Presenting the multi-phase solver implemented in the open source TrioCFD code based on the TRUST HPC platform

Corentin Reiss, Antoine Gerschenfeld, Elie Saikali, Yannick Gorsse, Alan Burlot

EPJ Web of Conferences **302**, 03001 (2024) https://doi.org/10.1051/epjconf/202430203001
SNA + MC 2024

Presenting the multi-phase solver implemented in the open source TrioCFD code based on the TRUST HPC platform

Corentin Reiss[1,3*], *Antoine* Gerschenfeld[1], *Elie* Saikali[2**], *Yannick* Gorsse[1], and *Alan* Burlot[1***]

[1]Université Paris-Saclay, CEA, Service de Thermo-hydraulique et de Mécanique des Fluides, 91191, Gif-sur-Yvette, France.
[2]Université Paris-Saclay, CEA, Service Génie Logiciel pour la Simulation, 91191, Gif-sur-Yvette, France.
[3]Institut de Mécanique des Fluides de Toulouse, Université de Toulouse, CNRS, INPT, UPS, Allée du Prof. Camille Soula, 31400 Toulouse, France

Abstract. A new open source CFD-scale multi-phase flow environment has been developed in the TrioCFD code using the TRUST HPC platform. Mass, momentum and energy equations can be solved for an arbitrary number of phases. This environment is scalable, can use multiple numerical schemes and resolution algorithms, and enables easy modification of the physical closures. It has been verified on adiabatic and boiling two-phase simulations.

1 Introduction

Multi-phase flows play a role in many nuclear applications [1]. These flows are extremely complex and a wide variety of flow patterns can exist [2]. For industrial purposes, empirical correlations that are dependent on flow conditions and geometries are used [3]. Mature multi-phase CFD codes, on the other hand, will be able to simulate flows without the need for expensive experimental calibration [4]. To this end, the CEA is developing a multiphase solver in its fluid dynamics code TrioCFD using the TRUST HPC Platform.

Section 2 describes the structure of the code, the numerical methods and turbulence models used and the multi-phase closures that are implemented. Section 3 presents the verification of the implementation of these terms. Work was also carried out on a set of closure relations that can describe high-pressure boiling flows, and be used with any two-fluid 3D code. The construction method and details of this set of closures isn't the subject of this paper. They can be found along with the comparison with experimental results in [5] and [6]. Section 4 discusses the HPC simulations on which work is ongoing at CEA with these closures.

2 TrioCFD multiphase solver

2.1 Inheritance from the open source HPC TRUST platform

The multi-phase solver is developed in the TrioCFD code [7] based on the HPC platform TRUST [8] (github.com/cea-trust-platform). Both the code and the platform are open-source

*e-mail: corentin.reiss@cea.fr
**e-mail: elie.saikali@cea.fr
***e-mail: alan.burlot@cea.fr

(BSD license) and developed at the Energy Division (DES) of the French Atomic and Alternative Energy Commission (CEA). They are based on an object-oriented intrinsically parallel approach and written in the C++ language.

The METIS library is used to perform HPC simulations where the computational domain is splitted into several overlapping sub-domains. METIS always insures a small load imbalance of the domain partition (defined by the product of the maximum cells among sub-domains and the number of sub-domains divided by the total number of cells). As a result, all sub-domains are normally distributed quite equally among different processor cores, which, by using message passing interface libraries (MPI), communicate only with required neighbor processors when data transfer is needed.

All I/O processes are parallelized with the possibility to read and write from a single HDF5 file or from distributed files. When a calculation domain comprises more than $2^{31}/8$ cells ($\sim$ 268.5M cells), the integers need to be represented on 64 bits. This feature is included as a compiler option. The largest simulation ever run on a TRUST architecture contained 2 billion cells [8].

In what follows, a detailed description of the TrioCFD multiphase solver will be presented.

2.2 Numerical methods

The TrioCFD multiphase solver inherits directly form the generic classes of the TRUST platform that define the kernel of this code: numerical methods, spatial and time discretizations, linear systems, equations, boundary conditions, pre/post-processing... The PETSc library is used to solve linear systems.

