A zonal grid method for incompressible two-phase flows

F. Dabonneville, N. Hecht, Julien Reveillon, G. Pinon, F. X Demoulin

A zonal grid method for incompressible two-phase flows

F. Dabonneville[1], N. Hecht[2], J.Reveillon[1*], G. Pinon[2], F.X. Demoulin[1]

[1] *CORIA-UMR6614 CNRS, Normandy University, University of Rouen*
Avenue de l'Universite, 76800 Saint Etienne du Rouvray, France
[*] *Corresponding author: Julien.Reveillon@coria.fr*
[2] *LOMC-UMR6294, Normandy University, University of Le Havre*
75, rue Bellot, 76600 Le Havre, France

Abstract

We present a zonal grid based numerical method applicable to two-phase flows. The method is aimed at reducing the computational costs for interfacial flows involving local high velocity gradients such as those encountered in atomization systems. The objective is to fully resolve the primary atomization region while at the same time limiting the number of grid points in secondary or dispersed zones where a very fine grid is not required, along with the ability to use a local zone based time step. Calculations of a rising bubble and liquid jet atomization configurations were performed on a coarse grid with a fine zone superimposed on small domains with the strongest gradient. The results show good agreement with simulations carried out with a complete refined mesh with only a fraction of the CPU time. The results indicate that the two-phase zonal grid model approach introduced here has the potential to provide an accurate and cost-effective approach for modeling two-phase flow problems that have multiple temporal and spatial scales.

Keywords: zonal grid, multi-phase, multi-solver, interface

1. Introduction

Two-phase flows are frequently encountered in nature or industrial devices such as, for example, in wave breaking, fluidized beds, chemical reactors, phase change heat exchangers, etc. These flows are characterized by the presence of an interface between two or more phases and one of their common characteristics is the presence of a wide range of temporal and spatial characteristic scales. The atomization of a liquid jet is a perfect example to illustrate these scales. In such a configuration, the destabilization of the gas-liquid interface occurs on a large length scale compared to the size of ligaments and droplets that follows the break-up of the interface, the thickness of which is smaller than the Kolmogorov length scale. Despite significant progress during the past decade, the modeling and simulation of two-phase flows with moving interfaces and a wide range of characteristic scales in complex geometries remains a challenging problem from both the physical and the numerical point of view.

Among several other imperatives, a fine mesh resolution is crucial to obtain an accurate estimation of the temporal and spatial evolution of the interfacial flow. However, computational resources are limited. Consequently, the use of a uniform fine mesh covering the whole geometry is far too costly in practical or industrial configurations. To overcome this difficulty, it is possible to refine the mesh only when necessary at very specific locations. Grid refinement is the adaptation of a computational grid to a flow solution by locally dividing the grid cells into smaller cells. It is an efficient way to address flow scaling problems.

Another difficulty is the diversity of the physical phenomena involved in such flows. A unique set of models to capture the flow evolution is not necessarily an optimal choice since the capture of the interface with large scale destabilization on the one hand and the computation of a vaporizing dispersed spray on the other hand need very different physical and numerical approaches to be correctly predicted by any simulation.

This is why the local grid refinement is only part of the solution. One should also be able to solve locally different physical and numerical models when necessary. Let's have a general overview of the existing techniques able to address these major points.

Grid refinement techniques may be broadly divided into two groups: Adaptive Mesh Refinement (AMR) and Domain Decomposition Methods (DDM).

Adaptive mesh refinement methods dynamically refine the mesh to capture sharp discontinuities and steep gradients according to the flow dynamics. A mesh broadening operation is also applied in areas with weak variations of the flow properties. Apart from minimizing the number of grid cells, which reduces the computational cost, a major advantage is the automation of the procedure, which eases the meshing work.

Among existing families of AMR, the most common and basic ones consist of locally refining a single mesh by dividing the grid cells considered. The main drawback of the procedure is the automatic division of the time step. This time step dependency comes from the reduced cell size in locally refined region. Thus, even if the number of cells is optimized, it leads anyway to a high computational cost. To gain computer resources, refined regions can be considered as independent blocks with a specific time step that does not impact the global time step. This kind of AMR is called AMR with subcycling in time. Refinement is performed in time as well as in space so that the ratio of the time step to the grid spacing is kept constant. This approach is based on the original method proposed by Berger and Oliger [1]. A series of studies have investigated this topic: Almgren *et al.* [2], Bell *et al.* [3], Martin *et al.* [4, 5] treated several important aspects of the Navier-Stokes solution, in particular the conservative aspects. This AMR methodology with refinement in time has been implemented in the AMR library AMReX [6] and can be used for various applications. The term *adaptive* means that refined regions are constructed or destroyed every few time-steps, based on given error criteria. This operation is called *book keeping* and it has an additional cost, especially in the framework of parallel computing, where intergrid connectivity has to be updated after each *book keeping* operation. Furthermore, *book keeping* is unnecessary in configurations where the size and location of zones needing refinement are a priori known before the beginning of the simulation. In this case, a static local mesh refinement technique is a better candidate as one of the domain decomposition methods.

The Domain Decomposition Method (DDM), also called multizonal approach, is a favorable candidate to handle complex flows and geometries. In DDM, the whole domain is divided into a number of geometrically simple zones or subdomains in which independent meshes can be generated. Different governing equations or numerical solvers can be applied on these meshes and a zone can be easily refined without affecting its neighbors. Each subdomain can be connected to a neighboring zone with a simple patch (patched grids) or they can share a common region (overlapping grids). A third possibility is to define a grid that can fully overlap a larger region (zonal grid). These connection methods are now detailed.

The patched grids method employs disjoint subdomains that share a simple interface or patch without overlapping. It was proposed by Rai [7] for the finite difference method and extended to the finite volume method by Walters *et al.* [8]. One of the main advantages of neighboring blocks that have neat interfaces is their ability to have a better conservative description than, for instance, in the overlapping grids method. Many studies have exploited this conservative property [9, 10] and improved it [11, 12]. On the other hand, this method is less efficient at handling complex geometries because of the interface constraint.

The overlapping grids method, also called the Chimera method, was introduced by Benek *et al.* [13, 14] in 1983. This method is used to perform simulations involving multiple bodies in relative motion and multiple grids that overlap. Its development was also motivated by the possibility of selectively refining the mesh in regions of interest and solving different flow models on each mesh. For example, it is possible to solve Navier-Stokes (NS) equations in near-wall areas and Euler equations in zones far from the walls. This NS/Euler hybrid approach has since been applied in many studies [15, 16, 17]. In the context of multiphase flows, an overlapping grids methods has been developed by Tu [18] to model complex turbulent two-phase flows with irregular geometries. It was recently applied by Nguyen *et al.* [19] to moving bodies to enter water or by Castro *et al.* [20] and Wan *et al.* [21] about to ship hydrodynamic problems.

In some specific cases, a given zone can fully overlap a larger region. This is called the zonal grid method. It uses a successive refinement of grids where a coarse grid solution constitutes the boundary conditions for the next finer grid level. It is similar to patch-structured AMR, with the difference that grids are static. It was used to improve the resolution of near wall boundary layers in [22] and [23] using respectively Large Eddy Simulation (LES) and Direct Numerical Simulation (DNS). Hybrid schemes with various turbulence models can also be used, as in the work of Richez *et al.* [24] where a Reynolds averaged Navier Stokes (RANS) turbulence model is used in the main domain while LES describes the walls' boundary layer. It has also been widely used in meteorological applications [25, 26, 27] to refine the mesh inside high velocity gradient structures such as hurricanes.

The objective of the present work is to extend the zonal grid approach in order to simulate two-phase flows, with refinement in space and time. This new TPZG (Two-Phase Zonal Grid) approach aims to combine most of the positive features of some of the previously described methods:

1/ Use the sub-cycling in time of the AMR to reduce the computational cost of the fine mesh zones.

2/ Adopt the coincident interfaces method from the patched grids approach for a better conservation of the flow properties between neighboring zones.

3/ When necessary, use the possibility of DDM to solve various flow models with adapted numerical schemes in each zone.

4/ Use the original zonal grid topology feature that keeps one global mesh for the overall computational domain adding when and where necessary embedded refined subgrids.

In the TPZG method detailed in this work, a coupling between one coarse domain and a refined region is considered. An extension to multiple refined regions is straightforward if necessary. This coupling is two-way and explicit. It means that matrices dedicated to each grid are defined and solved separately and then followed by an explicit coupling, which is the only way to obtain subcycling in time between domains. The coarser domain is solved first and provides boundary conditions for the refined region that is then solved with, possibly, a smaller time-step until it reaches the same time state as the coarse region. The fine grid solution is finally interpolated onto the coarse grid and additional corrections are performed.

In the following section, two different multiphase solvers are described in detail. The first one uses an Interface Capturing Method (ICM) to follow the flow evolution while the second one does not define the interface but uses a Turbulent Diffusion Flux (TDF). The two-phase zonal grid method is then described. It uses these two solvers as a basis to demonstrate the ability of the procedure to capture different kinds of flow. In the last section of this paper, results are shown where TPZG is applied on three different test cases: a rising bubble and two liquid jet atomization configurations.

2. Governing equations

The complex nature of two-phase flows, characterized by turbulence, deformable phase interface, phase slip and compressibility of the gas phase, makes it difficult to obtain reliable flow models. First of all, the nature of the flow needs to be characterized. A two-phase flow can be broadly classified into three categories: separated flows, mixed flows and dispersed flows. Generally, the description of each of these categories requires a dedicated approach.

In separated flows, each phase is continuous and occupies a distinct region of the domain; the volume fraction of the primary phase is high in one region and low in the other one and the two phases are separated by an interface, where surface tension force applies. In dispersed flows, one phase is assumed to be dilute (with a volume fraction smaller than 10% of the global volume) and composed of finite spherical inclusions dispersed inside the other carrier phase. All dispersed elements are assumed to be very small compared to the scale of the system. Thus, they are considered pointwise and the flow around and inside the particles does not need to be computed. Mixed flows are transitional states between the two above-mentioned flow regimes.

Several families of solvers exist to describe the evolution of two-phase flows at a macroscopic level. When modeling the evolution of two-phase flows, Euler-Lagrange descriptions are often considered. However, they assume a very specific topology (dispersed spherical droplets) and the Lagrangian tracking of numerous inclusions engulfed in a carrying phase. The carrier phase is solved by an Eulerian description which can lead to several numerical difficulties. On the other hand, Euler-Euler methods can deal with both dense and dispersed flows but with additional closure terms to account for the dynamics of the two phases. Lastly, Eulerian methods consider a single-fluid approach, thus avoiding the choice of both carrier and discrete phases [28]. In this study, the two-phase flow is considered as a single-fluid flow composed of two species with highly variable density. To illustrate the ability of the TPZG, two different Eulerian solvers have been coupled together. These solvers use a Volume-Of-Fluid (VOF) [29] surface tracking method. VOF defines and tracks in time and space the portion of primary phase present in a given cell. With a VOF description, the interface can be treated either with a resolved interface process that defines a sharp boundary where surface tension applies or as a diffuse transition between each phase. As these solvers are well-known, their algorithms are reported in Appendix B, along with the discretized equations in Appendix A. In both solvers, evolution of the primary phase is described by a balance equation of a phase indicator, α. The first solver uses a VOF method to track the interface with a specific algorithm surface compression method [30] to limit numerical diffusion. It will be referred to as the Resolved Interface (RI) solver in the following. As previously mentioned, the second solver does not consider any surface tension and since turbulent flows are considered, it can be shown that the turbulent diffusion flux term prevails in the primary phase volume fraction evolution. Thus, this solver is referred to as the Turbulent Diffusion Flux (TDF) solver.

Let's consider a two-fluid flow with a high density fluid denoted by 1 and a low density fluid by 2. Both fluids are incompressible and non miscible. In the present work, phase 1 is taken as the reference fluid component followed in space and time. The scalar field α represents the volume fraction of fluid 1. Thus,

$$\alpha = \begin{cases} 1 & \text{in fluid 1} \\ 0 & \text{in fluid 2} \\ 0 < \alpha < 1 & \text{at the interface} \end{cases} \tag{1}$$

Considering incompressible flows, without mass transfer across the interfaces, the governing equations include continuity:

$$\boldsymbol{\nabla} \cdot \boldsymbol{U} = 0 \quad , \tag{2}$$

where $\boldsymbol{U}$ is the velocity vector. The momentum balance is given by:

$$\frac{\partial \rho \boldsymbol{U}}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \boldsymbol{U}\boldsymbol{U}) - \boldsymbol{\nabla} \cdot (\mu_{eff} \boldsymbol{\nabla} \boldsymbol{U}) = -\boldsymbol{\nabla} P_{rgh} + \boldsymbol{\nabla} \boldsymbol{U} \cdot \boldsymbol{\nabla} \mu_{eff} + \boldsymbol{Q} \quad , \tag{3}$$

where ρ is the density and P_{rgh} is the dynamic pressure defined as:

$$P_{rgh} = P - \rho \boldsymbol{g} \cdot \boldsymbol{x} \quad , \tag{4}$$

where P is the pressure, $\boldsymbol{g}$ the gravitational acceleration and $\boldsymbol{x}$ the position vector. The efficient dynamic viscosity term $\mu_{eff} = \mu + \rho\nu_t$ is the sum of the molecular dynamic viscosity μ and the turbulent effects $\rho\nu_t$. The turbulent kinematic viscosity ν_t is given by the chosen turbulence model. The source term $\boldsymbol{Q}$ includes the gravitational acceleration $\boldsymbol{g}$. Moreover, when a resolved interface between the two fluids is considered, the surface tension is added to $\boldsymbol{Q}$. Then,

$$\boldsymbol{Q} = \begin{cases} \boldsymbol{Q}_D & = -\boldsymbol{g} \cdot \boldsymbol{x} \boldsymbol{\nabla} \rho \quad , \\ \boldsymbol{Q}_I & = -\boldsymbol{g} \cdot \boldsymbol{x} \boldsymbol{\nabla} \rho + \boldsymbol{f}_\sigma \quad , \end{cases} \tag{5}$$

where the subscript $()_D$ is specific to a turbulent diffusion flux solver (TDF) and $()_I$ stands for a resolved interface (RI) modeling. In the latter case, the surface tension $\boldsymbol{f}_\sigma$ is applied at the interface position. This force is defined by:

$$\boldsymbol{f}_\sigma = \sigma \kappa \boldsymbol{\nabla} \alpha_I \quad , \tag{6}$$

where σ is the surface tension coefficient. The interface curvature κ is defined as

$$\kappa = -\boldsymbol{\nabla} \cdot \boldsymbol{n} = -\boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla} \alpha_I}{|\boldsymbol{\nabla} \alpha_I|} \right) \quad , \tag{7}$$

the divergence of the normal vector $\boldsymbol{n}$ of the interface.

