_{g,L}u_{g,L}^{n}+\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)a_{p1,L}\right)\Delta t;\quad m_{2k,6}=\left(-\left(\frac{\widehat{\alpha\rho}}{\overline{\alpha\rho}}\right)_{g,j}^{n}a_{l,L}u_{l,L}^{n}+\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)a_{p3,L}\right)\Delta t.$$

The components of the vector at the right hand side n_{2k+1}^{n} and n_{2k}^{n} have form

$$n_{2k+1}^{n}=-\left(-\overline{\rho}_{m,j}^{n}g\Delta z_j+\left(\widehat{\alpha\rho}\right)_{g,j}^{n}F_{wg,j}^{n}u_{g,j}^{n}+\left(\widehat{\alpha\rho}\right)_{l,j}^{n}F_{wl,j}^{n}u_{l,j}^{n}+\Gamma_{g,j}^{n}\left(u_{g,j}^{n}-u_{l,j}^{n}\right)\right)\Delta x_j\Delta t-$$

$$\frac{\left(\overline{\alpha\rho}\right)_{g,j}^{n}}{2}\left(\left(u_{g,L}^{n}\right)^2-\left(u_{g,L-1}^{n}\right)^2\right)\Delta t-\frac{\left(\overline{\alpha\rho}\right)_{l,j}^{n}}{2}\left(\left(u_{l,L}^{n}\right)^2-\left(u_{l,L-1}^{n}\right)^2\right)\Delta t-\left(p_L^n-p_{L-1}^{n}+a_{p5,L}^{n}-a_{p5,L-1}^{n}\right)\Delta t;$$

and

$$n_{2k}^{n}=\overline{\rho}_{m,j}^{n}g\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)\Delta y_L^n\Delta t-\frac{\left(\overline{\alpha\rho}\right)_{g,j}^{n}}{2}\left(\left(u_{g,L}^{n}\right)^2-\left(u_{g,L-1}^{n}\right)^2\right)\Delta t+\frac{\left(\overline{\alpha\rho}\right)_{l,j}^{n}}{2}\left(\left(u_{l,L}^{n}\right)^2-\left(u_{l,L-1}^{n}\right)^2\right)\Delta t+$$

$$\left[\frac{\Gamma_{g,j}^{n}\overline{\rho}_{m,j}^{n}}{\left(\overline{\alpha\rho}\right)_{g,j}^{n}\left(\overline{\alpha\rho}\right)_{l,j}^{n}}\left(\lambda_j^n u_{g,j}^{n}-\left(1-\lambda_j^n\right)u_{l,j}^{n}\right)-\frac{\Gamma_{g,j}^{n}}{\left(\overline{\alpha\rho}\right)_{g,j}^{n}}u_{g,j}^{n}-\frac{\Gamma_{g,j}^{n}}{\left(\overline{\alpha\rho}\right)_{l,j}^{n}}u_{l,j}^{n}\right]\Delta x_j\Delta t-$$

$$\left(F_{wg,j}^{n}u_{g,j}^{n}-F_{wl,j}^{n}u_{l,j}^{n}\right)\Delta x_j\Delta t-\left(\frac{\overline{\rho}_l-\overline{\rho}_g}{\overline{\rho}_l\overline{\rho}_g}\right)\left(p_L^n-p_{L-1}^{n}+a_{p5,L}^{n}-a_{p5,L-1}^{n}\right)\Delta t-\overline{\rho}_{m,j}^{n}F_{i,j}^{n}\left(u_{g,j}^{n}-u_{l,j}^{n}\right)\Delta x_j\Delta t;$$

The new velocities obtained by solving (42) are substituted into (41) to find new pressures. The first step of the algorithm is completed by solving the rest of the expanded system of equations for density and energy and obtaining provisional values of void fractions, densities, energies, and temperatures. This part of the algorithm is equivalent to the semi-implicit scheme. The second step

of the algorithm is more involved because of the elevated implicitness of the unexpanded density and energy equations as we describe below in more details.