The software can handle arbitrary meshed geometries where two spatial discretizations can be used. The first is a finite difference volume (VDF) method for conformal cartesian unstructured-hexahedral type meshes [9] (implemented on a staggered grid of type Marker and Cell (MAC) [10]). This scheme can also handle axisymmetric pipe flows. The second, called PolyMAC_P0, is a method for an arbitrary polyhedral conform/non-conform mesh [11, 12]. Figure 1 illustrates mesh types that can be used by multiphase TrioCFD with PolyMAC_P0.

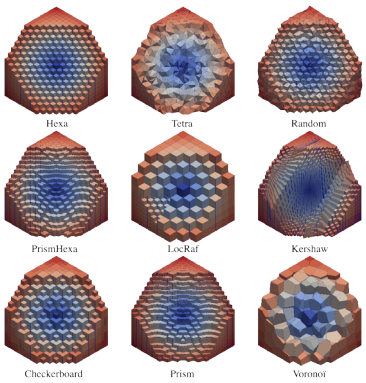

Figure 1: Meshes which can be used by the multiphase TrioCFD solver with PolyMAC_P0.

2.3 The new generic multi-phase problem

The historic single-phase TrioCFD code offers a flexible architecture where the user can specify one or more *problems* consisting each in one or more *equations*. For instance, the `Pb_Thermohydraulique` *problem* describes a single-phase incompressible fluid through a `Navier_Stokes_std` *equation* for the velocity/pressure $(p, \vec{v})$ and a `Convection_Diffusion_Temperature` *equation* for the temperature T. Default classes implement heat conduction problems as well as incompressible and quasi-compressible (thermal-)hydraulics problems. For turbulent simulations, the user can simply define a turbulent problem (invariant of the laminar one, say `Pb_Thermohydraulique_Turbulent`) by associating an appropriate turbulence and boundary layer model to each equation. Both RANS and LES models are available.

Aside from its equations, each problem is associated with a *medium* class (describing its medium properties), a *discretization* class (describing how the equations should be discretized in space), and a *time scheme* class (describing the solution algorithm to be performed at each time step).

In order to describe multi-phase flows, we implemented a new `Pb_Multiphase` class describing the following equations for the k-th phase ($1 \leq k \leq N$):

$$\begin{cases} \dfrac{\partial \alpha_k \rho_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k \vec{v}_k) = \displaystyle\sum_{l \neq k} \Gamma_{kl} & (\mathcal{M}_k) \\ \alpha_k \rho_k \dfrac{\partial \vec{v}_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k \vec{v}_k \otimes \vec{v}_k) - \vec{v}_k \nabla \cdot (\alpha_k \rho_k \vec{v}_k) = & \\ \qquad - \alpha_k \nabla p + \nabla \cdot (\alpha_k \mu_k (\nabla \vec{v}_k + {}^t \nabla \vec{v}_k)) + \displaystyle\sum_{l \neq k} \vec{F}^i_{kl} + \vec{F}^w_k & (\mathcal{Q}_k) \\ \dfrac{\partial \alpha_k \rho_k e_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k e_k \vec{v}_k) = -p \left(\dfrac{\partial \alpha_k}{\partial t} + \nabla \cdot (\alpha_k \vec{v}_k) \right) & \\ \qquad + \nabla \cdot (\alpha_k \lambda_k \nabla T_k) + \displaystyle\sum_{l \neq k} (q^i_{kl} + \Gamma_{kl} h_k) + q^w_k & (\mathcal{E}_k) \end{cases} \tag{1}$$

The primary unknowns of this system are the volumetric fractions (α_k), the phase velocities $\vec{v}_k$, the phase temperatures T_k and the (common) pressure p: the associated medium specifies the physical and transport properties for each phase (density ρ_k, internal energy e_k, enthalpy h_k, viscosity μ_k, thermal conductivity λ_k) as a function of (p, T_k). For the last, TRUST offers a Thermo-Physical Properties Interface (TPPI) that calls external libraries such as EOS (Neptune project [13] open source library), CoolProp [14], or others.

Physical models prescribe the momentum and heat transfers from phase k to the (k, l) interface ($\vec{F}^i_{kl}$, q^i_{kl}) and to the wall ($\vec{F}^w_k$, q^w_k), while interface jump conditions prescribe the phase change $\Gamma_{kl} = -\Gamma_{lk} = (q^i_{kl} + q^i_{lk})/(h_l - h_k)$ and impose $\vec{F}^i_{lk} = -\vec{F}^i_{kl}$. The system is finally closed by the condition $\sum \alpha_k = 1$.