The fluid density and viscosity are functions of the phase indicator α:

$$\rho = \alpha \rho_1 + (1 - \alpha) \rho_2 \quad , \tag{8}$$

and

$$\mu = \alpha \mu_1 + (1 - \alpha) \mu_2 \quad , \tag{9}$$

where density and viscosity in each separate phase 1 and 2 are considered constant.

Besides the mass and momentum equations, the VOF method requires keeping track of the volume fraction. This is performed by advecting the VOF field with the incompressible velocity field through the following equation:

$$\frac{\partial \alpha}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{U} \alpha) = 0 \quad . \tag{10}$$

Equation (10) transports the mixture properties and position of the interface. In the case of a sharp interface, an interface capturing method is necessary to define the geometric characteristics of the interface and correctly compute the surface tension term in the momentum equation (3).

Resolved interface (RI solver).

In the RI solver, equation (10) is solved with an additional surface compressive term in order to keep the interface sharp [31]:

$$\frac{\partial \alpha_I}{\partial t} + \boldsymbol{\nabla} \cdot (\boldsymbol{U} \alpha_I) + \underbrace{\boldsymbol{\nabla} \cdot [\boldsymbol{U}_c \alpha_I (1 - \alpha_I)]}_{\text{compressive term}} = 0, \tag{11}$$

where the $\alpha_I(1-\alpha_I)$ term enforces the compressibility term to be concentrated only at the interface region. Thus, it has no effect on the solution throughout the rest of the domain. $\boldsymbol{U}_c$ is a suitable velocity field selected to compress the interfacial region [32], defined as a relative velocity between the two phases:

$$\boldsymbol{U}_c = \min\left[C_\alpha \mid \boldsymbol{U} \mid, \max\left(\mid \boldsymbol{U} \mid\right)\right] \boldsymbol{n} \tag{12}$$

where the min operator is performed locally, at the faces surrounding the cells concerned and the max operator is performed globally, over the entire domain. The order of magnitude of the compressibility coefficient C_α is unitary and in this work, it was set to unity. As described by Rusche [31], the artificial compressive term of equation 11 provides interface straightening without the need to use an interface reconstruction method, with the advantage that the boundedness of α between 0 and 1 is respected.

Diffuse interface (TDF solver).

The TDF solver exploits a Reynolds averaged version of the balance equation for the primary phase volume fraction (Eq.10) with an additional term namely the *turbulent diffusion liquid term*:

$$\frac{\partial \overline{\alpha}_D}{\partial t} + \boldsymbol{\nabla} \cdot (\overline{\boldsymbol{U}} \overline{\alpha}_D) = \underbrace{-\boldsymbol{\nabla} \cdot \boldsymbol{R}_\alpha}_{\text{turb. diff. liquid term}} \quad . \tag{13}$$

This turbulent liquid flux defined as $\boldsymbol{R}_\alpha = \overline{\boldsymbol{U} \alpha_D} - \overline{\boldsymbol{U}} \overline{\alpha}_D$ represents the turbulent transport of the liquid volume fraction induced by velocity fluctuations. It is modeled thanks to a gradient-diffusion closure approximation

$$\boldsymbol{R}_\alpha = -\frac{\nu_t}{Sc_t} \boldsymbol{\nabla} \overline{\alpha}_D \quad , \tag{14}$$

where Sc_t is the turbulent Schmidt number set to 0.7.

3. Numerical method

Both two-phase solvers introduced previously are generally used to solve the evolution of two-phase flows on a single grid. Our objective is to use simultaneously both solvers and interlink them on two different grids, thanks to a zonal grid methodology, with separate refinement in time and space.

Depending on the local characteristics of the flow, it is then possible to use a particularly refined zonal grid making it possible to capture very high local gradients while a coarse grid is used to describe the rest of the flow.

In this section, TPZG is developed in the framework of the OpenFOAM® library. The methodology may be applied for any solver. In this work, the interface *(RI)* tracking solver used is `interFoam` while the diffusive *(TDF)* solver is `twoLiquidMixingFoam`. In both solvers, the distribution of the liquid volume fraction field α is calculated ahead of the PISO algorithm [33] that updates the velocity $\boldsymbol{U}$ and the pressure P_{rgh} fields. The two solver algorithms are described together in Appendix B.

3.1. Notations

In the following, the considered domains are referred to as the *Global Domain* (GD) and the *Zonal Domain* (ZD). GD is the complete geometry to be studied while ZD is a local area fully superimposed on part of GD. The ZD mesh is finer than the GD mesh, since its purpose is to capture physical processes finer than the resolution of GD (Fig. 1). The interface between ZD and GD is referred to as Γ.

Concerning finite volume notations, in the following, subscript $()_f$ indicates the face values. The cell volume is V and the volumetric flow rate through one of its faces f is defined as

$$F = \boldsymbol{S}_f \cdot \boldsymbol{U}_f, \tag{15}$$

where $\boldsymbol{S}_f$ is the surface normal vector. More details about the finite volume discretization of the governing equations can be found in Appendix A.

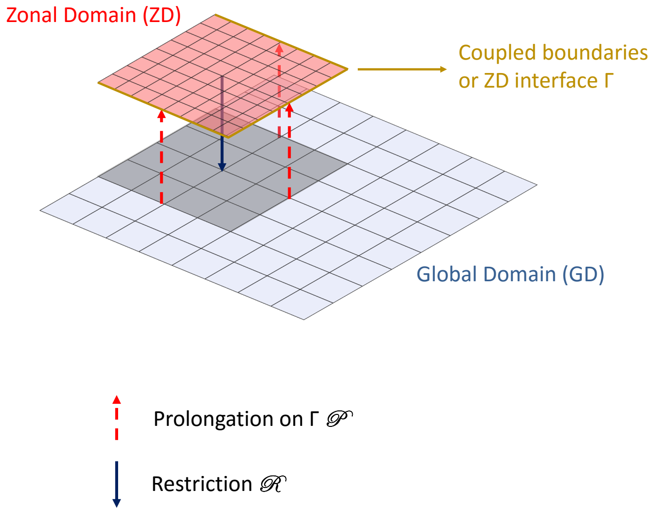

Zonal Domain (ZD)
Coupled boundaries or ZD interface Γ
Global Domain (GD)
Prolongation on Γ $\mathscr{P}$
Restriction $\mathscr{R}$

Figure 1: Global and zonal domains. Global domain (GD) is the main computational domain while Zonal Domain (ZD) is fully overlapped in GD. Communication from GD to interface Γ is done via the prolongation $\mathcal{P}$ operator. Communication from the ZD field to the underlying GD field is done via the restriction $\mathcal{R}$ operator.

3.2. Subcycling in time

A specific time-step is defined for each domain: Δt on GD and $\Delta\tau$ on ZD. Thus,

$$\Delta\tau = \frac{1}{r_t}\Delta t, \tag{16}$$

where r_t is the time-step ratio between ZD and GD. This ratio can be either fixed or adaptive for each iteration. In the case of a fixed time-step ratio, r_t is defined when Δt is fixed as well. In that case, r_t is equal to the spatial refinement ratio r_x between ZD and GD, which is the ratio between the GD average cell size $(\Delta x)_{GD}$ and the ZD average cell size $(\Delta x)_{ZD}$:

$$r_t = r_x = \frac{(\Delta x)_{GD}}{(\Delta x)_{ZD}} \quad . \tag{17}$$

On the other hand, for an adaptive time-step ratio, r_t varies at each time-step when GD and ZD time-steps are calculated based on a CFL constraint. An adaptive GD time-step Δt is calculated with the following relation [34]:

$$\Delta t = \min\left[\frac{Co_{max}}{Co}\Delta t^n, \left(1+\lambda_1\frac{Co_{max}}{Co}\right)\Delta t^n, \lambda_2\Delta t^n, \Delta t_{max}\right] \quad , \tag{18}$$

where the face-computed Courant number Co is defined as ,

$$Co = \frac{\boldsymbol{S}_f \cdot \boldsymbol{U}_f}{\boldsymbol{S}_f \cdot \boldsymbol{d}_{PN}}\Delta t \quad , \tag{19}$$

with $\boldsymbol{d}_{PN} = \boldsymbol{x}_P - \boldsymbol{x}_N$ the distance between the owner cell center P and the neighbor cell center N (see figure A.19). Δt^n refers to the time-step at the previous GD iteration (Δt being the one at the current iteration), Co_{max}= 0.5, Δt_{max} are user-defined parameters and $\lambda_1 = 0.1$ and $\lambda_2 = 1.2$ are two constant factors used to increase the time-step gradually and avoid unstable oscillations. A provisional value $\Delta\tau^*$ for the zonal grid time step is determined with a similar equation adapted to ZD.

Using relation (18) provides values for GD and ZD time-steps in the current time loop, respectively Δt and $\Delta\tau^*$. Since the time-step ratio between the two domains is an integer, the ZD time-step is adjusted to fulfill this requirement. First, the time step ratio is rounded to the upper integer bound:

$$r_t = floor\left(\frac{\Delta t}{\Delta\tau^*}\right) + 1 \quad , \tag{20}$$

where $floor(a)$ is a function to obtain the closest integer lower than a. Then, the effective ZD time-step $\Delta\tau$ is determined using equation (16).

3.3. Solver algorithm

The principle for one GD time-step is now described in the following and illustrated in figure 2. Details of the interpolation and correction steps are given later. One starts from the solution on both domains at $t = t_n$ and seeks to find the solution at $t_{n+1} = t_n + \Delta t$:

(1) **Solve entirely GD**, for one global Δt time step, using the single grid solver algorithm.

(2) Enter r_t loops:

(2.1) **Prolongation step**: The coupled boundary conditions of ZD, at the interface Γ, are updated based on interpolation in time and space of the GD solution at time t_{n+1}.

(2.2) **Solve entirely ZD**, for one zonal $\Delta\tau$ time step, using the single grid solver algorithm.

(2.3) Go back to (2.1) until t_{n+1} is reached.

(3) **Restriction step**: ZD solution fields at t_{n+1} are transferred to GD cells through interpolation procedures.

(4) **Correction of GD solution**: 1) A Poisson equation is solved to correct the volumetric face flux field. 2) Volume fraction values in cells along the ZD interface (lying outside) are corrected. 3) Mixture properties are updated.

The single grid solver algorithms, for *(RI)* and *(TDF)* models, are detailed in Appendix B.

3.4. Treatment of the zonal/global grid interface (step (2.1))

The values imposed on the coupled boundary conditions of ZD, at interface Γ, are computed at step (2.1) of the TPZG algorithm. A coupled boundary condition means that some of the values imposed at the patch are interpolated from the GD internal field. The two types of coupled boundary conditions that are used will be detailed later. This interpolation process is called *the prolongation step*, similarly to its denomination in the Multigrid method [35]. An illustration of the process is given in figure 4-(a). We now introduce the prolongation operator $\mathcal{P}(\phi)$ that interpolates any variable ϕ from the GD field to the zonal interface Γ. This operator combines the following three interpolation steps:

- step 2.1.1: Interpolation from cell centroids to face centers in GD,

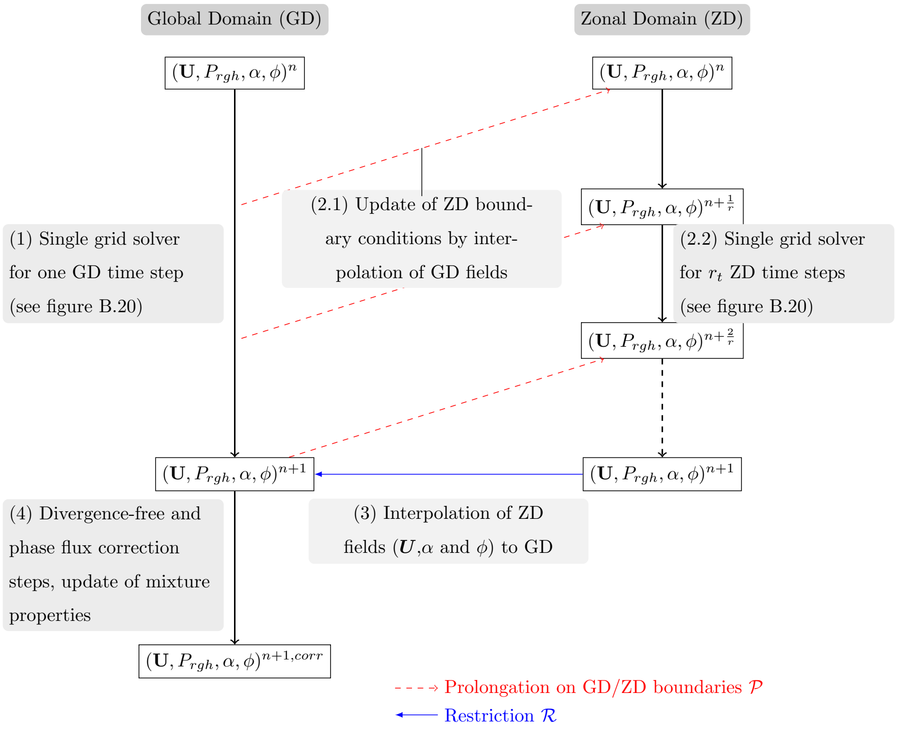

Global Domain (GD)
Zonal Domain (ZD)
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^n$
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^n$
(2.1) Update of ZD boundary conditions by interpolation of GD fields
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^{n+\frac{1}{r}}$
(1) Single grid solver for one GD time step (see figure B.20)
(2.2) Single grid solver for r_t ZD time steps (see figure B.20)
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^{n+\frac{2}{r}}$
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^{n+1}$
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^{n+1}$
(4) Divergence-free and phase flux correction steps, update of mixture properties
(3) Interpolation of ZD fields ($\boldsymbol{U}$,α and ϕ) to GD
$(\mathbf{U}, P_{rgh}, \alpha, \phi)^{n+1,corr}$
Prolongation on GD/ZD boundaries $\mathcal{P}$
Restriction $\mathcal{R}$

Figure 2: Zonal methodology algorithm; r_t is the time refinement ratio between ZD and GD. $\boldsymbol{U}$ is the velocity field, P_{rgh} is the pressure field, α the primary phase field and ϕ a transported scalar quantity field. Subscript $()^n$ is the index for time instant t and $()^{n+1}$ stands for the time instant $t + \Delta t$ after one time step.