Second step of the nearly-implicit algorithm

The second step consists of solution of the un-expanded equations for the density (15) and energy (20) conservation using elevated implicitness. Let us consider first gas density equation integrated over the control volume.

$$d\left(\alpha\rho\right)_{g,L}^{n+1}=\left(d\left(\alpha\rho\right)_{g,j+1}^{n+1}+\left(\alpha\rho\right)_{g,j+1}^{n}\right)u_{g,j+1}^{n+1}\frac{A_{j+1}\Delta t}{V_L}-\left(d\left(\alpha\rho\right)_{g,j}^{n+1}+\left(\alpha\rho\right)_{g,j}^{n}\right)u_{g,j}^{n+1}\frac{A_j\Delta t}{V_L}=\tilde{\Gamma}_{g,L}^{n}\Delta t\,. \qquad (43)$$

Similarly, for the liquid density we have

$$d\left(\alpha\rho\right)_{l,L}^{n+1}=\left(d\left(\alpha\rho\right)_{l,j+1}^{n+1}+\left(\alpha\rho\right)_{l,j+1}^{n}\right)u_{l,j+1}^{n+1}\frac{A_{j+1}\Delta t}{V_L}-\left(d\left(\alpha\rho\right)_{l,j}^{n+1}+\left(\alpha\rho\right)_{l,j}^{n}\right)u_{l,j}^{n+1}\frac{A_j\Delta t}{V_L}=-\tilde{\Gamma}_{g,L}^{n}\Delta t\,. \qquad (44)$$

We note that interface values $\left(\alpha\rho\right)_{g(l),j}^{n+1}$ are obtained using 2nd equation in (24). With such a substitution the solution of the density equations is reduced to the solution of the tridiagonal matrix. The latter is solved using algorithm described in Appendix C.

For the gas energy equations we obtain

$$\begin{aligned}&d\left(\alpha\rho e\right)_{g,L}^{n+1}=\left(d\left(\alpha\rho e\right)_{g,j+1}^{n+1}+\left(\alpha\rho e\right)_{g,j+1}^{n}+\left(\alpha\rho\right)_{g,j+1}^{n}p_L^{n+1}\right)u_{g,j+1}^{n+1}\frac{A_{j+1}\Delta t}{V_L}-\\&\left(d\left(\alpha\rho e\right)_{g,j}^{n+1}+\left(\alpha\rho e\right)_{g,j}^{n}+\left(\alpha\rho\right)_{g,j}^{n}p_L^{n+1}\right)u_{g,j}^{n+1}\frac{A_j\Delta t}{V_L}=-p_L^n d\tilde{\alpha}_{g,L}^{n}+\left(\tilde{h}_{wg,L}^{n}\left(\tilde{T}_{w,L}^{n}-\tilde{T}_{g,L}^{n}\right)S_{wg,L}^{n}+V_L\tilde{\Gamma}_{g,L}^{n}\tilde{h}_{g,L}^{n}\right)\frac{\Delta t}{V_L}.\end{aligned} \qquad (45)$$

Similarly, for the liquid energy we have

$$\begin{aligned}&d\left(\alpha\rho e\right)_{l,L}^{n+1}=\left(d\left(\alpha\rho e\right)_{l,j+1}^{n+1}+\left(\alpha\rho e\right)_{l,j+1}^{n}+\left(\alpha\rho\right)_{l,j+1}^{n}p_L^{n+1}\right)u_{l,j+1}^{n+1}\frac{A_{j+1}\Delta t}{V_L}-\\&\left(d\left(\alpha\rho e\right)_{l,j}^{n+1}+\left(\alpha\rho e\right)_{l,j}^{n}+\left(\alpha\rho\right)_{l,j}^{n}p_L^{n+1}\right)u_{l,j}^{n+1}\frac{A_j\Delta t}{V_L}=p_L^n d\tilde{\alpha}_{g,L}^{n}+\left(\tilde{h}_{wl,L}^{n}\left(\tilde{T}_{w,L}^{n}-\tilde{T}_{l,L}^{n}\right)S_{wl,L}^{n}-V_L\tilde{\Gamma}_{g,L}^{n}\tilde{h}_{l,L}^{n}\right)\frac{\Delta t}{V_L}.\end{aligned} \qquad (46)$$

To find primitive variables from the set of new conservative variables

$$\left\{\left(\alpha\rho\right)_{g,L}^{n+1},\left(\alpha\rho\right)_{l,L}^{n+1},\left(\alpha\rho e\right)_{g,L}^{n+1},\left(\alpha\rho e\right)_{l,L}^{n+1}\right\}$$

we notice that gas and liquid energies are readily obtained as

$$e_{g,L}^{n+1}=\left(\alpha\rho e\right)_{g,L}^{n+1}/\left(\alpha\rho\right)_{g,L}^{n+1};\qquad e_{l,L}^{n+1}=\left(\alpha\rho e\right)_{l,L}^{n+1}/\left(\alpha\rho\right)_{l,L}^{n+1}.$$