2.4 The resolution algorithm

Among the many possible forms for these equations [2], the above system uses a semi-conservative form for $\mathcal{Q}$ [15] and the internal energy form of $\mathcal{E}$, which facilitate the implementation of the semi-implicit scheme ICE [16].

The system $(\mathcal{M}, \mathcal{Q}, \mathcal{E})$ is in most cases nonlinear and tightly coupled by its source terms Γ_{kl}, $\vec{F}^i_{kl}$ and q^i_{kl}: ICE offers a way to integrate these terms implicitly while avoiding the solution of a full linear system in $(\alpha, \vec{v}, T, p)$. It consists in the following choice of implicit terms (in red):

$$\begin{cases} \dfrac{\partial \alpha_k \rho_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k \vec{v}_k) = \displaystyle\sum_{l \neq k} \Gamma_{kl} \\ \alpha_k \rho_k \dfrac{\partial \vec{v}_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k \vec{v}_k \otimes \vec{v}_k) - \vec{v}_k \nabla \cdot (\alpha_k \rho_k \vec{v}_k) = \\ \qquad - \alpha_k \nabla p + \nabla \cdot (\alpha_k \mu_k (\nabla \vec{v}_k + {}^t \nabla \vec{v}_k)) + \displaystyle\sum_{l \neq k} \vec{F}^i_{kl} + \vec{F}^w_k \\ \dfrac{\partial \alpha_k \rho_k e_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k e_k \vec{v}_k) = -p \left(\dfrac{\partial \alpha_k}{\partial t} + \nabla \cdot (\alpha_k \vec{v}_k) \right) \\ \qquad + \nabla \cdot (\alpha_k \lambda_k \nabla T_k) + \displaystyle\sum_{l \neq k} (q^i_{kl} + \Gamma_{kl} h_k) + q^w_k \end{cases} \tag{2}$$

Under this form, linearizing each equation E for a given increment $(\delta\alpha, \delta\vec{v}, \delta T, \delta p)$ of the primary variables as $\delta E = \frac{\partial E}{\partial \alpha} \cdot \delta\alpha + \frac{\partial E}{\partial \vec{v}} \cdot \delta\vec{v} + \frac{\partial E}{\partial T} \cdot \delta T + \frac{\partial E}{\partial p} \cdot \delta p$ leads to a linear system of the form

$$\begin{pmatrix} \frac{\partial \mathcal{M}}{\partial \alpha} & \frac{\partial \mathcal{M}}{\partial T} & \frac{\partial \mathcal{M}}{\partial \vec{v}} \\ \frac{\partial \mathcal{E}}{\partial \alpha} & \frac{\partial \mathcal{E}}{\partial T} & \frac{\partial \mathcal{E}}{\partial \vec{v}} \\ 0 & 0 & \frac{\partial \mathcal{Q}}{\partial \vec{v}} \end{pmatrix} \cdot \begin{pmatrix} \delta\alpha \\ \delta T \\ \delta\vec{v} \end{pmatrix} = \begin{pmatrix} \delta\mathcal{M} \\ \delta\mathcal{E} \\ \delta\mathcal{Q} \end{pmatrix} - \begin{pmatrix} \frac{\partial \mathcal{M}}{\partial p} \\ \frac{\partial \mathcal{E}}{\partial p} \\ \frac{\partial \mathcal{Q}}{\partial p} \end{pmatrix} \cdot \delta p \,, \tag{3}$$

where the blue and green blocks are purely *local*: when the equations are discretized, these blocks only contain non-zero terms for lines and columns belonging to the same mesh location. They can thus be inverted as a series of local linear systems: since the overall system is upper-triangular, it can thus be inverted as $\delta\alpha = \delta\alpha^0 + M_\alpha \delta p$, $\delta\vec{v} = \delta\vec{v}^0 + M_{\vec{v}} \delta p$ and $\delta T = \delta T^0 + M_T \delta p$, where the matrices M_α, $M_{\vec{v}}$ and M_T are sparse. Inserting the first relationship into the continuity constraint $\sum \alpha_k = 0$ leads to a system involving only the pressure increments δp: this system can then be solved at each step of a Newton algorithm for the complete nonlinear system.

Compared to a full system over the increments $(\alpha, \vec{v}, T, p)$, this reduced system offers several advantages:

- it is vastly smaller (by a factor of $\sim 8N + 1$ in 3D) and does not increase in size with N;
- its structure is in most cases elliptic (it can be shown to be the sum of a Poisson matrix and a diagonal matrix), and can thus be solved efficiently at large scale and/or in parallel by multigrid methods.