- step 2.1.2: interpolation in time,
- step 2.1.3: interpolation tangential to the zonal interface.

The first step (2.1.1) depends on the variable that is interpolated. The other two are generic. We detail these procedures below.

step 2.1.1: In GD, cell centered fields are interpolated to the face center f:

Before interpolating from one domain to another, it is necessary to interpolate cell centered fields to face centers. Variables that are already face centers, *i.e.* a surface normal gradient $\boldsymbol{\nabla}_{\perp}\phi$, do not require this step. A surface normal gradient is the component, normal to the face, of the gradient of values at the centers of the two cells that the face connects.

Velocity:

The cell centered velocity field is interpolated with a linear scheme and a correction is added to ensure mass conservation. It consists of adding to the velocity vector magnitude the difference between the face flux value (which is mass conservative) and the scalar product of the velocity vector with interface normal:

$$\boldsymbol{U}_f = \bar{\boldsymbol{U}}_f + \left(\frac{F}{|\boldsymbol{S}_f|} - \bar{\boldsymbol{U}}_f \cdot \boldsymbol{n}_f\right)\boldsymbol{n}_f \quad , \tag{21}$$

where $\bar{\boldsymbol{U}}_f$ is a linear interpolation of the neighbor cell center values at face f (see figure A.19) and $\boldsymbol{n}_f$ the face normal vector,

$$\boldsymbol{n}_f = \frac{\boldsymbol{S}_f}{|\boldsymbol{S}_f|} \quad . \tag{22}$$

In that way, by dot producting equation (21) with $\boldsymbol{n}_f$, one gets

$$\boldsymbol{U}_f \cdot \boldsymbol{n}_f = \frac{F}{|\boldsymbol{S}_f|} \quad . \tag{23}$$

This approach ensures that an interpolated face velocity is coherent with the volumetric flux at this face.

Phase indicator:

Mass conservative approaches are considered for the phase indicator field interpolation. It is thus interpolated differently depending on which single grid solver is used:

– *(RI)* An expression for face center data is derived from the phase flux equation (A.7), detailed in Appendix A:

$$F_\alpha = F \times \alpha_f + F_r \times \alpha_f\left(1 - \alpha_f\right) \quad . \tag{24}$$

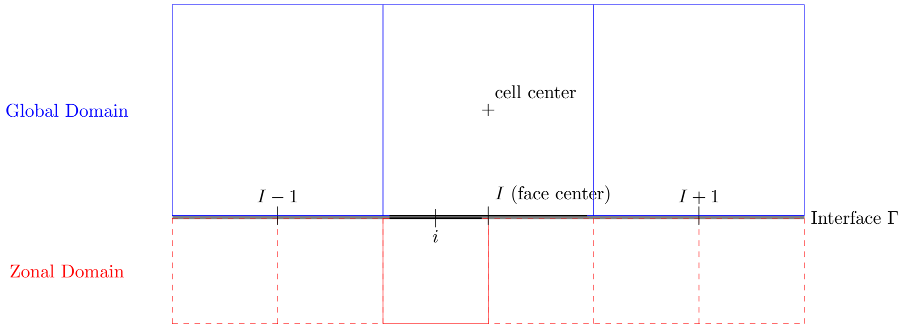


Figure 3: Interface Γ between the Zonal Domain (ZD) and the Global Domain (GD) for a mesh resolution ratio r_x of 2:1.

Equation (24) is rearranged to obtain the following expression:

$$\alpha_f = \frac{F_\alpha}{F + F_r\,(1-\alpha_f)} \quad . \tag{25}$$

Equation (25) is solved iteratively starting from the last known value.

– *(TDF)* An explicit expression for the face center phase indicator field is derived from equations (B.2) and (B.6) (Appendix B):

$$\alpha_f = \frac{\rho F - F \times \rho_2 - F_{\nu,\alpha} \times (\rho_1 - \rho_2)}{F \times (\rho_1 - \rho_2)} \quad . \tag{26}$$

Other variables:

As far as other cell centered variables are concerned (P_{rgh}, k or ε), a linear interpolation scheme is employed.

step 2.1.2: Previous variables from (2.1.1) are interpolated in time:

As stated before, sub-cycling in time of the zonal solver is used to reduce computational cost. Since GD is solved first, the solution is known at time indices n and $n+1$. Thus, any quantity can be interpolated in time between the bounds n and $n+1$, to solve the right part of the algorithm in figure 2.

Considering a k^{th} time-step from instants $n+\frac{(k-1)}{r_t}$ to $n+\frac{k}{r_t}$, boundary values of ϕ at instant $n+\frac{k}{r_t}$ are derived using a first order linear interpolation:

$$\phi^{n+k/r_t} = \phi^n + \frac{k}{r_t}\left(\phi^{n+1} - \phi^n\right) \quad , \tag{27}$$

with $k \in [1, r_t]$.

step 2.1.3: Previous variables from (2.1.2) are interpolated tangentially to the zonal interface Γ:

We use *left-skewed and right-skewed blended interpolation* for its property to conserve mass fluxes and its 2^{nd} order accuracy [23]. An illustration of this interpolation procedure is shown in figure 3 with a space ratio $r_x = 2:1$.

For a face centered value ϕ_f, blended left and right-skewed linear interpolation to point i is expressed as a function of ϕ_f at face indexes I, $I+1$ and $I-1$:

$$\left(\phi_f\right)_i = \left(\phi_f\right)_I + \frac{\boldsymbol{d}_{I,i}\cdot\boldsymbol{d}_{I,I+1}}{\boldsymbol{d}_{I,I+1}\cdot\boldsymbol{d}_{I,I+1}}\left(\phi_f\right)_{I+1} + \frac{\boldsymbol{d}_{I,i}\cdot\boldsymbol{d}_{I,I-1}}{\boldsymbol{d}_{I,I-1}\cdot\boldsymbol{d}_{I,I-1}}\left(\phi_f\right)_{I-1} \quad , \tag{28}$$

where $\boldsymbol{d}_{a,b}$ is the vector between two points a and b.

For interface corners only and to ease implementation, a 1^{st} order interpolation scheme is used, namely the *injection* scheme [23]. It consists of associating the ZD boundary face value with its nearest neighbor GD face value:

$$\left(\phi_f\right)_i = \left(\phi_f\right)_I \quad . \tag{29}$$

Boundary condition types at interface Γ.

Three types of boundary conditions are used at the zonal interface Γ and are detailed below. Their assignation to the field variables will be given later, depending on the numerical test considered.

(i) A *coupled Dirichlet* condition simply consists of imposing an interpolated field quantity from the GD to the ZD interface Γ, by using the prolongation operator $\mathcal{P}$ previously defined.

(ii) A *coupled inlet-outlet* condition is a mixed Dirichlet-Neumann condition, which switches between a coupled normal gradient when the fluid flows out of ZD at a patch face, and a coupled Dirichlet, when the fluid is flowing into ZD. A coupled normal gradient is defined by $\mathcal{P}(\boldsymbol{\nabla}_\perp \phi)$, while a coupled Dirichlet is defined by $\mathcal{P}(\phi)$. Table 1 describes this boundary condition.

Coupled inlet-outlet boundary condition	
$F_\Gamma \geq 0$ (out of domain)	apply $\mathcal{P}(\boldsymbol{\nabla}_\perp \phi)$
$F_\Gamma < 0$ (into domain)	apply $\mathcal{P}(\phi)$

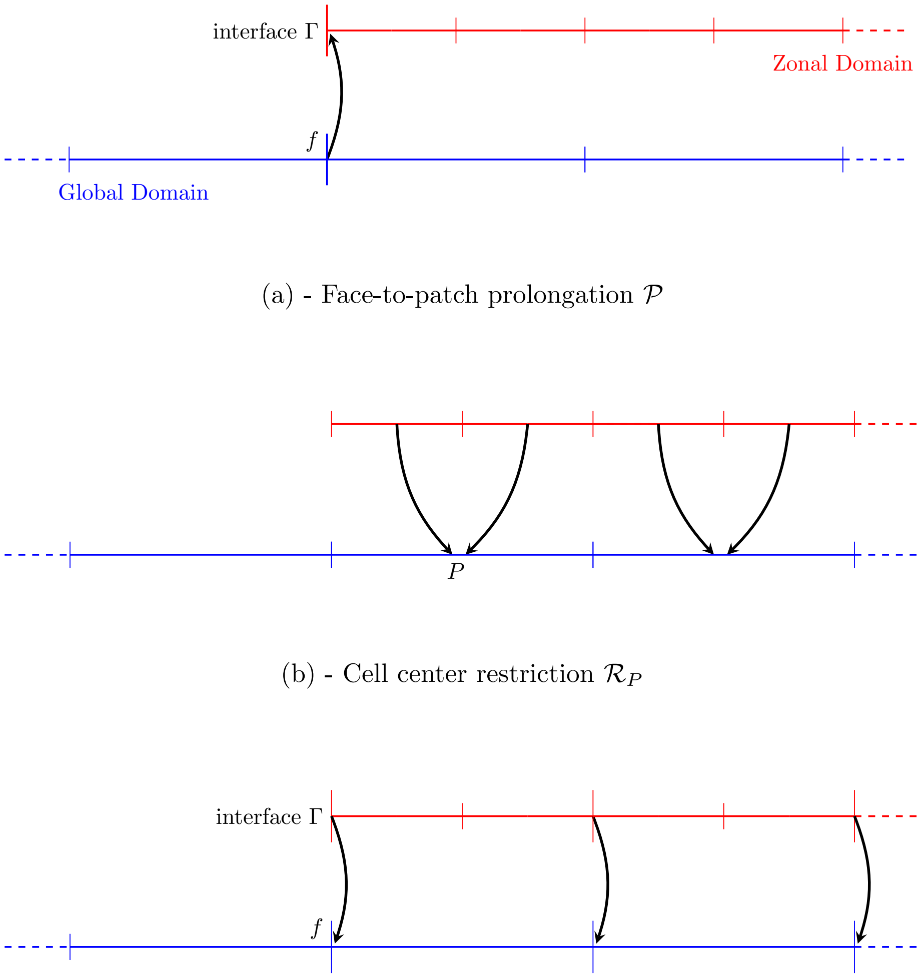
Table 1: Description of a *coupled inlet-outlet* boundary condition at interface Γ in ZD for a variable ϕ.

This approach makes it possible to benefit of the accuracy of the fine mesh when information is leaving ZD. It will be used as a coupled boundary condition for the phase indicator α and other transported scalars. It will also be used for velocity $\boldsymbol{U}$ when a *coupled Dirichlet* boundary condition (i) is used for pressure.

(iii) A Neumann *null gradient* boundary condition is used for dynamic pressure P_{rgh}, when a *coupled Dirichlet* boundary condition (i) is used for velocity.

3.5. Interpolation from zonal to global domain (step (3))

After resolution of the ZD solution over Δt, part of the solution field is interpolated from ZD to GD, to correct the coarse solution at instant t_{n+1}. This step is numbered (3) in the zonal solver algorithm and in figure 2. This process is called *restriction*, similarly to its denomination in the Multigrid method [35]. The cell centered restriction operator $\mathcal{R}_P$ brings variables from a pack of fine cells in ZD to an overlapped coarser cell in GD. We denote $\mathcal{R}_P\left(\phi\right)$ a standard cell centered variable on a domain after the restriction operator has been applied. We use a first order volume average for its simplicity and conservative property.



(a) - Face-to-patch prolongation $\mathcal{P}$

(b) - Cell center restriction $\mathcal{R}_P$

(c) - Face center restriction $\mathcal{R}_f$

Figure 4: Three types of interpolation between Zonal Domain (in red) and Global Domain (in blue), for a mesh refinement $r_x = 2$.

The volume average restriction operator is expressed as:

$$\mathcal{R}_P\left(\phi^{GD}\right) = \frac{\sum_i \left(\phi_i V_i\right)^{ZD}}{\sum_i V_i^{ZD}} \quad , \tag{30}$$

where subscript index $()_i$ denotes the subset of control volumes on the ZD level, that forms the control volume considered on the coarse GD level. An illustration of cell center restriction is given in figure 4-(b).

Similarly, the face centered restriction operator $\mathcal{R}_f$ brings variables from a group of fine faces of ZD to an overlapped coarser face in GD. Restriction of fine face fluxes consists of summing over the group each face flux value to get the coarser face flux value. The flux summing restriction operator is expressed as:

$$\mathcal{R}_f \left(F^{GD} \right) = \sum_i F_i^{ZD} \quad , \tag{31}$$

where subscript index $()_i$ denotes the subset of faces on the ZD level, that forms the face considered on the coarse GD level. An illustration of face center restriction is given in figure 4-(c).

The fields concerned by the restriction step are: the velocity field $\boldsymbol{U}$, the phase indicator field α,the volumetric face flux field F, the turbulent kinetic energy k and turbulent dissipation ε fields, when the case is turbulent and a $k-\varepsilon$ RANS approach is used.

Both cell center and face center restriction operators are applied on all covered cells and all covered faces in GD. The covered faces also include physical external boundary conditions.

3.6. Treatment of global grid correction (step (4))

In the previous restriction step, the phase indicator α, velocity $\boldsymbol{U}$, face flux F and transported scalar fields ϕ are interpolated from ZD to GD. The final step (4) in solver algorithm in figure 2, now consists of correcting the GD solution at time $n+1$. It is similar to active AMR [35] where the influence of local domain is spread over the entire global domain.