To find void fractions and densities are found using

$$\alpha_{l,L}^{n+1}=\left(\alpha\rho\right)_{l,L}^{n+1}/\tilde{\rho}_{l,L}^{n+1};\qquad \alpha_{g,L}^{n+1}=1-\alpha_{l,L}^{n+1};\qquad \rho_{g,L}^{n+1}=\left(\alpha\rho\right)_{g,L}^{n+1}/\alpha_{g,L}^{n+1}.$$

Phase appearance/disappearance

Numerical experimentation with semi-implicit and nearly-implicit algorithms reveals that one of the most sever stability issues is related to the phase appearance/disappearance. The second critical issue in phase-transition is the lack of positivity [25], that frequently appear in the control volumes with small mass fraction of one of the phases. In the latter case, even small inaccuracies easily lead to negative mass fractions, especially with large time-steps.

Phase appearance and disappearance

According to [4] appearance and disappearance presents very serious numerical challenge for all effective field models and no fully satisfactory solutions exist, but several *ad hoc* numerical treatments used in different codes.
To get further insight into the physical origin of this problem we follow [25] and notice that in the limit of vanishing volume fraction of one of the phases, the two phases almost decouple and the minority phase obeys a pressureless gas dynamics system. Indeed, consider the reduced model of isothermal two-phase flow with the non-hyperbolicity corrected in the form of [26]

$$\delta\alpha_g\alpha_l\rho_m\left(u_g-u_l\right)^2\left(\alpha_g\right)_{,x}$$

$$\left(\alpha_g\rho_g\right)_{,t}+\left(\alpha_g\rho_g u_g\right)_{,x}=0$$

$$\left(\alpha_g\rho_g u_g\right)_{,t}+\left(\alpha_g\rho_g u_g^2\right)_{,x}+\alpha_g p_{,x}+\delta\alpha_g\alpha_l\rho_m\left(u_g-u_l\right)^2\left(\alpha_g\right)_{,x}=0$$

$$\left(\alpha_l\rho_g\right)_{,t}+\left(\alpha_l\rho_l u_l\right)_{,x}=0$$

$$\left(\alpha_l\rho_l u_l\right)_{,t}+\left(\alpha_l\rho_l u_l^2\right)_{,x}+\alpha_l p_{,x}+\delta\alpha_g\alpha_l\rho_m\left(u_g-u_l\right)^2\left(\alpha_l\right)_{,x}=0$$

By introducing a small parameter for the vanishing phase $\alpha_g=\varepsilon\alpha_{g0}$ and using relation $\varepsilon\alpha_{g0}+\alpha_l=1$ we have in the limit $\varepsilon\to 0$

$$\left(\alpha_{g0}\rho_g\right)_{,t}+\left(\alpha_{g0}\rho_g u_g\right)_{,x}=0$$

$$\left(\alpha_{g0}\rho_g u_g\right)_{,t}+\left(\alpha_{g0}\rho_g u_g^2\right)_{,x}+\alpha_{g0}p_{,x}+\varepsilon\delta\alpha_{g0}\alpha_l\rho_m\left(u_g-u_l\right)^2\left(\alpha_{g0}\right)_{,x}=0$$

$$\left(\rho_g\right)_{,t}+\left(\rho_l u_l\right)_{,x}=0$$

$$\left(\rho_l u_l\right)_{,t}+\left(\rho_l u_l^2\right)_{,x}+p_{,x}-\varepsilon\delta\alpha_{g0}\alpha_l\rho_m\left(u_g-u_l\right)^2\left(\alpha_l\right)_{,x}=0$$

where correction terms also vanish in this limit. As a result, the liquid dynamics is completely decoupled and the pressure is determined by the liquid dynamics alone. In this case the gas dynamics becomes pressureless, the Jacobian matrix of gas dynamics does not have a complete basis of eigenvectors, and the system of equations becomes non-hyperbolic [27], [28].