On the other hand, the explicit discretization of convected quantities in (3) means that timesteps are limited by a CFL condition. In order to overcome this limit, additional predictor steps can be implemented to provide initial estimates for convected quantities: this scheme, known as the prediction-correction scheme SETS [17], was implemented as well.

In order to implement the ICE scheme, the underlying architecture must be capable of providing the sparse matrices corresponding to each block of the Jacobian (3). In TRUST, this computation is spread between the space discretization, equation and medium classes in order to maximize code reuse. For instance, the derivative of the convective term of the mass equation $\mathcal{M}_k$ w.r.t. the phase temperature T_k is computed as:

$$\frac{\partial \nabla \cdot (\alpha_k \rho_k \vec{v}_k)}{\partial T_k} = \frac{\partial \nabla \cdot (\alpha_k \rho_k \vec{v}_k)}{\partial \alpha_k \rho_k} \cdot \frac{\partial \alpha_k \rho_k}{\partial \rho_k} \cdot \frac{\partial \rho_k}{\partial T_k} \tag{4}$$

where:

- the red term is computed by the convection operator of the underlying space discretization, such as PolyMAC or VDF. This operator implements a term of the form $\nabla \cdot (F\vec{v})$ for a

convected field F: the instance of this term associated with $\mathcal{M}_k$ operates with $F = \alpha_k\rho_k$, while another instance associated to $\mathcal{E}_k$ operates with $F = \alpha_k\rho_k e_k$. Both equations rely on the same code to compute the matrix $\partial\nabla\cdot(F\vec{v})/\partial F$;

- the green term $\partial\alpha_k\rho_k/\partial\rho_k$ is local, and computed by the equation $\mathcal{M}_k$ where the convected field $F = \alpha_k\rho_k$ is defined. This field relies on the fields α_k (a primary unknown) and ρ_k (the density, provided by the medium);
- finally, the blue term is computed by the medium class, which defines the density field $\rho_k(p, T_k)$.

This architecture has been designed to maximize code reuse, in particular at the numerical scheme level. It has been found to be flexible and efficient.

The $3N$-equation system $(\mathcal{M}, \mathcal{Q}, \mathcal{E})$ presents particular difficulties in the case of a vanishing phase ($\alpha_k \to 0$). In Pb_Multiphase, a limiter is placed on the total phase change $\Gamma_k = \sum_l \Gamma_{kl}$ to deal with this issue. For a time step Δt, the time-discretized form of the mass equation $\mathcal{M}_k$ reads

$$\frac{\alpha_k^+\rho_k^+ - \alpha_k^-\rho_k^-}{\Delta t} + \nabla\cdot(\alpha_k\rho_k\vec{v}_k) = \Gamma_k \tag{5}$$

with the -/+ superscripts denoting values at times t and $t + \Delta t$: hence, the condition $\alpha_k^+ \geq 0$ leads to

$$\Gamma_k \geq \Gamma_k^{lim} = \nabla\cdot(\alpha_k\rho_k\vec{v}_k) - \frac{\alpha_k^-\rho_k^-}{\Delta t}\,. \tag{6}$$

At a given iteration, if $\Gamma_k < \Gamma_k^{lim}$, the mass flux Γ_{kl} to the phase l with the highest α_l (the *dominant* phase) is modified so that $\Gamma_k = \Gamma_k^{lim}$: then, the heat flux q_{lk}^i is modified in order to preserve the jump condition $\Gamma_{kl} = (q_{kl}^i + q_{lk}^i)/(h_l - h_k)$. This process ensures that the converged solution of the Newton algorithm will satisfy $\alpha_k \geq 0$ while maintaining consistent mass, energy and momentum balances between the phases. At the end of each Newton iteration, the field α_k must be post-processed to ensure $\alpha_k = 0$ while preserving $\sum\alpha_k = 1$. Final convergence is only declared once the algorithm converges to a solution satisfying $\alpha_k \geq 0$.