First, since volumetric face fluxes F have been interpolated from ZD to GD, major continuity errors can be observed at the interface between ZD and GD, because of flux mismatches. It is crucial to correct GD fluxes to ensure a divergence free velocity, low continuity errors and to obtain a converged solution. We refer to this correction step as the *divergence free correction* step. It consists of "smoothing" the face flux field by solving a Poisson equation and in ensuring divergence free constraint. A dummy pressure field P_{corr} is solved using following equation ,

$$\boldsymbol{\nabla} \cdot \left(\frac{1}{a_P} \boldsymbol{\nabla} P_{corr} \right) = \boldsymbol{\nabla} \cdot \boldsymbol{U} \quad , \tag{32}$$

where a_P are the diagonal coefficients of the system matrix. Equation 32 is expressed in discretized form as,

$$\sum_f \boldsymbol{S}_f \cdot \left[\left(\frac{1}{a_P} \right)_f (\boldsymbol{\nabla} P_{corr})_f \right] = \sum_f F \quad . \tag{33}$$

Then, the face flux field is corrected following the relation:

$$F_{corr} = F - \boldsymbol{S}_f \cdot \left[\left(\frac{1}{a_P} \right)_f (\boldsymbol{\nabla} P_{corr})_f \right] \quad . \tag{34}$$

This process will affect both face flux values in uncovered and covered parts of the domain, leading to different flux fields in zonal and global domains. However, it was observed that the impact of this correction is mainly focused outside the covered region while fluxes in the covered region are weakly corrected. In order to avoid any flux correction in the covered part, one could impose a zero pressure gradient, or also increase drastically the weight coefficients $1/a_P$ in the dummy pressure matrix. One could also solve this Poisson equation at both levels to keep the fine and coarse flux fields coherent. This would require a more complex algorithm and intergrid interpolations during the resolution, which has a non negligible cost.

Second, it is important to take into account phase flux mismatch at the interface and to correct the α values in coarse cells along the interface Γ, to ensure mass conservation. We refer to this correction step as the *phase flux correction* step. A coupled boundary conditions treatment in ZD imposes α values when the flux is going in ZD and a normal gradient when the flux is going out. Consequently, at the interface, phase fluxes in ZD do not match the underlying phase fluxes in GD. The cell-centered restriction step will correct the α in the covered part of GD, but not in the coarse side of the ZD interface. The current correction aims to correct α in these cells. To do so, a face-centered restriction is operated at the interface. To explain this correction, we shall start from equation (A.4) for the RI model and (A.8) for the TDF model. It is now expressed independently of the model as:

$$\alpha^{n+1} = \alpha^n - \frac{\Delta t}{V} \sum_f F_{\alpha,o} \quad , \tag{35}$$

where $F_{\alpha,o}$ stands for the overall phase flux in each model, i.e. advective plus compressive fluxes in the RI model and advective plus diffusive fluxes in the TDF model. Now, one introduces a similar conservation equation that takes into account the phase flux at the interface $(F_{\alpha,o})_\Gamma$, after face-centered restriction , namely $\mathcal{R}_f\left(F_{\alpha,o}\right)_\Gamma$, and the corrected phase value at the new instant α^{n+1}_{corr}:

$$\alpha^{n+1}_{corr} = \alpha^n - \frac{\Delta t}{V} \left[\sum_{f-1} F_{\alpha,o} + \mathcal{R}_f\left(F_{\alpha,o}\right)_\Gamma \right] \quad , \tag{36}$$

where $\sum_{f-1}$ stands for the sum over all faces of the cell except the face underlying the interface. Rearranging equation (36) and including equation (35), one obtains the following expression for α^{n+1}_{corr}:

$$\alpha^{n+1}_{corr} = \alpha^{n+1} - \frac{\Delta t}{V} \left[\mathcal{R}_f\left(F_{\alpha,o}\right)_\Gamma - \left(F_{\alpha,o}\right)_\Gamma \right] \quad . \tag{37}$$

Equation (37) is applied over all the cells along the zonal grid interface Γ.

Third and last, when the RI solver is used in GD, the interface geometry properties, curvature and position, are corrected using new values of the liquid volume fraction field α.

4. Applications

To assess the accuracy of solutions calculated with the TPZG approach previously detailed, two 2D test cases and one 3D application were set up with the proposed solver. The first 2D configuration is a rising bubble configuration set up by Hysing *et al.* [36]. The second 2D test case is a turbulent jet of liquid fuel into steady ambient air. For each of these two cases, validations of the TPZG approach are based on comparisons between a simulation using a coarse global mesh coupled with a refined zonal domain and a reference simulation using a single fine mesh over the whole domain.

In a third and last application, the TPZG solver was used to predict the primary breakup of a single cylindrical liquid jet in an air-blast atomizer configuration. Numerical results will be compared with experimental results from [37]. This configuration was chosen since experimental data are available about the mean liquid volume fraction in the primary atomization region and it seems pertinent to use two different two-phase flow models in this configuration.

4.1. Rising bubble

4.1.1. Configuration

The rising bubble problem investigates the ascending motion of a bubble of gas inside a quiescent liquid. A benchmark of this configuration was set up by Hysing *et al.* [36] and used later for studying the accuracy of `interFoam` solver by Klostermann *et al.* [38]. Hysing *et al.* [36] identified two different set-ups of the numerical experiment: ellipsoidal bubble and skirted bubble. They differ by the density ratio and the viscosity ratio between the two phases. In this work, the skirted set-up was selected. Indeed, it is better adapted to mesh refinement testing since in the finest mesh, the bubble forms thin filaments with small satellite bubbles. These structure detachments do not occur when the mesh is coarse [38]. It is thus pertinent to apply the current TPZG approach on this configuration.

On the basis of the benchmark definition of Hysing *et al.* [36], a two-dimensional computational domain with an aspect ratio $x : y = 1 : 2$ was employed, see figure 5. The bubble is initially centered at $(x, y) = (0.5, 0.5)$ with $r_{b0} = 0.25$ as the initial radius. The domain is fully enclosed by no-slip walls at the top and the

ρ_1	ρ_2	μ_1	μ_2	g	σ	Re
$kg.m^{-3}$	$kg.m^{-3}$	$kg.m^{-1}s^{-1}$	$kg\ m^{-1}s^{-1}$	m/s^2	kgs^{-2}	–
1000	1	10	0.1	0.98	1.96	35

Table 2: Physical properties of the rising bubble test case.

bottom and free slip walls on the left and right sides. The gravitational acceleration $\boldsymbol{g}$ points towards the bottom of the domain. It is a laminar low-Reynolds case, hence without turbulent modeling. The RI model was employed to follow the interface. The primary phase α refers to the liquid. Physical properties of the configuration are reported in table 2. The Reynolds number is defined as

$$Re = \frac{\rho_1 U_g L}{\mu_1} \quad , \tag{38}$$

with the rising velocity $U_g = \sqrt{g2r_{b0}}$ and the bubble characteristic length $L = 2r_{b0}$.

The fine zone is placed over the initial bubble position, as shown in figure 5, in a region in which the highest gradients will occur during the bubble rising. In this configuration, at the interface Γ, a *coupled Dirichlet* condition is employed for velocity, a *coupled inlet-outlet* condition for phase fraction α and a *null gradient* condition for dynamic pressure P_{rgh}.

The TPZG simulation is performed using a global coarse mesh coupled with a fine zone. Two different grid spacings are employed: fine and coarse. The fine mesh refinement relative to the coarse refinement is 2 : 1. Mesh sizes with these refinements are shown in table 3. The simulation time is $t_{final} = 3s$ with a fixed time step. The zonal sub-cycling ratio r_t is equal to the grid refinement ratio $r_x = 2$.

4.1.2. Results

The final bubble interface, at $t = 3s$, obtained with *coarse*, *fine* and *zonal* simulations are reported in figure 6. With the *coarse* simulation (left), the

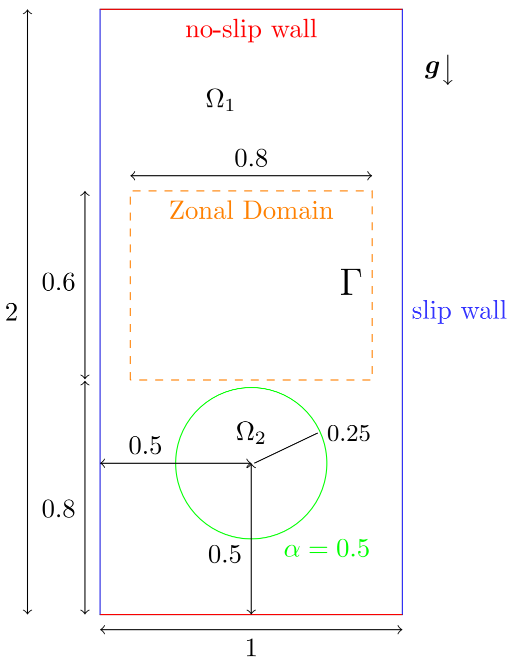


Figure 5: Simulation domain, boundary conditions and initial configuration of Hysing et al. [36] rising bubble problem. The zonal domain, with interface Γ, is placed above the initial position, with a mesh refinement of 2 : 1.

Configurations	Main (global) domain			Zonal domain		
	Mesh ref.	Mesh size	Two-phase model	Mesh ref.	Mesh size	Two-phase model
Coarse	Coarse	$13k$ cells	RI model	–	–	–
Fine	Fine	$51k$ cells	RI model	–	–	–
Zonal	Coarse	$13k$ cells	RI model	Fine	$13k$ cells	RI model

Table 3: Two-phase numerical simulations performed in the rising bubble configuration. Red and blue text colors respectively stand for coarse and fine mesh refinements.

main bubble structure is skirted and followed by two independent small structures. By decreasing the cell size (*fine* case, figure 6-central-($t = 3s$)), two additional satellite bubbles appear. The main mushroom shaped structure also shows shorter filaments. The bubble surface obtained with the *zonal* simulation (right) is in good agreement with the fine one, since the two additional satellite bubbles are present, with similar sizes. The size of the two lowest structures is

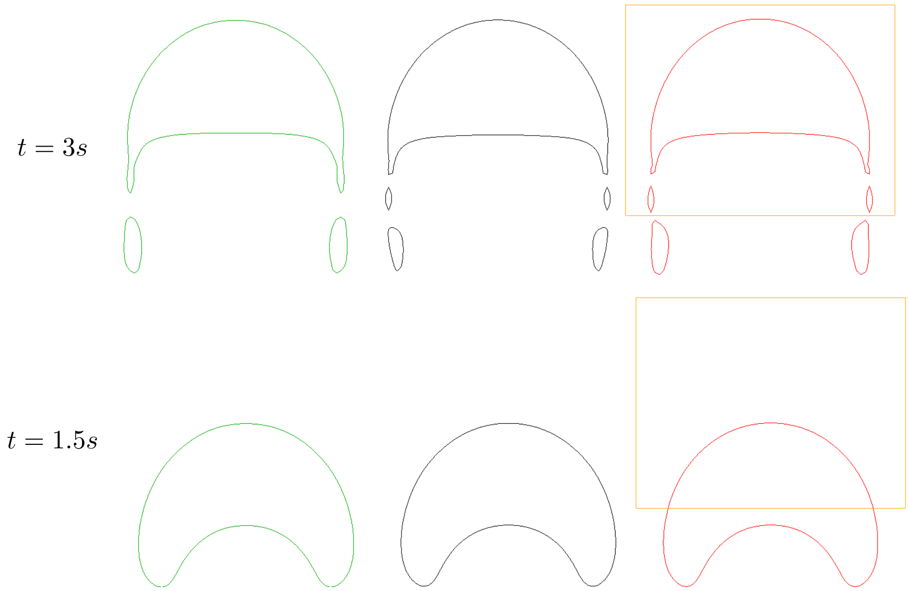


Figure 6: Interface ($\alpha = 0.5$) at (a) $t = 1.5s$ and (b) $t = 3s$ for three simulations of Hysing skirted bubble configuration [36]. From left to right, the green line is the coarse mesh, the black line is the fine mesh and the red line is the coarse mesh coupled with a fine zonal mesh. The zonal interface Γ is shown with an orange rectangular box.

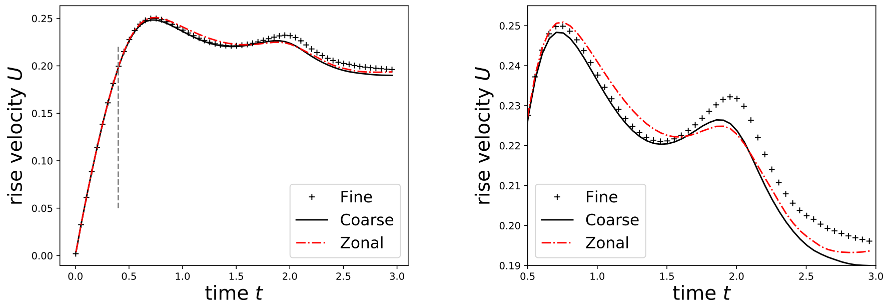
closest to that of the coarse ones, since they have not crossed the zonal interface Γ. In figure 6, at $t = 1.5s$, it can be seen that the bubble shape crosses the zonal domain smoothly without being affected by the presence of the mesh refinement discontinuity.

Quantitative results consist of the bubble rising velocity U, expressed as

$$U = \frac{\int_\Omega \alpha_g \left(\boldsymbol{U} \cdot \boldsymbol{y}\right) \mathrm{d}\Omega}{\int_\Omega \alpha_g \, \mathrm{d}\Omega} \quad , \tag{39}$$

with the gas volume fraction $\alpha_g = 1 - \alpha$. The evolution of this quantity is reported in figure 7, for the three cases. It can be seen that the two velocity peak values, V_{max1} and V_{max2}, at times $t = 0.7s$ and $t = 2.0s$, respectively, are slightly higher with the *fine* simulation than with the *coarse* one (respectively 0.63% and 2.49% higher), as well as the final velocity at time $t = 3s$.