Lack of positivity

An additional difficulty related to the use of the semi- and nearly-implicit schemes is the lack of positivity. The notion of "positively conservative" schemes was introduced in [29], where it was shown that while Godunov scheme is positively conservative, i.e. preserve positivity of the density and internal energy, no linearized Riemann solver, including the Roe scheme, is positively conservative.
At present no fully satisfactory solutions exist to these problems, but several *ad hoc* numerical treatments are suggested in different codes. In this work, we follow recommendations by Liou [30] and adjust temperature, velocity, and density according to the following expression

$$\phi_{adj}=g(x)\phi_d+\left(1-g(x)\right)\phi_c;\qquad g(x)=x^2\left(2x-3\right);\qquad x=\frac{\alpha_d-x_{\min}}{x_{\max}-x_{\min}},\tag{47}$$

where “d” stands for disappearing phase and “c” for conducting phase. The exact values of the minimum and maximum void fraction, for which smoothing (47) is applied should be established

wining extensive numerical experimentation. The usual values are of the order of $x_{max} = 10^{-3}$ and $x_{min} = 10^{-7}$.
The role of the limiters and smoothers described in this section is critical for the code stabilization. However, their application is not sufficient to eliminate instabilities. An external loop with time step control is required to improve code performance as described in the following section.

Time step control

The time step control is critical for reliable performance of the algorithm [2]. Its implementation is somewhat different in the semi-implicit and nearly-implicit schemes because of the CFL limitations related to the material waves in the former scheme. Here we describe common step controls for both schemes. The priority is given to the control of the mass and energy conservation. In the present version of the algorithm the time-step control follows very closely the recommendations of [2]. Mass conservation is controlled in two ways. Firstly, the total density

$$\rho_{m,L}^{n+1} = \alpha_{g,L}^{n+1}\rho_{g,L}^{n+1} + \left(1-\alpha_{g,L}^{n+1}\right)\rho_{l,L}^{n+1}$$

for each control volume is calculated using solution of the unexpanded form of the mass and energy conservation equations at the second step of the algorithm (see e.g. eqs. (43)-(46)). These total densities are compared to the densities obtained by Taylor expansion (33)

$$\tilde{\rho}_{m,L}^{n+1} = \tilde{\alpha}_{g,L}^{n}\left(\rho_{g,L}^{n} + \left(\frac{\partial\rho}{\partial p}\right)_{g,L}^{n} dp_{L}^{n+1} + \left(\frac{\partial\rho}{\partial e}\right)_{g,L}^{n} de_{g,L}^{n+1}\right) + \tilde{\alpha}_{l,L}^{n}\left(\tilde{\rho}_{l,L}^{n} + \left(\frac{\partial\rho}{\partial p}\right)_{l,L}^{n} dp_{L}^{n+1} + \left(\frac{\partial\rho}{\partial e}\right)_{l,L}^{n} de_{l,L}^{n+1}\right)$$

in the first step of the algorithm. The maximum error is found as follows

$$Err_{\max} = \max\left(\frac{\rho_{m,L}^{n+1} - \tilde{\rho}_{m,L}^{n+1}}{\tilde{\rho}_{m,L}^{n+1}}\right).$$

Secondly, the error for the total mass in the pipe is found as follows

$$Err_{tot} = \frac{2\sum_{L}\left(\rho_{m,L}^{n+1} - \tilde{\rho}_{m,L}^{n+1}\right)^{2}}{\sum_{L}\left(\tilde{\rho}_{m,L}^{n+1}\right)^{2}}.$$

If either of these errors is larger than 10^{-2} the time step is halved.
In addition, the total mass “in” and “out” of the pipe through the input/output and damp valves is continuously monitored and compared with the total changes of the mass within the pipe. If the difference exceed a few percent of the total mass, the integration is terminated.
We also monitor the mass addition/removal for each phase in every control volume. If mass addition/removal exceeds half of the mass currently present in the control volume the time step is halved.
Despite this control the values of the key thermodynamic variables may occasionally be found to lie outside predefined range. If this happens the time step is halved. The reduction of the time step continues until Δt reached the minimum predefined value (usually 10 mks). If all the field values are found to be inside the range, and both mass errors are smaller than 10^{-3}, the time step is doubled until Δt reached the maximum predefined value (usually 10 ms).

Conclusions

Using limiters and smoothers to control of the phase appearance/disappearance in conjunction with external loop for the time-step control a reliable and stable performance of the semi-implicit and nearly-implicit algorithms is achieved.

Note that a significant part of the algorithm related to the suitable choice of the source terms and the discussion of the verification of the algorithm performance will be discussed in separate parts of this report. The reason is that the functional form of the source terms depend ssubstantially on the application of the algorithm. Furthermore, currently the exact form of the source terms is not available/known for the application to the real time control of cryogenic loading. Accordingly, in the Part III of the report we will discuss a few possible approaches to the solution of this problem. In addition, we will provide example of a full set of complete correlations for the flow patterns, friction losses and heat transfer that were developed for two industrial applications.