Additionally, the momentum and energy equations must ensure that $\vec{v}_k \to \vec{v}_l$ and $T_k \to T_{sat}$ when $\alpha_k \to 0$. In `Pb_Multiphase`, this is obtained by ensuring that the interfacial exchange terms $\vec{F}_{kl}^i$ and q_{kl}^i do not cancel out as $\alpha_k \to 0$. For cases where α_k becomes close to zero ($\approx 10^{-6}$), a vanishing operator is used to manage the situation [11].

2.5 Turbulence modeling

A $k-\omega$ or a $k-\tau$ model is used for shear-induced turbulence [18, 19], with a constant turbulent Prandtl number fixed at 0.9 for the energy equation.

We implement an adaptive wall function that determines the friction velocity then calculates the shear stress $\vec{F}_k^w$ [20]. The boundary condition on k is $k = 0$ at the wall for $y_+ < 10$, where $y_+ = yu_\tau/\nu_l$ and y is the distance between the wall and the first element center. For larger wall elements, it is a zero-flux condition. The transition is smoothed by a transition factor $\tanh\left((y_+/10)^2\right)$. For ω, Knopp et al. [21] give an analytical value in the near-wall region. A simple solution would be to enforce this value in the first element. However, it was already used in the single-phase version of TrioCFD and creates numerical issues for tetrahedron meshes. Instead, we calculate the analytical solution at a distance $y/2$ from the wall. We then enforce a Dirichlet boundary condition at the wall for the first element: $\omega_{\text{wall}} = 2\cdot\omega(y/2)$. Finally, the single-phase heat transfer coefficient implemented was proposed by Kader [22].

2.6 Multi-phase closure terms

Virtually all multi-phase closure terms require a bubble diameter. In multiphase TrioCFD, one can impose a constant diameter, a 3D time-dependent field, use an interfacial area transport equation [23] or a MUSIG population balance model [24].

A detailed description of the interfacial forces available in TrioCFD can be found in [5]. The interfacial force exerted by the liquid on the gas is $\vec{F}^i_{lg} = -\vec{F}^i_{gl}$. All forces written here apply to the gas phase. We separate the interfacial force term in five different contributions:

$$\vec{F}^i_{lg} = \vec{F}_{\text{drag}} + \vec{F}_{\text{VM}} + \vec{F}_{\text{lift}} + \vec{F}_{\text{TD}} + \vec{F}_{\text{wall}} \tag{7}$$

A constant-coefficient drag force, the formulations of Ishii & Zuber [25] or that of Tomiyama et al. [26] can be used. The virtual mass force uses a constant coefficient [27]. The lift forces coded include a constant-coefficient lift force, the Tomiyama force [28] and the Sugrue formulation [29]. The Burns et al. turbulent dispersion force [30] and the wall correction of Antal et al. [31] and Lubchenko et al. [32] are also implemented.

For the energy and mass equations, the interfacial heat transfer can be chosen with a constant Nusselt number, or using the Ranz & Marshall [33] or Chen & Mayinger [34] formulations. This allows to compute $(q^i_{kl} + \Gamma_{kl} h_k)$. The Kurul & Podowski heat flux partition [35] was implemented to compute q^w_k.

The flexibility of TrioCFD multiphase solver means that other formulations of these closure terms can easily be added by creating a new class, and be used in any numerical scheme and with any resolution method.

3 Verification

3.1 Single-phase turbulence

Single-phase turbulence has been verified on three canonical geometries: a channel flow, a circular pipe and a backward facing step. For the latter, the database provided by the Turbulence Modeling Group was used [36]. The validation protocol was design to compare every model available[1] in TrioCFD with every compatible discretization. This is summarized in Table 1. The turbulence models in the multi-phase framework are compared with the validated historical turbulence in the single-phase framework. For the sake of brevity, we only describe and report the validation on the channel flow and the backward step.

Discretization	$k-\varepsilon$	$k-\omega$	$k-\tau$
Historic VEF	✓		
Historic VDF	✓		
Multiphase TrioCFD VDF		✓	✓
Multiphase TrioCFD PolyMAC		✓	✓

Table 1: Tested models and their associated discretizations for single-phase turbulence validation.

The channel flow is a rectangular channel of length 100 meters and of 2 meter height. The fluid density is 1000 kg/m^3 and its dynamic viscosity is 0.02 Pa.s. A symmetric condition is applied at half the channel size. The inlet velocity is 1 m/s. The length of the channel was

[1] After this validation process, the $k-\tau$ model was abandoned to concentrate the effort on the $k-\omega$ model.