The decrease of the second peak V_{max2} is more noticeable on the zoomed



(a) - Complete simulation period

(b) - Zoomed on velocity maxima

Figure 7: Evolution of bubble rising velocity V. The vertical grey dashed line indicates the time the bubble interface is in ZD.

profile. The *zonal* simulation gives results that are close to the *fine* and *coarse* ones. It is in good agreement with the fine simulation until the second peak. Then, the rising velocity is governed by the two lowest gas structures. In the *zonal* simulation, these structures stay outside ZD, *i.e.* in the coarse mesh. This explains why, after $t = 2\ s$, the rising velocity profile obtained with the *zonal* simulation is closer to the *coarse* profile than the *fine* profile.

The mass conservation property of the zonal scheme was checked. Since the computational domain is closed, the initial mass should remain constant during the whole simulation. The maximum mass error obtained with the zonal simulation is 0.005%, which is in a very low and acceptable range.

4.1.3. Performances

In comparison with a fully refined mesh, the zonal simulation reduce the computational cost by 73%, *i.e.* a speedup of 3.7. During the CPU effort, the percentages dedicated to the GD single grid solver, the ZD single grid solver, the prolongation step, the restriction step and the correction step are respectively 27%, 64%, 1.5%, 0.5% and 7%. Intergrid-communications and correction step account for 9% of total CPU effort.

Liquid injection	Turbulent intensity T_i 7.5 %	Turbulent length scale l 10 μm	Velocity U_1 100 m/s	Injector diameter D 100 μm
Property	Surface tension σ 0.07 kgs^{-2}	Liquid density ρ_1 696 kgm^{-3}	Gas density ρ_2 25 kgm^{-3}	Liquid viscosity μ_1 6.96×10^{-4} $kg\ m^{-1}s^{-1}$

Table 4: Properties of liquid and air in the liquid-air jet configuration.

4.2. Liquid-air jet

4.2.1. Configuration

This configuration deals with the turbulent injection of liquid into steady ambient air through a circular pipe. The mesh refinement inside the pipe has a major influence over jet velocity at the nozzle exit. Liquid is injected through the inlet boundary with a uniform velocity of 100 m/s. Its physical properties are shown in table 4. The primary phase α will refer to the liquid. The Reynolds number at the nozzle outlet $Re = 10^5$ demonstrates the turbulent character of the flow. Hence, a usual $k - \varepsilon$ RANS model is used in this configuration to account for the turbulence. This model solves two additional scalar fields (k and ε), which will be coupled in the zonal approach. Turbulence intensity and turbulent length scales at the inlet are equal to 7.5% and 10% of the mean outlet velocity and the injection diameter respectively.

The physical boundary conditions are shown in figure 8 along with the grids and domain dimensions. A uniform velocity is prescribed at the inlet boundary and total pressure is assigned on the atmosphere boundaries. Injector walls are modeled with a no-slip condition and high-Reynolds standard wall-functions are employed for boundary layer modeling. An angular symmetry is considered and slip conditions are used along the axis.

As aforementioned, the zonal grid is placed in the injection zone. Thus, two mesh refinements are used: fine and coarse. The fine mesh refinement relative to the coarse refinement is 5 : 1 in the y-direction close to the jet center line and 2 : 1 otherwise, as shown in figure 9. Mesh sizes with these refinements

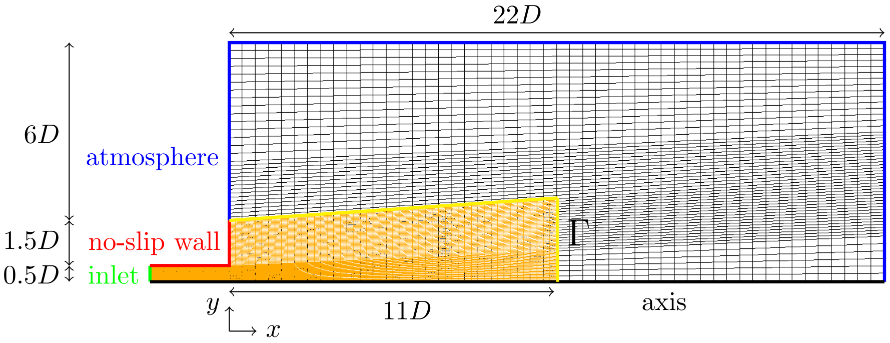


Figure 8: Configuration of the liquid-air jet configuration. The zonal domain (ZD), with fine refinement, is shown in yellow. The main (global) domain (GD), with coarse refinement, is shown in black. Γ stands for the zonal domain interface with the global domain and is indicated by the yellow line.

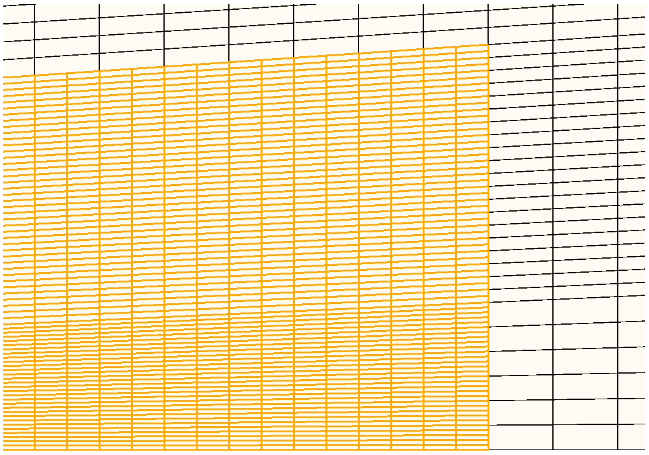


Figure 9: Zoom on the fine zonal mesh, shown in yellow. The fine mesh refinement relative to the coarse refinement is 5 : 1 in the y-direction close to the jet center line and 2 : 1 otherwise.

are shown in table 5, both for standard and zonal simulations, along with the solvers employed.

495 The length of the zonal domain is set up to half of the global domain length. Its height is sufficient to avoid having high velocity gradients between the liquid jet and the steady gas in the coarse mesh.

Boundary conditions of the zonal domain consists of physical (inlet, no-slip,

Configurations	Main (global) domain			Zonal domain		
	Mesh ref.	Mesh size	Two-phase model	Mesh ref.	Mesh size	Two-phase model
Fine TDF	Fine	$10k$ cells	TDF model	–	–	–
Fine RI	Fine	$10k$ cells	RI model	–	–	–
Zonal Hybrid	Coarse	$3k$ cells	TDF model	Fine	$4k$ cells	RI model

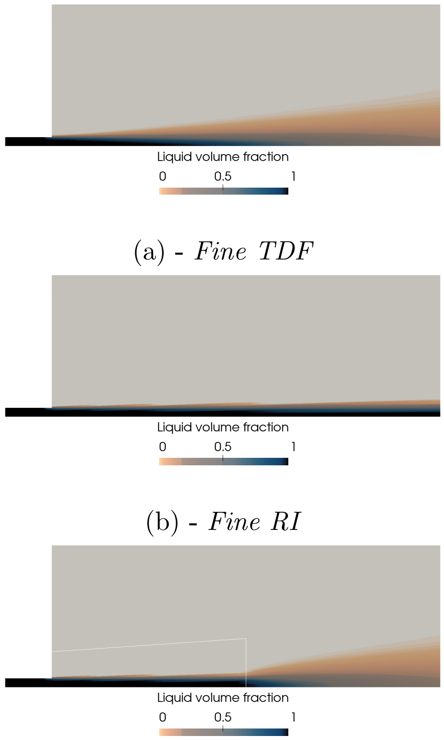
Table 5: Types of RANS performed in the liquid-air jet configuration. Red and blue text colors respectively stand for coarse and fine mesh refinements.

wall and axis) and coupled (interface Γ) boundaries, as shown in figure 8. Along Γ, pressure is imposed with values interpolated from GD. Thus, we use a *coupled Dirichlet* boundary condition for dynamic pressure P_{rgh} and *coupled inlet-outlet* boundary conditions for velocity $\boldsymbol{U}$, α, k and ε. In this test case, the ability of the zonal solver to handle two different two-phase solvers is tested. RI and TDF solvers are used in the zonal and global domains, respectively. During the prolongation step, the TDF phase fraction α_D is used to obtain boundary conditions for the RI phase fraction α_I at the zonal interface. Then, during the restriction step, the TDF field in the covered region of GD is corrected using the RI phase fraction α_I. Simulations are carried out with a dynamic GD time-step, following relation (18). Thus, the time-step ratio r_t is adjusted at each time-step by applying equation (20). Simulations show that this time-step ratio varies in a bounded range $r_t \in [2, 5]$ with an average value of $|r_t| = 3.1$.

4.2.2. Results

Steady state liquid volume fraction fields are shown in figure 10.

One can observe the diffusion of the liquid jet under action of the turbulent diffusion flux in the TDF model (fig. 10-(a)). The liquid jet penetrates further with the RI model (fig. 10-(b)). Indeed, the RI model requires a finer mesh and a three dimensional computation among other characteristics to capture the atomization process. The numerical diffusion is visible near the end of the domain. With the zonal simulation (fig. 10-(c)), diffusion of the liquid occurs after 11 diameters, *i.e.* the position of the interface Γ, where the governing equations



(a) - *Fine TDF*

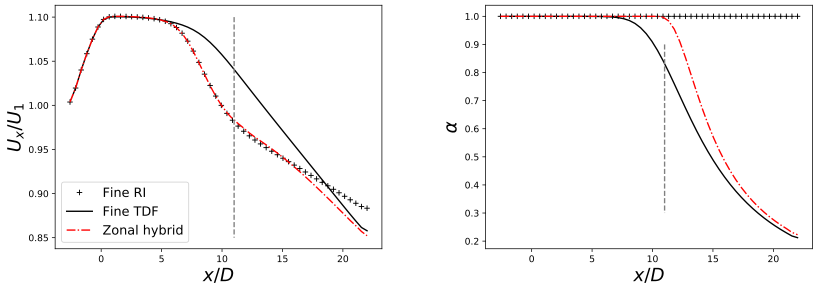
(b) - *Fine RI*

(c) - *Zonal hybrid*

Figure 10: Steady state liquid volume fraction fields. The zonal domain interface Γ is indicated by a white line.

switch from the RI to the TDF model.

Steady state velocity (a) and liquid volume fraction (b) profiles along the jet center line are shown in figure 11. The TDF model diffusion starts after the interface Γ as shown in figure 11-(b). The *zonal hybrid* profile is bounded by *fine RI* and *fine TDF* profiles. *The zonal hybrid* velocity profile (fig. 11-(a)) matches the reference *fine RI* profile upstream the zonal interface Γ, despite the coarse mesh refinement, thanks to the coupling with the fine zonal mesh. Downstream Γ, it tends toward the *fine TDF* profile. The transition between the two models is smoothed thanks to the mixed Dirichlet-Neumann boundary condition at Γ for velocity.



(a) (b)

Figure 11: Steady state (a) velocity and (b) liquid volume fraction profiles along jet center line. Zonal domain interface Γ at $x = 11D$ is indicated by a vertical dashed line.

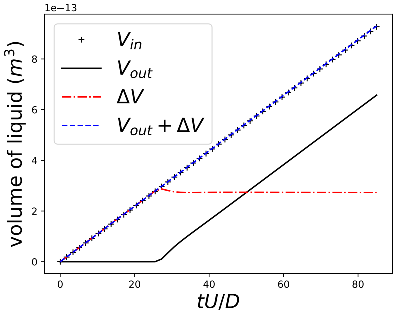


Figure 12: Evolution of volume of liquid using the zonal grid solver.

Mass conservation is a crucial point. The two single-grid two-phase solvers described in this work are strictly mass conservative; however, the zonal coupling may degrade this property. Let's monitor the conservation of liquid mass over time in the zonal run. If liquid mass is conserved, then the following volumetric relation should be satisfied in an incompressible flow:

$$V_{in}(t) = V_{out}(t) + \Delta V(t), \tag{40}$$

with $V_{in}(t)$ the liquid volume injected into the domain at the inlet during a time period t, the beginning of this period being set at $t = 0$, $V_{out}(t)$ the liquid volume flowing out of the domain and $\Delta V(t) = V(t) - V(0)$ the liquid volume in the domain at time t. The first two quantities are computed from surface integrals of the liquid volume fraction on the domain boundaries and the third quantity by a volume integral within the domain.

Figure 12 shows the evolution of these three quantities in the zonal run. Since the injected liquid mass flux is constant, $V_{in}(t)$ increases linearly with time. After $tU_1/D = 30$, the simulation reaches a steady state and the amount of liquid inside the computational domain is stabilized while the liquid outflow rate increases linearly. The evolutions of V_{in} and $V_{out} + \Delta V$ are very close with a maximum relative error of 0.4%, showing that equation (40) is satisfied using the zonal approach with the TDF and RI models in global and zonal domains respectively.

These results show that the present zonal algorithm is able to handle a hybrid approach for two-phase flows in a simple test case. It will be applied to a more complex configuration in the next section with the coaxial air-blast injector.

4.2.3. Performances

In comparison with a fully refined mesh, the zonal simulation reduces the computational cost by 40%, *i.e.* a speedup of 1.7. During the CPU effort, the percentages dedicated to the GD single grid solver, the ZD single grid solver, the prolongation step, the restriction step and the correction steps are respectively 15.5%, 66.0%, 9.0%, 0.5% and 9.0%. Intergrid-communications and the correction step thus account for 18.5% of total CPU effort. These performances are independent of the two-phase solver employed, since the structure of the RI and TDF algorithms is very similar.