Appendices

Appendix A. Equations of the two-phase flow

Various approaches can be utilized in the derivation of the two-phase flow equations (see e.g. [8]-[11],[18],[19]). Here we follow mainly derivation proposed by H. Staedtke [8], which in turn follows closely derivation [20] based on the averaging method [21]. We note that this derivation does not provide first principles proof of the two-phase flow equations. However, it highlight the key properties of the flow that have to be understood for such proof to become possible. The method [20], [21] can be shortly summarized as follows (repeated from [31])

- Introduce phase indicator $\chi_k(x,t)=\begin{cases}1, & if \quad x \quad belongs\ to\ phase\ k,\\ 0, & otherwise.\end{cases}$ Note that the gradient of the distribution function is zero everywhere except the interface. Then the unit vector at the interface is defined as $\vec{n}_k^{\,\mathrm{int}}=-\left(\nabla\gamma_k/\left|\nabla\gamma_k\right|\right)^{\mathrm{int}}$. The $\nabla\gamma_k$ is assumed to have properties of the delta-function.
- To shorten notations introduce volume average as $\langle\psi\rangle=\frac{1}{V}\int_V\psi dV$
- then Gauss theorem reads $\langle\chi_k\nabla\psi\rangle=\langle\nabla(\chi_k\psi)\rangle-\langle\psi\nabla\chi_k\rangle$
- and Leibniz rule reads $\langle\chi_k\psi_t\rangle=\langle(\chi_k\psi)_t\rangle-\langle\psi(\chi_k)_t\rangle$
- taking into account (*ad hoc* no surface break up or coalition) that for an observer moving with the interface χ_k does not change in time $(\chi_k)_t+u_0\nabla\chi_k=0$.
- The mass conservation after averaging takes the form

$\langle\chi_k\rho_{,t}\rangle+\langle\chi_k\nabla(\rho u)\rangle=\langle(\chi_k\rho)_{,t}\rangle-\langle\rho(\chi_k)_{,t}\rangle+\langle\nabla(\chi_k\rho u)\rangle-\langle\rho u\nabla(\chi_k)\rangle=0$ or

$\langle(\chi_k\rho)_{,t}\rangle+\langle\nabla(\chi_k\rho u)\rangle=\langle\rho(\chi_k)_{,t}\rangle+\langle\rho u\nabla(\chi_k)\rangle$ and taking into account equation for χ_k we have $\langle(\chi_k\rho_k)_{,t}\rangle+\langle\nabla(\chi_k\rho_k u_k)\rangle=\langle\rho_k(u_k-u_0)\nabla(\chi_k)\rangle$. In derivation we take into account that χ_k is nonzero only when $\rho=\rho_k$. Note that the term on the right hand side is the volumetric source term for the inter-phase mass flow into phase k.

- Similarly for the momentum equation we have
  $\langle\chi_k\rho_k u_k\rangle+\nabla\langle\chi_k\rho_k u_k^2\rangle+\nabla\langle\chi_k p\rangle=\langle p\nabla\chi_k\rangle+\langle\rho_k u_k(u_k-u_0)\nabla\chi_k\rangle$
- And for the total energy we have
  $\langle\chi_k\rho_k E_k\rangle+\nabla\langle\chi_k\rho_k u_k(E_k+p/\rho_k)\rangle=\langle p\,u_k\nabla\chi_k\rangle+\langle\rho_k E_k(u_k-u_0)\nabla\chi_k\rangle$

Note that this derivation reproduces non-hyperbolic character of the model. Note also terms that contain $\nabla\chi_k$ and correspond to the interfacial contributions. Specifically, terms proportional to the

$\rho_k(u_k - u_0)\nabla(\chi_k)$ correspond to the terms in the model (14) that are proportional to the interfacial mass flow Γ_a.

Appendix B. Single-phase flow equations.

To contrast a formal derivation above to a simplified derivation of the single- and two-phase flow in a pipe we follow here a discussion in [19]. For simplicity, we consider rectangular pipe with stratified flow shown in the Fig. 4. Let us assume initially a single-phase flow when the colored part of the flow represents the entire control volume with dimensions $\Delta x_1, \Delta x_2$, and Δx_3 and flow direction along x_1.