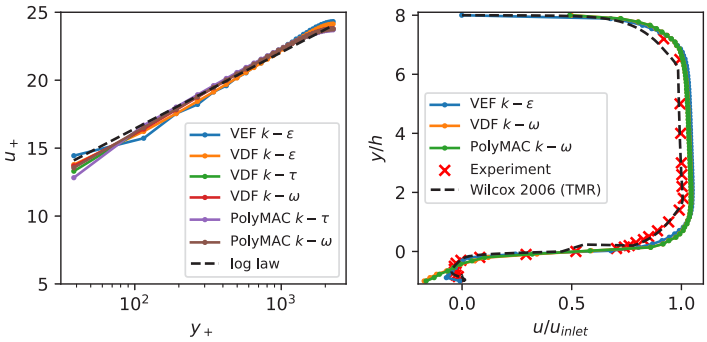

Figure 2: Left: velocity profile u^+ in wall unit y^+ for all the models for the channel flow. Right: Velocity profile (abscissa) in normalized vertical unit y/h at an axial distance h for the backward step.

chosen to let the boundary layer develop itself on 25 hydraulic diameter. The main result is summarized in Figure 2 (left). The velocity profile u^+ is plotted against the wall unit y^+. All models give the same profile close to the theoretical log law. An evaluation of the head loss was also conducted. It gives a relative difference of maximum 10 % compared to a common Reynolds-based analytical evaluation of the friction factor.

The backward step is the test case described by [37]. The domain length is divided in two sections. The first one is 1.1133 meter long and 8 meters height. The second section has an abrupt increase of height of $h = 0.0127$ meter and is 0.5 meter long. The inlet velocity is 44.2 m/s. The fluid density is 1 kg/m^3 and its dynamic viscosity is 1.469×10^{-5} Pa.s. The velocity profile is plotted against the normalized height y/h at an axial distance $x = h$ on Figure 2 (right). Results are in a very good agreement with the experimental reference values (red crosses) and the numerical reference from Wilcox.

One of the criteria of validation is the capture of the recirculation length after the step. The relative gap with the experiment value is lower than 20 % for the different models and discretizations. This value is in compliance with other CFD codes for this level of discretizations.

3.2 Adiabatic multi-phase flow

To verify the implementation of the multi-phase force terms, we verify the analytic solution found by Marfaing et al. for a simplified set of closures [38]. They find that, using constant coefficient drag, lift and turbulent dispersion forces and the Antal wall correction, the radial void fraction in a developed pipe and channel can be expressed analytically.

We run simulations in 3 different geometrical configurations for both schemes using the SETS solver and meshes with 40 radial and 200 axial elements. In all configurations, a gas-liquid mixture enters at 0.53 m/s and 4.2% void fraction at the bottom of the domain and we use the same coefficients as Marfaing et al. The domain is 1.5 m-long and 0.019 m-wide. A fixed-pressure boundary condition is enforced at the top. The first configuration is a 2D channel. The second is a 3D channel, with symmetry boundary conditions in the additional direction. This should behave exactly like a 2D channel. The third is a round pipe. It is simulated in VDF using the axisymetric module, and with PolyMAC with a 1-cell wide slice with symmetry boundary conditions in the core. The results are shown in figure 3. Both

schemes agree with the analytical solution in a channel and in a pipe, which validates the implementation of interfacial forces.

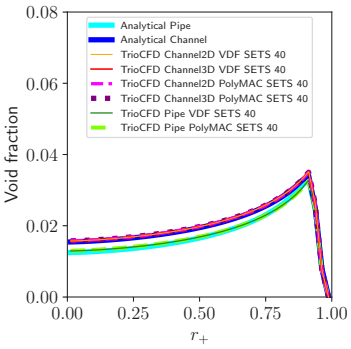

Figure 3: Simulation results for a simplified set of closures in an adiabatic flow vs analytical solution. All numerical schemes behave in the expected way.

3.3 Boiling multi-phase flow

To the best of our knowledge, no steady-state analytic solution exists in the literature for boiling flow. To verify our implementation, as source terms are coded separately, we can compare simulations that use VDF and PolyMAC.