4.3. Coaxial air-blast injector

4.3.1. Configuration

The configuration considered here, taken from Stepowski *et al.* [37], consists of injecting a low-speed liquid through a circular pipe and a high-speed gas through an annular pipe into a steady atmosphere. The liquid used is pure water, and ambient gas is dry air, leading to a density ratio of approximately 1000. The sketch of the injector is presented in figure 13. The geometrical

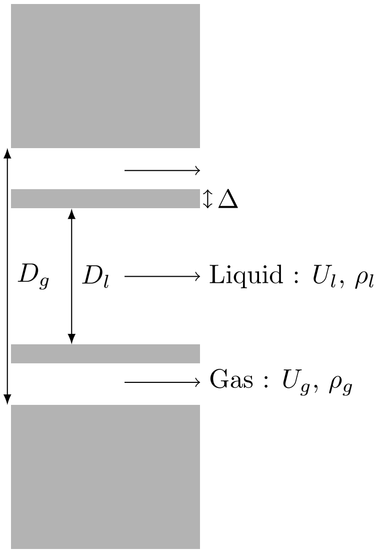


Figure 13: Schematic of the injector used by Stepowski et al. [37].

characteristics of the experimental device are as follows: $D_l = 1.8\ mm$, $D_g = 3.4\ mm$, $\Delta = 0.25\ mm$, $U_g = 115\ m/s$, $\rho_g = 1.2\ kg/m^3$, $U_l = 1.3\ m/s$, $\rho_l = 1000\ kg/m^3$. To obtain the near field of the liquid volume fraction, α, the fluorescence emission of an additional species incorporated into the water induced by a pulsed laser sheet was used [37]. The surface tension coefficient σ is derived from the aerodynamic Weber number, defined as,

$$We_g = \frac{\rho_g \left(U_g - U_l\right)^2 D_l}{\sigma} \quad . \tag{41}$$

Gas and liquid Reynolds numbers, respectively expressed as,

$$Re_g = \frac{\rho_g U_g \left(D_g - D_l - 2\Delta\right)}{\mu_g} \quad , \tag{42}$$

and

$$Re_l = \frac{\rho_l U_l D_l}{\mu_l} \quad , \tag{43}$$

are high enough for turbulent flows to be expected. The momentum flux ratio J, expressed as,

$$J = \frac{\rho_g U_g^2}{\rho_l U_l^2} \quad , \tag{44}$$

plays an important role in destabilization of the liquid jet and in the liquid core length. The values of the previous characteristic non-dimensional numbers are reported in table 6.

U_g (m/s)	U_l (m/s)	We	Re_g	Re_l	J
115	1.3	500	8000	2600	10

Table 6: Simulated flow conditions.

The WALE (Wall-Adapting Local Eddy-viscosity) model [39] is used in this configuration. This LES model recovers the proper near-wall scaling for the eddy viscosity without requiring a dynamic procedure. Moreover, this model does not need to solve an additional scalar transport equation. Hence, no additional variable coupling is necessary between global and zonal domains.

The global computational domain is defined as a cylindrical mesh with a height of $16D_l$ and a diameter of $8D_l$, as shown in figure 14.

Independently of mesh refinement, the mesh is expanded in axial and radial directions to focus refinement efforts in a region of size $[-0.1D_l, 3D_l] \times [-1.1D_l, 1.1D_l] \times [-1.1D_l, 1.1D_l]$ that covers the primary atomization area. Three different mesh refinements will be used in the following simulations: one *coarse* refinement to reduce computational cost; and one *medium* refinement to obtain, with a reasonable computational cost, correct predictive results regarding the mean liquid volume fraction in the primary atomization region. Finally, one *fine* refinement to get accurate predictive results. The refinement ratio between each level of refinement is $r_x = 2$. Thus, a cell with *coarse* refinement is split into 8 cells in *medium* refinement.

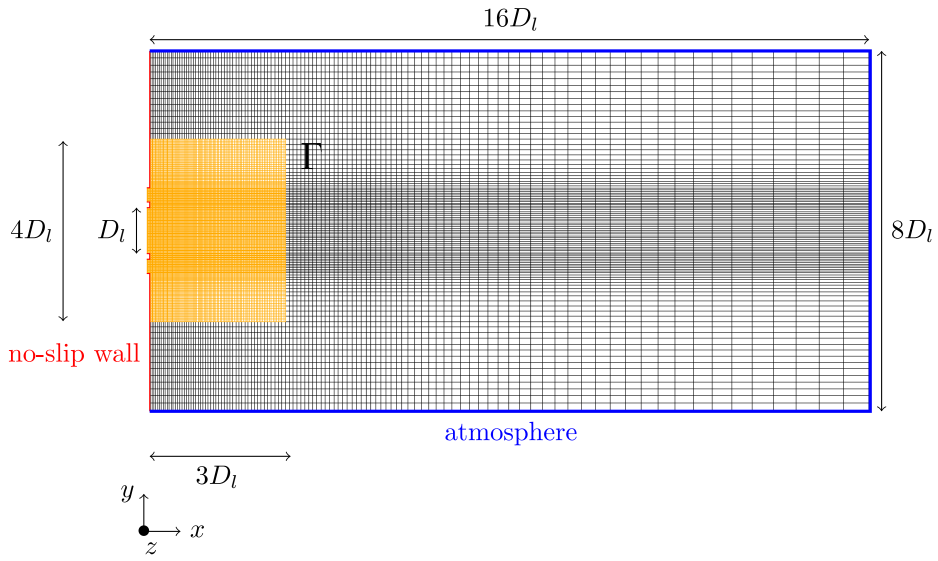


Figure 14: Dimensions and boundary conditions of the 3D computational domain. The zonal domain, with medium refinement, is shown in yellow. The global domain, with coarse refinement, is shown in black. Γ stands for the zonal interface with the global domain.

Concerning the zonal simulations, a *coupled Dirichlet* boundary condition is used for pressure along the global-zonal interface Γ, while *coupled inlet-outlet* conditions are used for velocity $\boldsymbol{U}$ and liquid volume fraction α. During the prolongation step, the TDF phase fraction α_D is used to obtain boundary conditions for the RI phase fraction α_I at the zonal interface. Then, during the restriction step, the RI phase fraction α_I is used to correct the TDF field in the covered region of GD. The zonal domain is placed in the primary atomization, defined by $[-0.1D_l, 3D_l] \times [-2D_l, 2D_l] \times [-2D_l, 2D_l]$, as shown in yellow in figure 14. The length of the zonal domain, equal to $3D_l$, is chosen based on the experimental profile of the mean liquid volume fraction along the jet center line. As shown in figure 17, it is approximatively equal to 0.1 at $x = 3D_l$. This value is often considered as the transition value between separated and dispersed two-phase flow zones [40]. Therefore, it is pertinent to switch from the RI to the TDF model at this longitudinal location. The diameter of the zonal domain, equal to $4D_l$, was chosen to avoid high gradients of velocity and liquid volume

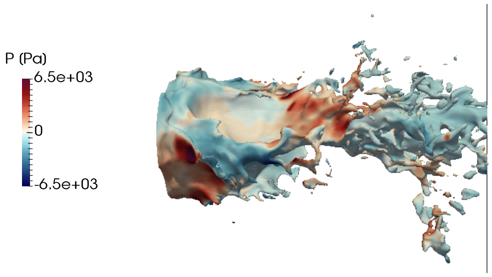


Figure 15: Three-dimensional snapshot of the phase indicator isosurface $\alpha = 0.5$ obtained with the *fine zonal hybrid* simulation, along with the dynamic pressure field. The zonal interface Γ is shown with a black line.

fraction along the zonal-global interface.
Coarse mesh refinement will be used in the global domain to reduce the impact of the mesh refinement in the dispersed spray zone on the computational cost.

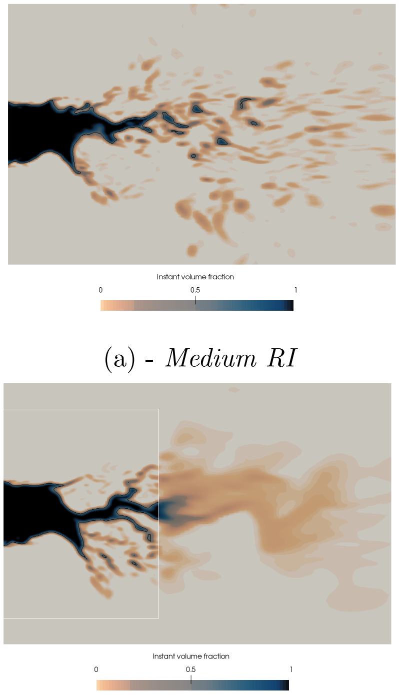
Four different LES of the coaxial atomization configuration from Stepowski *et al.* [37] were performed. Their mesh sizes and the two-phase model employed are given in table 7. Simulations were carried out with a dynamic GD time-step, following relation (18).

Configurations	Main (global) domain			Zonal domain		
	Mesh ref.	Mesh size	Two-phase model	Mesh ref.	Mesh size	Two-phase model
Medium RI	Medium	2.1M cells	RI model	–	–	–
Medium TDF	Medium	2.1M cells	TDF model	–	–	–
Medium zonal hybrid	Coarse	260k cells	TDF model	Medium	690k cells	RI model
Fine zonal hybrid	Coarse	260k cells	TDF model	Fine	5.5M cells	RI model

Table 7: LES performed in the coaxial atomizer configuration of Stepowski *et al.* [37]. Brown, red and blue text colors respectively stand for coarse, medium and fine mesh refinements.

4.3.2. Results

The instantaneous liquid structures obtained with the *fine zonal hybrid* simulation are shown in figure 15 with the isosurface $\alpha = 0.5$. The influence of pressure in liquid detachments is highlighted. Over pressure destabilizes and provokes the separation of structures from the liquid core. The isosurface



(a) - *Medium RI*

(b) - *Medium zonal hybrid*

Figure 16: Instant liquid volume fraction fields. The isocontour $\alpha = 0.5$ is shown with black lines when the RI model is used. The interface of the zonal domain is shown with white a rectangle.

$\alpha = 0.5$ is cut by the zonal domain right interface Γ which defines the end of the region governed by the RI model. In figure 16-(a) is shown the instant liquid volume fraction field using the RI model in the whole computational domain (*medium RI* configuration). In the second half of the picture, *i.e.* dowstream the primary atomization region, it is observed that the isolated liquid droplets are not correctly captured by the RI method. Droplets are numerically diffused despite the *medium* mesh refinement. The compressive scheme of the RI model also ensures that unresolved droplets remain smaller than the cell size they are advected in. Thus, it is pertinent to switch to another model, hence the *medium zonal hybrid* configuration shown in figure 16-(b), in which the TDF model is used after $x = 3D_l$. The TDF model field shows a diffusive behavior that is

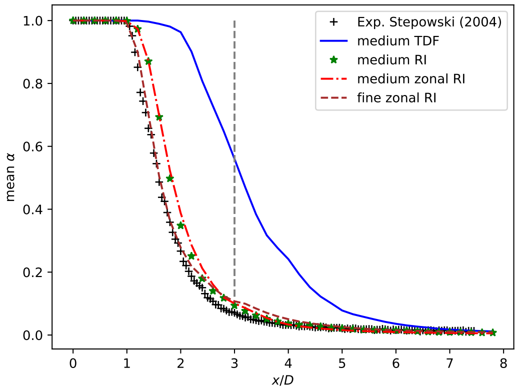


Figure 17: Mean liquid volume fraction axial profiles, along the injector central axis. The vertical gray dashed line stands for the zonal grid interface Γ.

here not numerical but physical. This approach makes it possible to coarsen the mesh in the dispersed spray region and thus to reduce the computational cost, as discussed below.

We can carry out a quantitative comparison between the different computations and the experiment [37] with the mean volume fraction axial profiles along the jet center line shown in figure 17. The TDF model is shown to be unsuitable for treating atomization in the primary zone, since, despite the *medium* mesh refinement, the *medium TDF* profile is far from the experimental profile. On the other hand, the *medium RI* profile shows a good agreement with experimental liquid penetration in the primary atomization region. However, as shown previously with qualitative results, this model is to be avoided in the dispersed spray region.

The *medium zonal hybrid* profile is here in good agreement with the *medium RI* one, meaning that the coupling does not affect the flow in the zonal domain region. Refining the mesh close to the nozzle gives results closer to the experimental data: with the *fine zonal hybrid* configuration, the liquid core penetration is accurately predicted.

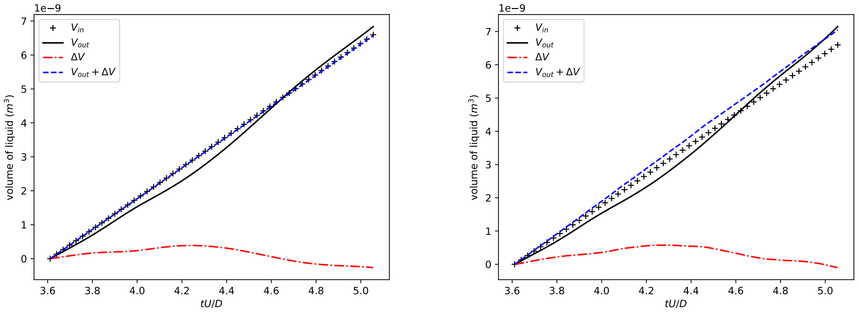
The mass conservation property of the *medium zonal hybrid* was checked. When it was judged that the initial transient period was over, *e.g.* $tU_l/D_l = 3.25$, the quantities $V_{in}(t)$, $V_{out}(t)$ and $\Delta V(t)$, previously detailed in the liquid-air jet test case, were calculated.

Figure 18-(a) shows the evolution of these quantities, plus the sum $V_{out}(t) + \Delta V(t)$, for a short period of time. Since the liquid mass influx is constant, $V_{in}(t)$ increases linearly with time. However, the rate of liquid volume exiting the simulation varies due to turbulent fluctuations and the flapping of the liquid jet. The calculated value of $V_{out}(t) + \Delta V(t)$ matches $V_{in}(t)$ with a maximum error of 0.4%.

In order to examine the influence of the *phase flux correction* step (see section 3.6 for more details), a simulation without this correction was run. The result plotted in figure 18-(b) shows an increase of the liquid volume inside the domain (ΔV), hence a higher liquid volume error (6.8%). This is due to the fact that since the phase flux mismatch along the coarse-fine interface is not longer corrected, more liquid is accumulated inside the domain. It confirms the importance of this correction step in the present numerical algorithm ensure mass conservation.

4.3.3. Performances

The computational costs of the *medium RI* and the *medium zonal hybrid* simulations were compared. The *medium zonal hybrid* simulation requires a global CPU effort of 520 h for a 10 ms simulation while the *medium RI* one requires $1,230$ h. Thus, the zonal simulation reduces computational cost by 57%, *i.e.* a speedup of 2.4. This illustrates one the main advantage of this zonal approach, *i.e.* to lower the impact of the primary atomization resolution on the global computational cost, by diminishing the number of grid points in



(a) - With correction of α

(b) - Without

Figure 18: Evolution of volume of liquid in the *medium zonal hybrid* simulation.

the secondary atomization and dispersed droplet zones that are far from the injectors. The speedup of the *fine zonal hybrid* simulation compared to a *fine RI* simulation with $17M$ cells was estimated to be 3.