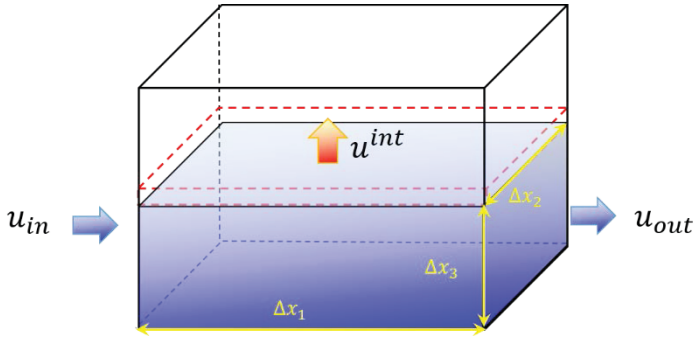

Figure 4 Rectangular pipe with stratified flow.

In this case mass conservation for a given control volume reads

$$\frac{d}{dt}(\rho)\Delta x_1\Delta x_2\Delta x_3 = j_1(x_1)\Delta x_2\Delta x_3 - j_1(x_1+\Delta x_1)\Delta x_2\Delta x_3 = \left[(\rho u)_{x_1} - (\rho u)_{x_1+\Delta x_1}\right]\Delta x_2\Delta x_3 .$$

Let us consider now the case of two-phase flow with interfacial mass flux. Because of the mass exchange, the volume of liquid will also change as shown by the red dashed lines. In this case Δx_3 becomes function of time and the mass conservation for liquid in a given control volume takes the form

$$\frac{1}{\Delta t}\left(\rho(t+\Delta t)(\Delta x_3 + u_0\Delta t) - \rho(t)\Delta x_3\right)\Delta x_1\Delta x_2 = j_1(x_1)\Delta x_2 - j_1(x_1+\Delta x_1)\Delta x_2\Delta x_3 + \Gamma_l\Delta V$$

where $u_0\Delta t$ is the change of the Δx_3, u_0 is the velocity of the interface surface, and the interfacial mass flow is given by

$$\Gamma_l\Delta V = \rho u_0\Delta x_1\Delta x_2 .$$

Here $\Delta x_1\Delta x_2$ is the interface area and Γ_l is normalized to the control volume.

Appendix C. Standard tridiagonal solver

```
void solve_tridiagonal_in_place_destructive(double x[], const size_t N, const
double a[], const double b[], double c[]) {
/* unsigned integer of same size as pointer */
size_t in;
/*
solves Ax = v where A is a tridiagonal matrix consisting of vectors a, b, c
```

```
note that contents of input vector c will be modified, making this a one-time-use function
x[] - initially contains the input vector v, and returns the solution x. indexed from [0, ..., N - 1]
N - number of equations
a[] - subdiagonal (means it is the diagonal below the main diagonal) -- indexed from [1, ..., N - 1]
b[] - the main diagonal, indexed from [0, ..., N - 1]
c[] - superdiagonal (means it is the diagonal above the main diagonal) -- indexed from [0, ..., N - 2]
*/
c[0] = c[0] / b[0];
x[0] = x[0] / b[0];
/* loop from 1 to N - 1 inclusive */
for (in = 1; in < N; in++) {
double m = 1.0 / (b[in] - a[in] * c[in - 1]);
c[in] = c[in] * m;
x[in] = (x[in] - a[in] * x[in - 1]) * m;
}
/* loop from N - 2 to 0 inclusive, safely testing loop end condition */
for (in = N - 1; in-- > 0; )
x[in] = x[in] - c[in] * x[in + 1];
}
```