We simulate 2 physical configurations with the ICE solver, both schemes, and meshes with 20 radial and 200 axial elements. The domain is 0.0096 m-wide and 5 m-long with a boiling length of 3.5 m. Physical quantities are extracted near the outlet. The fluid used is freon-R134A, with 14.6 bar outlet pressure, 2723 kg/(m^{2}s) mass velocity and 44.5°C inlet temperature. The deformable Ishii-Zuber drag, Burns turbulent dispersion, and constant coefficient $C_l = -0.03$ lift forces are used. The heat flux partition is the Kurul & Podowski one, and the condensation is taken with a constant Nusselt number at 15 and a bubble diameter equal to the capillary length $\sqrt{\sigma/(\Delta\rho g)}$. The first configuration is a pipe where the imposed flux is 81.4 kW/m^2. It is simulated with the 2D axi-symmetric module in VDF. In PolyMAC, the mesh is a 3D. A 2° 1-element wide slice with symmetry boundary conditions is used. Both meshes contain 20x200 elements. The second is a 2D channel, simulated with an identical 20x200 Cartesian grid in both schemes, where the imposed flux is 120.0 kW/m^2.

The results are shown in figure 4. Both schemes behave in the same way in channels and in pipes. This validates the implementation of boiling-specific terms: heat flux partition, condensation and their interplay with interfacial forces.

4 Towards HPC

Due to a lack of space in this paper, we cannot present HPC results here. However, work is ongoing for the following configurations and will be presented at SNA+MC:

- Flow in boiling sloped heated channels
- Simulation of the void drift between two boiling subchannels with different heating powers
- Coupled simulations on reactor geometries to run the lower and upper plenums with multiphase TrioCFD and the reactor core with a subchannel code

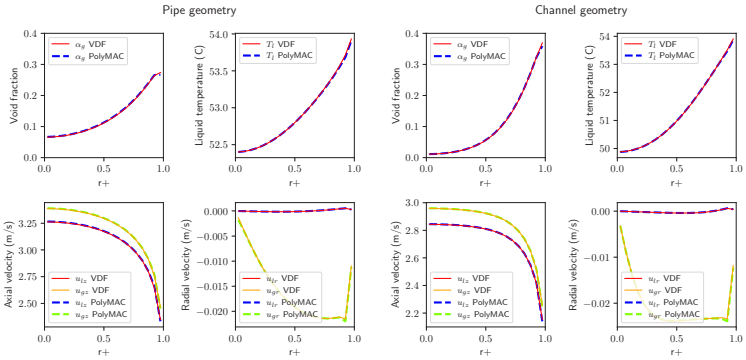

Figure 4: Comparison of VDF and PolyMAC simulation results for boiling multi-phase flow. Both schemes behave in the same way.

5 Conclusion

TrioCFD is a verified and scalable code for multi-phase applications. It can already be used to simulate complex systems. Future work includes continuing to improve the numerical scheme and the resolution method, studying the effect of bubble-induced turbulence and coupling TrioCFD with neutronics and system-scale codes.

References

[1] N.E. Todreas, M.S. Kazimi, *Nuclear Systems I: Thermal Hydraulic Fundamentals* (CRC press, 2021)
[2] M. Ishii, T. Hibiki, *Thermo-fluid dynamics of two-phase flow* (Springer, 2011)
[3] P. Emonot, A. Souyri, J. Gandrille, F. Barré, Nuclear Engineering and Design (2011)
[4] R.T. Lahey, E. Baglietto, I.A. Bolotnov, Nuclear Engineering and Design (2021)
[5] C. Reiss, A. Gerschenfeld, C. Colin, *Building a boiling-flow multiphase CFD framework for interfacial area and heat transfer modeling*, in *NURETH20-20th International Topical Meeting on Nuclear Reactor Thermal Hydraulics.* (2023)
[6] C. Reiss, A. Gerschenfeld, C. Colin, submitted to Nuclear Engineering and Design
[7] P.E. Angeli, U. Bieder, G. Fauchet, *Overview of the TrioCFD code: Main features, V&V procedures and typical application to nuclear engineering*, in *NURETH-16* (2015)
[8] E. Saikali, P. Ledac, A. Bruneton, A. Khizar, C. Bourcier, G. Bernard-Michel, E. Adam, D. Houssin-Agbomson, *Numerical modeling of a moderate hydrogen leakage in a typical two-vented fuel cell configuration*, in *International Conference of Hydrogen Safety* (2021)
[9] E. Saikali, G. Bernard-Michel, A. Sergent, C. Tenaud, R. Salem, international journal of hydrogen energy **44**, 8856 (2019)
[10] F. Harlow, J. Welch, Physics of Fluids **8**, 2182 (1965)