In the *medium zonal hybrid* CPU duration, the percentages dedicated to the GD single grid solver, the ZD single grid solver, the prolongation step, the restriction step and the global grid correction step are respectively 18%, 62%, 3%, 2% and 15%. The correction step is more costly compared to intergrid-communications since it involves solving implicitly a Poisson equation. Its cost is in the same order as the part relative to the GD solver.

5. Conclusion

An algorithm that allows for the coupled multi-approach resolution of turbulent and laminar two-phase flows within the Finite-Volume framework OpenFOAM® has been presented. One advantage of the algorithm lies in sub-cycling in time making it possible to refine the grid locally without impacting the global computational time-step. Another advantage is the possibility to use two different single grid solvers, a sharp interface solver in one part of the computational domain and a diffuse interface solver in the rest of the domain.

Qualitative and quantitative measures were used to validate the zonal approach. Simulation of a rising air bubble was carried out, with a refined zone placed above the initial bubble position. It shows good agreement with the fully refined solution, the bubble crossing the coarse-fine interface smoothly. RANS modeling liquid jet atomization was also carried out with grid refinement in the first half of the domain. Zonal solutions show good agreement with the fully refined mesh, with both sharp and diffuse interface models. A hybrid model was successfully applied , with a sharp interface in the refined zone and a diffuse interface in the rest of the domain. Lastly, LES of a coaxial air-blast injector was performed. The zonal domain was placed near the nozzle injector and extended up to 3 diameters further. Inside this domain, the sharp interface solver captured well the liquid core penetration given by experimental results. Downstream the zonal domain boundary, the diffuse interface solver managed to solve the dispersed two-phase flow topology with a coarse mesh refinement. This method represents a speedup of 2.4 in comparison with a single grid solver. Mass conservation was checked; the zonal coupling respects this conservative aspect inherent to finite volume solvers.

In the three applications, the benefit of the zonal approach in terms of computational cost is obvious in that the computational time is reduced.

6. Acknowledgments

The authors acknowledge the financial support of the French Agence Nationale de la Recherche (ANR), through the program Investissements d avenir(ANR-10-LABX-09-01) , LabEx EMC3. This work was granted access to the HPC resources of IDRIS, TGCC and CINES under the allocation A0032B06153 made by GENCI (Grand Equipement National de Calcul Intensif).

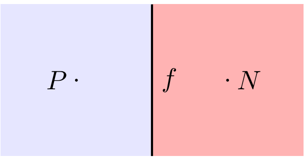


Figure A.19: The face f whose owner is cell center P and the neighbour cell center N.

Appendix A. Discretized model equations

The discretized forms, using finite volumes, of the momentum and transport equations are detailed in this appendix since the coupling between the two grids on which two different solvers are used involves an exchange of information between the discrete parameters such as the face fluxes or volume data.

Appendix A.1. Momentum equation

The discretized form of the momentum equation (3) is presented following the standard OpenFOAM® formalism :

$$\begin{aligned}&\frac{V}{\Delta t}\left(\rho^{n+1}\boldsymbol{U}^{n+1}-\rho^{n}\boldsymbol{U}^{n}\right)+\sum_f(\rho F)^{n+1}\boldsymbol{U}_f^{n+1}=\\&-V\boldsymbol{\nabla}(P_{rgh}{}^{n})+\sum_f\left(\mu_{eff}^{n+1}\right)_f\boldsymbol{S}_f\cdot\left(\boldsymbol{\nabla}\boldsymbol{U}^{n+1}\right)_f\\&+\sum_f\boldsymbol{U}_f^{n}\boldsymbol{S}_f\cdot\left(\boldsymbol{\nabla}\mu_{eff}^{n+1}\right)_f+V\boldsymbol{Q}^{n},\end{aligned}\tag{A.1}$$

where the superscripts $()^n$ and $()^{n+1}$ identify instants t and $t+\Delta t$, respectively. Subscript $()_f$ indicates the face values, interpolated between the center P of the volume considered and a neighbor cell center N (see figure A.19). The cell volume is V and the volumetric flow rate through one of its faces f is defined as

$$F=\boldsymbol{S}_f\cdot\boldsymbol{U}_f,\tag{A.2}$$

where $\boldsymbol{S}_f$ is the face normal vector.

Appendix A.2. Phase indicator

RI solver. The transport equation for volume fraction α_I (Eq. 11) can be semi-discretized as,

$$\alpha_I^{n+1} = \alpha_I^n - \Delta t\{\boldsymbol{\nabla} \cdot [\boldsymbol{U}\alpha_I + \boldsymbol{U}_c\alpha_I\,(1-\alpha_I)]\}^n \quad , \tag{A.3}$$

and it is fully discretized as,

$$\alpha_I^{n+1} = \alpha_I^n - \frac{\Delta t}{V}\sum_f \left[F \times \alpha_{I,f} + F_c \times \alpha_{I,f}\,(1-\alpha_{I,f})\right]^n \quad , \tag{A.4}$$

where F_c is the compressive face flux,

$$F_c = (U_c)_f\,\hat{S} \quad . \tag{A.5}$$

The magnitude of the interface normal face vector $\hat{S}$ is defined as:

$$\hat{S} = \boldsymbol{n} \cdot \boldsymbol{S}_f = \frac{\boldsymbol{\nabla}\alpha_I}{|\nabla\alpha_I|} \cdot \boldsymbol{S}_f \quad . \tag{A.6}$$

For the sake of clarity, the two terms present in the RHS sum of equation (A.4) are grouped together into one global phase flux term namely $F_{\alpha,I}$:

$$F_{\alpha,I} = F \times \alpha_{I,f} + F_C \times \alpha_{I,f}\,(1-\alpha_{I,f}) \quad . \tag{A.7}$$

TDF solver. The transport equation for the primary phase indicator α_D is given by equation (13). It is discretized as:

$$\alpha_D^{n+1} = \alpha_D^n - \frac{\Delta t}{V}\sum_f \left(\underbrace{{F_{\alpha,D}}^n}_{advective} + \underbrace{{F_{\nu_t}}^{n+1}}_{diffusive} \right) \quad , \tag{A.8}$$

where the turbulent diffusive phase flux F_{ν_t} is defined as,

$$F_{\nu_t} = -\left(\frac{\nu_t}{Sc_t}\right)_f \boldsymbol{S}_f \cdot \boldsymbol{\nabla}\alpha_{D,f} \quad . \tag{A.9}$$

and the advective phase flux as,

$$F_{\alpha,D} = F \times \alpha_{D,f} \quad . \tag{A.10}$$

Appendix B. Segregated pressure-based solver

In this section a joint description is proposed for both RI and TDF solvers, their structures being very similar. Parts specific to the resolved interface solver are highlighted by *(RI)* while parts specific to the turbulent diffusive flux solver are denoted by *(TDF)* . The RI and TDF solvers are existing methods, respectively named as `interFoam` and `twoLiquidMixingFoam` solvers, available in the OpenFOAM® library. The RI solver has been detailed in the thesis of Rusche [31] and studied for instance in [38, 41]. The TDF solver is based on the Eulerian-Lagrangian Spray Atomization (ELSA) method from Vallet and Borghi [42]. It is a single-fluid approach composed of two species. The modeling approach employed in this work is similar to the turbulent mixing of two miscible fluids showing highly variable densities. It thus justifies th@incollectione use of the `twoLiquidMixingFoam` solver as TDF solver.

(1) The first step of the algorithm consists in **resolving recursively the advection equation for the phase fraction** α. This equation is derived from equations A.4 and A.8 for *(RI)* and *(TDF)* solvers respectively, except that the implicit diffusive term in equation A.8 is left aside for a later resolution. It is common in VOF-based methods for the convergence and stability of the solution procedure to be very sensitive with respect to the phase fraction equation. Consequently, it is beneficial to solve the phase fraction equation in several subcycles within a single time step. This subcycle consists in solving the advection equation $N_{\alpha SC}$ times and then averaging the mass flux. This advection equation is defined as:

$$\alpha^{i+1} = \alpha^i - \frac{\delta t}{V} \sum_f F_\alpha(\alpha^i) \quad , \tag{B.1}$$

with subcycle index $i \in [1, N_{\alpha SC}]$ and subcycle time-step $\delta t = \Delta t / N_{\alpha SC}$. The first α^i is initialized with $\alpha^i = \alpha^n$. At the end of the loop, $\alpha^{n+1} = \alpha^{i+1}$. Depending on the solver, equation (A.7) or (A.10) is used to recursively determine F_α in equation (B.1). For each sub-cycle loop, equation

(B.1) is solved to update the F_α^i value. In this work, $N_{\alpha SC}$ is equal to 2 , to divide the CFL constraint by 2. The MULES (Multidimensional Universal Limiter for Explicit Solution) [43] limiter is used to integrate equation (B.1) to ensure boundedness of the solution.

At the end of each sub-cycle loop, the advective mass flux ρF, necessary in stages (4) and (5) when computing the momentum conservation equation, is computed at sub-cycle index i thanks to the following expression:

$$\rho F^i = F_\alpha^i \times (\rho_1 - \rho_2)_f + F \times (\rho_2)_f \quad . \tag{B.2}$$

Finally, after the $N_{\alpha SC}$ loops, the advective mass flux is computed by time-averaging the previous sub-cycled mass fluxes:

$$\rho F^{n+1} = \frac{\sum_i \rho F^i}{N_{\alpha SC}} \quad . \tag{B.3}$$

(2) *(TDF)* **Transport equation of α with the TDF term of equation** (A.8) **is now determined**. We introduce $\alpha_{D,adv}$ as the previous solution of the advective transport equation (B.1):

$$\alpha_{D,adv} = \alpha^n - \frac{\Delta t}{V} \sum_f F_\alpha(\alpha^n) \quad , \tag{B.4}$$

and insert it in equation (A.8) to obtain:

$$\frac{\alpha_D^{n+1} - \alpha_D^n}{\Delta t} - \frac{\alpha_{D,adv} - \alpha_D^n}{\Delta t} = \sum_f F_{\nu_t}^{n+1} \quad . \tag{B.5}$$

Once equation (B.5) has been solved, the turbulent diffusive mass flux $F_{\nu_t} \times (\rho_1 - \rho_2)$ is added to the advective mass flux,

$$\rho F^{n+1} = \rho F^{n+1} + F_{\nu_t}^{n+1} \times (\rho_1 - \rho_2) \quad . \tag{B.6}$$

(3) **Update of mixture properties:**

(3.1) Density and viscosity are then updated,

$$\rho^{n+1} = \alpha^{n+1}\rho_1 + (1-\alpha^{n+1})\rho_2 \quad , \tag{B.7}$$

$$\mu^{n+1} = \alpha^{n+1}\mu_1 + (1-\alpha^{n+1})\mu_2 \quad . \tag{B.8}$$

(3.2) *(RI)* Curvature is also updated,

$$\kappa^{n+1} = -\boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}\alpha_I^{n+1}}{|\boldsymbol{\nabla}\alpha_I^{n+1}|}\right) \quad , \tag{B.9}$$

(4) **Solving the momentum predictor:**

This step is optional: it helps to get closer to the velocity solution at $n+1$ with an estimated velocity $\boldsymbol{U}^*$. It involves an implicit resolution which has a non negligible cost. The solution is derived from the pressure p^n and velocity face flux F^n fields at the previous time t_n. Thus, this predicted velocity is not divergence-free, *i.e.* $\boldsymbol{\nabla}\cdot\boldsymbol{U}^* \neq 0$. This will be corrected in the next PISO loop stage.

Starting from semi-discretized momentum equation (A.1) and introducing matrix coefficients, one gets the following expression for $\boldsymbol{U}^*$:

$$\boldsymbol{U}^* = \frac{\boldsymbol{H}(\boldsymbol{U}^*)}{a_P} + \frac{-\boldsymbol{\nabla}({P_{rgh}}^n) + \boldsymbol{Q}}{a_P} \quad , \tag{B.10}$$

where a_P are the diagonal coefficients of the system matrix and $\boldsymbol{H}(\boldsymbol{U}^*)$ groups the source terms of the transient, advective and diffusive terms:

$$\begin{aligned}\frac{\boldsymbol{H}(\boldsymbol{U}^*)}{a_P} =& \frac{\rho^n}{\rho^{n+1}}\boldsymbol{U}^n \\ &- \frac{\Delta t}{\rho^{n+1}V}\left[\sum_f (\rho F)^{n+1}\boldsymbol{U}_f^*\right] \\ &- \frac{\Delta t}{\rho^{n+1}V}\left[\sum_f \left(\mu_{eff}^{n+1}\right)_f \boldsymbol{S}_f\cdot(\boldsymbol{\nabla}\boldsymbol{U}^*)_f + \sum_f \boldsymbol{U}_f^n\boldsymbol{S}_f\cdot\left(\boldsymbol{\nabla}\mu_{eff}^{n+1}\right)_f\right] \quad .\end{aligned} \tag{B.11}$$

The source term expression depends on which model is used. In the RI solver, it is expressed as:

$$\boldsymbol{Q_I} = -\left(\boldsymbol{g}\cdot\boldsymbol{x}\right)\left(\boldsymbol{\nabla}\rho^{n+1}\right) + (\sigma\kappa)_f(\boldsymbol{\nabla}\alpha^{n+1}) \quad . \tag{B.12}$$

and in the TDF solver:

$$\boldsymbol{Q_D} = -\left(\boldsymbol{g}\cdot\boldsymbol{x}\right)\left(\boldsymbol{\nabla}\rho^{n+1}\right) \quad . \tag{B.13}$$

(5) **The PISO loop is solved N_{corr} times to obtain solutions for $\boldsymbol{U}^{n+1}$ and P_{rgh}^{n+1}:**

The PISO loop index $()^m$ is introduced in the following.