Appendix D. Hyperbolic analysis of the model equations

Below we use vector V where we substituted alpha with r only to avoid differentiation of alpha

```
b := 'b';
 U := [alpha, u, h, v, p, H];
V := [r, u, h, v, p, H];
a := Vector([alpha*rho, alpha*rho*u, alpha*rho*h, (1-alpha)*r, (1-alpha)*r*v, (1-alpha)*r*H]);
 b := Vector([alpha*rho*u, alpha*rho*u^2, alpha*rho*h*u, (1-alpha)*r*v, (1-alpha)*r*v^2, (1-
alpha)*v*r*H]);
c := Vector([0, 0, -alpha*p, 0, 0, -(1-alpha)*p]);
d := Vector([0, alpha*p, 0, 0, (1-alpha)*p, 0]);
A := Jacobian(a, U);
S := Jacobian(c, V);
M := A-S;
B := Jacobian(b, U);
Y := Jacobian(d, V);
N := B-Y;
Ev := simplify(Eigenvalues(M, N));
egv := Ev[[3, 4]];
egv := Ev[[1, 2]];
egv := Ev[[5, 6]]

Cm := 'Cm';
x := 'x';
Cm := R*x*(1-x)*rm*(u-v);
a := Vector([alpha*rho, alpha*rho*u+Cm, alpha*rho*h, (1-alpha)*r, (1-alpha)*r*v-Cm, (1-
alpha)*r*H]);
b := Vector([alpha*rho*u, alpha*rho*u^2, alpha*rho*h*u, (1-alpha)*r*v, (1-alpha)*r*v^2, (1-
alpha)*v*r*H]);
c := Vector([0, 0, -alpha*p, 0, 0, -(1-alpha)*p]);
d := Vector([0, alpha*p, 0, 0, (1-alpha)*p, 0]);
A := Jacobian(a, U);
```

```
S := Jacobian(c, V);
M := A-S;
B := Jacobian(b, U);
Y := Jacobian(d, V);
N := B-Y;
x := alpha;
Ev := simplify(Eigenvalues(M, N));
egv := Ev[[3, 4]];
egv := Ev[[1, 2]];
egv := Ev[[5, 6]]

S := 'S';
S := R^2*alpha^2*rm^2*u^2-2*R^2*alpha^2*rm^2*u*v+R^2*alpha^2*rm^2*v^2-
2*R^2*alpha*rm^2*u^2+2*R^2*alpha*rm^2*u*v-
4*R*alpha^2*r*rm*u*v+4*R*alpha^2*r*rm*v^2+4*R*alpha^2*rho*rm*u^2-
4*R*alpha^2*rho*rm*u*v+R^2*rm^2*u^2+4*R*alpha*r*rm*u*v-4*R*alpha*r*rm*v^2-
4*R*alpha*rho*rm*u^2+4*R*alpha*rho*rm*u*v+4*alpha^2*r*rho*u^2-
8*alpha^2*r*rho*u*v+4*alpha^2*r*rho*v^2-4*alpha*r*rho*u^2+8*alpha*r*rho*u*v-4*alpha*r*rho*v^2;
s1 := collect(R^2*alpha^2*rm^2*u^2-2*R^2*alpha^2*rm^2*u*v+R^2*alpha^2*rm^2*v^2, [R^2,
alpha^2, rm^2], 'recursive');
s2 := collect(-4*alpha*r*rho*u^2+8*alpha*r*rho*u*v-4*alpha*r*rho*v^2, [r, alpha, rho], 'recursive');
s3 := collect(4*alpha^2*r*rho*u^2-8*alpha^2*r*rho*u*v+4*alpha^2*r*rho*v^2, [r, alpha^2, rho],
'recursive');
s31 := ((-alpha^2+2*alpha-1)*4)*R*rho*rm*u^2-4*R*rho*rm*u(alpha-1)^4;
s5 := -2*R^2*alpha*rm^2*v^2-4*R*alpha^2*r*rm*v^2+R^2*rm^2*v^2;
s6 := 2*R^2*alpha*rm^2*u*v+4*R*alpha^2*r*rm*u*v+4*R*alpha^2*rho*rm*u*v-
8*R*alpha*rho*rm*u*v;
```

References

[1] Committee on Microgravity Research, Space Studies Board Commission on Physical Sciences, Mathematics, and Applications National Research Council, 2001

[2] INEL, 2012. RELAP5/MOD3 Code Manual. Volume I: Code Structure, System Models and Solution Methods INEEL-EXT-98-00834-V1/Rev4, Vol. 1

[3] C.E. Brennen, *Fundamentals of Multiphase Flow*, Cambridge University Press, Cambridge, 2005.

[4] R. Nourgaliev, M. Christon, "Solution Algorithms for Multi-Fluid-Flow Averaged Equations", INL/EXT-12-27187 (151 p.), September 27, 2012

[5] TRACE code development team. TRACE v5.0 Theory Manual: Field equations, Solution methods, and Physical models. http://spot.infosyslabs.com:8080/nrccodes/how to obtain.html.

[6] J.M. Delhaye, Équations fondamentales des écoulements diphasiques; Part 1 and 2. Technical Report CEA-R-3429, CEA, 1968

[7] G. B. Wallis, *One-dimensional Two-phase Flow*, McGraw-Hill, New York, 1969.

[8] H. Staedtke, *Gasdynamic Aspects of Two-Phase Flow: Hyperbolicity, Wave Propagation Phenomena, and Related Numerical Methods*, Darmstadt: John Wiley & Sons, 2006.