[11] A. Gerschenfeld, Y. Gorsse, *Development of a Robust multiphase flow solver on General Meshes; application to sodium boiling at the subchannel scale*, in *NURETH-19* (2022)
[12] P.L. Bacq, A. Gerschenfeld, M. Ndjinga, *PolyMAC: Staggered Finite Volume Methods on General Meshes for Incompressible Navier–Stokes Problems*, in *International Conference on Finite Volumes for Complex Applications* (Springer, 2023), pp. 149–156
[13] A. Guelfi, D. Bestion, M. Boucker, P. Boudier, P. Fillion, M. Grandotto, J.M. Hérard, E. Hervieu, P. Péturaud, Nuclear Science and Engineering **156**, 281 (2007)
[14] I.H. Bell, J. Wronski, S. Quoilin, V. Lemort, Industrial & engineering chemistry research **53**, 2498 (2014)
[15] I. Park, H. Cho, H. Yoon, J. Jeong, Nuclear Engineering and Design **239**, 2365 (2009)
[16] F.H. Harlow, A.A. Amsden, Journal of Computational Physics **3**, 80 (1968)
[17] J.H. Mahaffy, Journal of Computational Physics **46**, 329 (1982)
[18] J. Kok, Tech. Rep. NLR-TP-99295, National Aerospace Laboratory NLR (1999)
[19] J. Kok, S. Spekreijse, Tech. Rep. NLR-TP-2000-144, National Aerospace Laboratory NLR (2000)
[20] J.R. Carlson, V.N. Vatsay, J. Whitey, *Node-Centered Wall Function Models for the Unstructured Flow Code Fun3D*, in *22nd AIAA Computational Fluid Dynamics Conference* (2015), p. 2758
[21] T. Knopp, T. Alrutz, D. Schwamborn, Journal of Computational Physics **220**, 19 (2006)
[22] B. Kader, Int. J. Heat Mass Transfer **24**, 1541 (1981)
[23] P. Ruyer, N. Seiler, La Houille blanche (2009)
[24] E. Krepper, D. Lucas, T. Frank, H.M. Prasser, P.J. Zwart, Nuclear Engineering and Design **238**, 1690 (2008)
[25] M. Ishii, N. Zuber, AlChE Journal **25** (1979)
[26] A. Tomiyama, I. Kataoka, I. Zun, T. Sakaguchi, JSME International Journal Series B Fluids and Thermal Engineering (1998)
[27] N. Zuber, Chemical Engineering Science **19**, 897 (1964)
[28] A. Tomiyama, H. Tamai, I. Zun, S. Hosokawa, Chemical Engineering Science **57**, 1849 (2002)
[29] R. Sugrue, Ph.D. thesis, Massachusetts Institute of Technology (2017)
[30] A.D. Burns, T. Frank, I. Hamill, , J.M. Shi, *The Favre Averaged Drag Model for Turbulent Dispersion in Eulerian Multi-Phase Flows*, in *5th International Conference on Multiphase Flow* (2004)
[31] S.P. Antal, R.T. Lahey, J.E. Flaherty, Int. J. Multtphase Flow **17**, 635 (1991)
[32] N. Lubchenko, B. Magolan, R. Sugrue, E. Baglietto, International Journal of Multiphase Flow **98**, 36 (2018)
[33] W.E. Ranz, W. Marshall, Chem. Eng. Prog. **48**, 141 (1952)
[34] Y.M. Chen, F. Mayinger, Int. J. Muhiphase Flow **18**, 877 (1992)
[35] N. Kurul, M. Podowski, *Multidimensional effects in forced convection subcooled boiling*, in *International Heat Transfer Conference Digital Library*, edited by B.H. Inc. (1990)
[36] C. Rumsey, B. Smith, G. Huang, *Description of a Website Resource for Turbulence Modeling Verification and Validation*, in *40th Fluid Dynamics Conference and Exhibit* (2010), https://doi.org/10.2514/6.2010-4742
[37] D.M. Driver, H.L. Seegmiller, AIAA Journal **23**, 163 (1985)
[38] O. Marfaing, M. Guingo, J. Laviéville, G. Bois, N. Méchitoua, N. Mérigoux, S. Mimouni, Chemical Engineering Science **152**, 579 (2016)