(5.1) **Computing pseudo-velocity:**

So-called pseudo-velocity is computed without the pressure and source terms contribution:

$$\tilde{\boldsymbol{U}} = \frac{\boldsymbol{H}(\boldsymbol{U}^m)}{a_P} \quad , \tag{B.14}$$

where $\boldsymbol{U}^m$ is the last known solution for velocity, *i.e.*

$*$ For the first PISO loop,

$\boldsymbol{U}^m = \boldsymbol{U}^*$ if step (2) (momentum predictor) has been performed,

$\boldsymbol{U}^m = \boldsymbol{U}^n$ otherwise,

$*$ Otherwise, solution $\boldsymbol{U}^m$ from step (5.5).

(5.2) **Computing pseudo-volumetric face flux:**

A pseudo-volumetric face flux is computed based on the pseudo-velocity $\tilde{\boldsymbol{U}}$ interpolated to face f.

$$\tilde{F}_u = \tilde{\boldsymbol{U}}_f \cdot \boldsymbol{S}_f \quad . \tag{B.15}$$

Then, the face interpolated source term multiplied by the surface is added to this flux:

$$\tilde{F} = \tilde{F}_u + \frac{|S|\boldsymbol{Q}_f}{(a_P)_f} \quad . \tag{B.16}$$

(5.3) **Pressure direct solution:**

The Poisson equation for pressure is solved in order to compute pressure at the iteration index m,

$$\sum_f \boldsymbol{S}_f \cdot \left[\left(\frac{1}{a_P} \right)_f \left(\boldsymbol{\nabla} P_{rgh}^m \right)_f \right] = \sum_f \tilde{F} \quad . \tag{B.17}$$

(5.4) **Volumetric face flux corrector:**

The pressure term is added to the volumetric face flux,

$$F^m = \tilde{F} - \boldsymbol{S}_f \cdot \left[\left(\frac{1}{a_P} \right)_f \left(\boldsymbol{\nabla} P_{rgh}^m \right)_f \right] \quad . \tag{B.18}$$

(5.5) **Velocity corrector:**

Finally, velocity is computed by summing pseudo-velocity, the pressure term at index m and the source term,

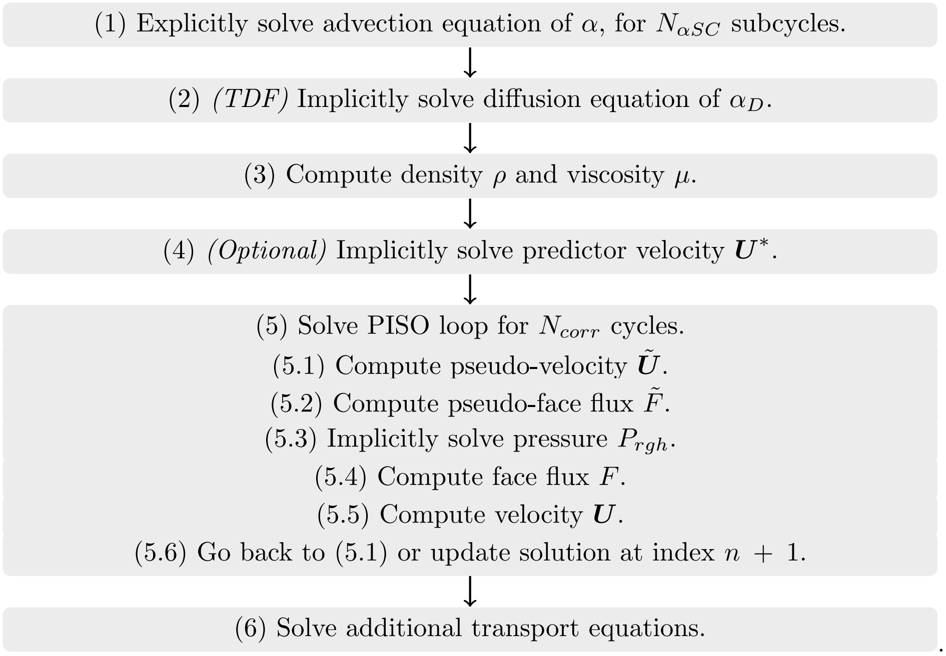
$$\boldsymbol{U}^m = \tilde{\boldsymbol{U}} + \frac{1}{a_P} \left[-\boldsymbol{\nabla} P_{rgh}^m + \boldsymbol{Q} \right] \quad . \tag{B.19}$$

(5.6) After N_{corr} loops in (5), velocity, face fluxes and dynamic pressure are finally updated,

$$\begin{aligned} \boldsymbol{U}^{n+1} &= \boldsymbol{U}^m \quad , \\ F^{n+1} &= F^m \quad , \\ P_{rgh}^{n+1} &= P_{rgh}^m \quad . \end{aligned} \tag{B.20}$$

(6) **Solving additional transport equations:** k^{n+1} and ε^{n+1} if it is turbulent and the $k - \varepsilon$ model is selected.

The RI and TDF single grid solver algorithms are synthesized in figure B.20, following similar stage numbering.



(1) Explicitly solve advection equation of α, for $N_{\alpha SC}$ subcycles.

(2) *(TDF)* Implicitly solve diffusion equation of α_D.

(3) Compute density ρ and viscosity μ.

(4) *(Optional)* Implicitly solve predictor velocity $\boldsymbol{U}^*$.

(5) Solve PISO loop for N_{corr} cycles.

(5.1) Compute pseudo-velocity $\tilde{\boldsymbol{U}}$.

(5.2) Compute pseudo-face flux $\tilde{F}$.

(5.3) Implicitly solve pressure P_{rgh}.

(5.4) Compute face flux F.

(5.5) Compute velocity $\boldsymbol{U}$.

(5.6) Go back to (5.1) or update solution at index $n + 1$.

(6) Solve additional transport equations.

Figure B.20: Algorithm structure of both single grid two-phase solvers.

References

[1] M. J. Berger, On conservation at grid interfaces, SIAM Journal on Numerical Analysis 24 (5) (1987) 967–984.

[2] A. S. Almgren, J. B. Bell, L. Howell, P. Colella, An adaptive projection method for the incompressible euler equations, Tech. rep., Lawrence Livermore National Lab., CA (United States) (1993).

[3] J. B. Bell, P. Colella, H. M. Glaz, A second-order projection method for the incompressible navier-stokes equations, Journal of Computational Physics 85 (2) (1989) 257–283.

[4] D. F. Martin, P. Colella, A cell-centered adaptive projection method for the incompressible euler equations, Journal of computational Physics 163 (2) (2000) 271–312.

[5] D. F. Martin, P. Colella, D. Graves, A cell-centered adaptive projection method for the incompressible navier–stokes equations in three dimensions, Journal of Computational Physics 227 (3) (2008) 1863–1886.

[6] Amrex, https://ccse.lbl.gov/amrex.
URL `https://ccse.lbl.gov/AMReX`

[7] M. M. Rai, A conservative treatment of zonal boundaries for euler equation calculations, Journal of Computational Physics 62 (2) (1986) 472–503.

[8] R. W. WALTERS, G. F. SWITZER, J. L. THOMAS, Aspects and applications of patched grid calculations, AIAA Journal 29 (5) (1991) 676–677. `doi:10.2514/3.10637`.

[9] H. Chen, S. Fu, F. Li, Navier-stokes simulations for transport aircraft wing/body high-lift configurations, Journal of Aircraft 40 (5) (2003) 883–890.

[10] Y. Zhang, H. Chen, S. Fu, Improvement to patched grid technique with high-order conservative remapping method, Journal of Aircraft 48 (3) (2011) 884.

[11] E. Rinaldi, P. Colonna, R. Pecnik, Flux-conserving treatment of non-conformal interfaces for finite-volume discretization of conservation laws, Computers & Fluids 120 (2015) 126–139.

[12] H. J. Aguerre, S. M. Damián, J. M. Gimenez, N. M. Nigro, Conservative handling of arbitrary non-conformal interfaces using an efficient supermesh, Journal of Computational Physics 335 (2017) 21–49.

[13] J. Benek, J. Steger, F. Dougherty, A flexible grid embedding technique, AIAA paper (1985) 85–1523.

[14] J. Benek, P. Buning, J. Steger, A 3-D chimera grid embedding technique, in: 7th Computational Physics Conference, Meeting Paper Archive, American Institute of Aeronautics and Astronautics, 1985.

[15] J. W. Kim, S. H. Park, Y. H. Yu, Euler and navier-stokes simulations of helicopter rotor blade in forward flight using an overlapped grid solver, in: 19th AIAA CFD Conference, 2009.

[16] J. Ekaterinaris, A. Cricelli, M. Platzer, A zonal method for unsteady, viscous, compressible airfoil flows, Journal of Fluids and Structures 8 (1) (1994) 107–123.

[17] S. R. Allmaras, A coupled euler/navier-stokes algorithm for 2-d unsteady transonic shock/boundary-layer interaction, Tech. rep., Cambridge, Mass.: Gas Turbine Laboratory, Massachusetts Institute of Technology,[1989] (1989).

[18] J. Tu, Computation of turbulent two-phase flow on overlapped grids, Numerical Heat Transfer 32 (2) (1997) 175–195.

[19] V.-T. Nguyen, D.-T. Vu, W.-G. Park, C.-M. Jung, Navier–stokes solver for water entry bodies with moving chimera grid method in 6dof motions, Computers & Fluids 140 (2016) 19–38.

[20] J. Li, A. M. Castro, P. M. Carrica, A pressure–velocity coupling approach for high void fraction free surface bubbly flows in overset curvilinear grids, International Journal for Numerical Methods in Fluids 79 (7) (2015) 343–369.

[21] Z. Shen, D. Wan, P. M. Carrica, Dynamic overset grids in openfoam with application to kcs self-propulsion and maneuvering, Ocean Engineering 108 (2015) 287–306.

[22] S. KHANNA, J. G. BRASSEUR, Analysis of moninobukhov similarity from large-eddy simulation, Journal of Fluid Mechanics 345 (1997) 251–286. doi:10.1017/S0022112097006277.
URL http://journals.cambridge.org/article_S0022112097006277

[23] M. Manhart, A zonal grid algorithm for {DNS} of turbulent boundary layers, Computers & fluids 33 (3) (2004) 435 – 461. doi:http://dx.doi.org/10.1016/S0045-7930(03)00061-6.
URL http://www.sciencedirect.com/science/article/pii/S0045793003000616

[24] F. Richez, I. Mary, V. Gleize, C. Basdevant, Near stall simulation of the flow around an airfoil using zonal rans/les coupling method, Computers & Fluids 37 (7) (2008) 857–866.

[25] G. W. Ley, R. L. Elsberry, Forecasts of typhoon irma using a nestedgrid model, Monthly Weather Review 104 (9) (1976) 1154–1161.

[26] Y. Kurihara, G. J. Tripoli, M. A. Bender, Design of a movable nested-mesh primitive equation model, Monthly Weather Review 107 (3) (1979) 239–249.

[27] P. P. Sullivan, J. C. McWilliams, C.-H. Moeng, A grid nesting method for large-eddy simulation of planetary boundary-layer flows, Boundary-Layer Meteorology 80 (1) (1996) 167–202. doi:10.1007/BF00119016.
URL http://dx.doi.org/10.1007/BF00119016

[28] R. Lebas, T. Menard, P.-A. Beau, A. Berlemont, F.-X. Demoulin, Numerical simulation of primary break-up and atomization: Dns and modelling study, International Journal of Multiphase Flow 35 (3) (2009) 247–260.

[29] C. W. Hirt, B. D. Nichols, Volume of fluid (vof) method for the dynamics of free boundaries, Journal of computational physics 39 (1) (1981) 201–225.

[30] H. Weller, A new approach to vof-based interface capturing methods for incompressible and compressible flow, OpenCFD Ltd., Report TR/HGW/04.

[31] H. Rusche, Computational fluid dynamics of dispersed two-phase flows at high phase fractions, Ph.D. thesis, Imperial College London (University of London) (2003).

[32] E. Berberović, N. P. van Hinsberg, S. Jakirlić, I. V. Roisman, C. Tropea, Drop impact onto a liquid layer of finite thickness: Dynamics of the cavity evolution, Physical Review E 79 (3) (2009) 036306.

[33] R. I. Issa, Solution of the implicitly discretised fluid flow equations by operator-splitting, Journal of computational physics 62 (1) (1986) 40–65.

[34] S. M. Damián, N. M. Nigro, An extended mixture model for the simultaneous treatment of small-scale and large-scale interfaces, International Journal for Numerical Methods in Fluids 75 (8) (2014) 547–574.

[35] J. H. Ferziger, M. Peric, Computational methods for fluid dynamics, Springer Science & Business Media, 2012.

[36] S.-R. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, L. Tobiska, Quantitative benchmark computations of two-dimensional

bubble dynamics, International Journal for Numerical Methods in Fluids 60 (11) (2009) 1259–1288.

[37] D. Stepowski, O. Werquin, Measurement of the liquid volume fraction and its statistical distribution in the near development field of a spray, Atomization and Sprays 14 (3).

[38] J. Klostermann, K. Schaake, R. Schwarze, Numerical simulation of a single rising bubble by vof with surface compression, International Journal for Numerical Methods in Fluids 71 (8) (2013) 960–982.

[39] F. Nicoud, F. Ducros, Subgrid-scale stress modelling based on the square of the velocity gradient tensor, Flow, turbulence and Combustion 62 (3) (1999) 183–200.

[40] M. Ishii, T. Hibiki, Thermo-fluid dynamics of two-phase flow, Springer Science & Business Media, 2010.

[41] F. Raees, D. Van der Heul, C. Vuik, Evaluation of the interface-capturing algorithm of OpenFoam for the simulation of incompressible immiscible two-phase flow, Delft University of Technology Delft, The Netherlands, 2011.

[42] A. Vallet, R. Borghi, Modélisation eulerienne de l'atomisation d'un jet liquide, Comptes Rendus de l'Académie des Sciences-Series IIB-Mechanics-Physics-Astronomy 327 (10) (1999) 1015–1020.

[43] S. M. Damián, An extended mixture model for the simultaneous treatment of short and long scale interfaces, Dissertationsschrift, Universidad Nacional del Litoral, Argentinien. Krepper, E., Lucas, D., Frank, T., Prasser, H.-M., & Zwart, PJ (2008). The inhomogeneous MUSIG model for the simulation of polydispersed flows. Nuclear Engineering and Design 238 (7) (2013) 1690–1702.