[9] M. Ishii, T. Hibiki, Thermo-Fluid Dynamics of Two-Phase Flow. Springer, Bücher, 2010.

[10] V. H. Ransom, Numerical Modeling of Two-Phase Flows for Presentation at Ecole d'Ete d'Analyse Numerique, EGG-EAST-8546, 1989.

[11] A. K. Vij, W. E. Dunn, Modeling of Two-Phase Flows in Horizontal Tubes, ACRCTR-98, University of Illinois Mechanical & Industrial Engineering Dept., May 1996

[12] J.A. Borkowski et al. TRAC-BF1: An advanced best-estimate computer program for BWR accident analysis. TR EGG-2626, US NRC Report NUREG/CR-4356, Idaho National Laboratories, 1992.

[13] D. Bestion, The physical closure laws in the CATHARE code. Nuclear Engineering and Design, 124:229–245, 1990.

[14] J.H. McFadden. RETRAN-02: User's mamual. Technical report, Electric Power Institute, 1981.

[15] D.R. Liles and Wm.H. Reed, A Semi-Implicit method for two-phase fluid dynamics. Journal of Computational Physics, 26:390–407, 1978.

[16] F.H. Harlow and A.A. Amsden, Numerical calculation of almost incompressible flow. Journal of Computational Physics, 3:80–93, 1968.

[17] F.H. Harlow and A.A. Amsden, A numerical fluid dynamics calculation method for all flow speeds. Journal of Computational Physics, 8:197–213, 1971.
[18] D.A. Drew, S.L. Passman, *Theory of Multicomponent Fluids*, Springer; 1998.
[19] Z. Chen, G. Huan, Y. Ma, *Computational Methods for Multiphase Flows in Porous Media*, Society for Industrial and Applied Mathematics; 2006.
[20] R. Saurel and T. Gallouet, Mod`eles et m´ethodes num´eriques pour les ´ecoulements fluides, Cours de DEA, Centre de Math´ematiques et d'Informatique, Universit´e de Provence (1998).
[21] D.A. Drew, Mathematical Modeling of Two-Phase Flow. Annual Review of Fluid Mechanics. 1983;15(1):261-291.
[22] D. Drew, I. Cheng, and R.T. Drew, The Analysis of Virtual Mass Effects in Two-phase Flow, International Journal of Multiphase Flow, 5, 233–242, 1979.
[23] V.A. Mousseau, Implicitly balanced solution of the two-phase flow equations coupled to nonlinear heat conduction. Journal of Computational Physics. 2004;200(1):104-132.
[24] A. S. Shieh, V. H. Ransom, R. Krishnamurthy, INEL, 2001. RELAP5/MOD3 Code Manual. Volume VI: Validation of numerical techniques in relap5/mod3.0, NUREG/CR-5535/Rev 1-Vol VI
[25] F. Cordier, P. Degond, A. Kumbaro, Phase Appearance or Disappearance in Two-Phase Flows. Journal of Scientific Computing. 2013: 1-34, ISSN: 0885-7474.
[26] I. Toumi, A. Kumbaro, and H. Paillere, Approximate Riemann solvers and flux vector splitting schemes for two-phase flow, Von-Karman Institute for fluid dynamics, 1999.
[27] F. Bouchut, On zero pressure gas dynamics, in "Advances in kinetic theory and computing: selected papers (B. Perthame ed.)", World Scientic, 1994, pp. 171-190.
[28] F. Bouchut, S. Jin, and X. Li, Numerical approximations of pressureless and isothermal gas dynamics, SIAM J. Num. Anal., 41, 2004, 135-158.
[29] B. Einfeldt, C. Munz, P. Roe, and B. Sjögreen, On Godunov-type methods near low densities, J. Comput. Phys., 92, 1991, 273-295.
[30] C.H. Chang and M.S. Liou. A robust and accurare approach to computing compressible multiphase flow: Stratified flow model and AUSM+-up scheme. Journal of Computational Physics, 225:840–873, 2007.
[31] Keh-Ming Shyue, Mathematical Models and Numerical Methods for Compressible Multicomponent Flow, Workshop on CFD based on Unified Coordinates– Theory & Applications, Feb. 24-25, 2006 – p. 8/34

Abbreviations

AUSM -- Advection Upstream Splitting Method schemes for hyperbolic systems of conservation laws which do not require any analytical calculation of the Jacobian
RELAP -- Reactor Excursion and Leak Analysis Program
TRAC -- Transient Reactor Analysis Code
TRACE -- TRAC/RELAP Advanced Computational Engine