The 8th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2022
5-–9 June 2022, Oslo, Norway

NUMERICAL METHODS FOR DIFFUSE INTERFACE MULTIFLUID MODELS

CLEMENT LE TOUZE[1,⋆] and NICOLAS RUTARD[1]

[1] DMPE, ONERA, Université Paris Saclay, F-91123 Palaiseau - France,
{clement.le_touze,nicolas.rutard}@onera.fr

Key words: Multiphase Flows, Diffuse Interface Multifluid Models, Numerical Methods

Abstract. We describe some numerical methods developed in ONERA's Cedre platform to solve diffuse interface multifluid models with a view to realistic industrial applications. These methods are illustrated on test cases such as a shock-droplet interaction case.

1 INTRODUCTION

Multiphase flows are involved in many engineering applications and a large variety of scientific fields. For example, they are encountered in the aerospace industry within all kinds of reaction engines involving liquid reactants injected in the combustion chamber. The use of numerical simulations for design and optimization purposes is increasingly developing, and it is therefore crucial to develop more and more reliable and efficient numerical platforms. To simulate complex multiphase flows in realistic industrial applications, diffuse interface multifluid models are the best suited. In this family, the most general model is the 7-equation model [1, 2] allowing the fluids to be locally in full desequilibrium. From this model, it is possible to derive a hierarchy of models by successive relaxations considering infinitely small relaxation times. This gives rise to some well-known models such as the 4-equation model [3, 4], the 5-equation model [5], or the 6-equation model with immediate pressure relaxation which has gained much attention in the past decade [6–9].

The implementation of multifluid models in Cedre [10], the ONERA's CFD platform dedicated to industrial multiphysics applications (aerodynamics and energetics, heat and radiative transfers, turbulent combustion, two-phase flows), is a long-standing objective for ONERA. The challenge is twofold: first to equip these models with the physical submodels required to achieve a sufficient level of precision, which means e.g. including a proper modelling for fluid thermodynamics and phase change, turbulence or combustion. Second, to develop efficient, robust and accurate numerical methods to deal with the challenging conditions involved: complex geometries and general unstructured meshes, strong gradients of physical quantities, stiff relaxation and source terms, thermodynamics of phase change, interaction with turbulence and combustion, wide range of Mach numbers and large variety of spatial and temporal scales... This paper focuses on the second topic regarding the numerical methods. A compressive technique to reduce the numerical diffusion of the interface is implemented in the framework of multislope MUSCL methods. In order to deal with the whole range of Mach numbers, low-Mach corrections are also implemented in HLLC Riemann solvers. Using the 4-equation diffuse interface model in Cedre, these methods are illustrated on different test cases such as a shock-droplet interaction case.

2 THE 4-EQUATION DIFFUSE INTERFACE MODEL

2.1 Governing equations

The 4-equation diffuse interface model is very similar to the compressible Navier-Stokes equations for a multi-species homogeneous mixture, also referred to as the homogeneous relaxation model when mass transfer between phases is considered [11]. The system of conservation equations can be written in vector form as [3]:

$$\partial_t \boldsymbol{Q} \, + \, \nabla . \left(\boldsymbol{f_C} + \boldsymbol{f_D} + \boldsymbol{f_\sigma} \right) \, = \, \boldsymbol{S} \, , \tag{1}$$

where $\boldsymbol{Q} = (\rho Y_{j,k} \quad \rho \boldsymbol{u} \quad \rho E)^t$ is the vector of the conservative variables, comprising species partial densities (j is the index for species within the fluid k), momentum and total energy of the mixture, and ρ is the density of the whole mixture. From $\boldsymbol{Q}$, one can derive a set of primitive variables $\boldsymbol{U} = (Y_{j,k} \quad \boldsymbol{u} \quad P \quad T)^t$, i.e. the species mass fractions with respect to the mixture, plus the mixture mean velocity, pressure and temperature. The convective fluxes $\boldsymbol{f_C}$ are that of the standard compressible Euler equations, the diffusive fluxes $\boldsymbol{f_D}$ may include laminar and turbulent diffusion (either in a RANS or LES context), $\boldsymbol{f_\sigma}$ stands for surface tension (CSS method), while $\boldsymbol{S}$ comprises the source terms to be taken into account depending on the configuration (e.g. body forces, chemical reactions or coupling with another solver). The thermodynamic closure of the system is based on the assumption of thermal and mechanical equilibrium between phases [11], together with dedicated equations of state for each phase (see section 2.2 below). As a result, the liquid phase volume fraction α_l is not an independent transported variable of system (1), but can be straightforwardly obtained from the liquid mass fraction Y_l and the densities of pure fluids ρ_g and ρ_l as $\alpha_l = \rho Y_l / \rho_l = \rho_g Y_l / \left[\rho_g Y_l + \rho_l (1 - Y_l) \right]$.

2.2 Equations of state

A fluid made up of N_g gaseous species will be modeled most of the time as an ideal gas mixture, leading to

$$\rho = \frac{P}{RT} \left(\sum_{i=1}^{N_g} Y_i / W_i \right)^{-1} \, , \tag{2}$$

with R the ideal gas constant and W_i the molar mass of a given gaseous species i. For a fluid made up of one liquid species, our usual choice is to use a linearized version of the Isentropic Stiffened Gas EOS (ISG, also referred to as the Tait's law). Starting from the ISG law:

$$\frac{P + P_\infty}{P_0 + P_\infty} = \left(\frac{\rho}{\rho_0} \right)^\gamma \qquad \Leftrightarrow \qquad \rho = \rho_0 \left[1 + \gamma \beta_0 (P - P_0) \right]^{1/\gamma} \, , \tag{3}$$

where P_∞ is the Stiffened Gas constant of the liquid, $\beta_0 = \left[\gamma (P_0 + P_\infty) \right]^{-1}$ the isothermal compressibility at a given reference state (ρ_0, P_0) and γ the isentropic exponent. Then, the linearization of this law under the assumption $\beta_0 (P - P_0) \ll 1$ (see [12]) leads to:

$$\rho = \rho_0 \left[1 + \beta_0 (P - P_0) \right] \qquad \Leftrightarrow \qquad P = P_0 + c_0^2 \left(\rho - \rho_0 \right) \, , \tag{4}$$

where $c_0 = \left(\beta_0 \rho_0 \right)^{-1/2}$ is the constant sound speed of the liquid. This is a barotropic (isothermal) linearized equation of state, which is accurate in the limit of small variations around the reference state (ρ_0, P_0), and restricted to the pure liquid area of the phase diagram.

3 NUMERICAL METHODS

The numerical discretization in CEDRE is based on the finite volume method on general unstructured meshes. In this paper, we will focus on the discretization of convective fluxes, and in particular on the two following topics: a compressive limitation technique in the framework of the multislope MUSCL method to reduce the numerical diffusion of the interface in multifluid simulations, and low-Mach corrections for HLLC-type Riemann solvers.

3.1 The multislope MUSCL method for general unstructured meshes

To briefly introduce the multislope MUSCL method for general unstructured meshes (see further details and a visual representation on figure 1 in [13]), let us consider a domain Ω made up of general polyhedral elements. For each element $K_i \in \Omega$, $\mathcal{V}_i = \{K_j | K_i \cap K_j = S_{ij}\}$ is the set of its neighbor cells sharing a common face S_{ij}. For each face $S_{ij} \in \mathcal{V}_i$, we seek two pairs of two (in 2D) or three (in 3D) vertex-neighbor cells of K_i as close as possible to the axis connecting $\boldsymbol{B}_i$ and $\boldsymbol{M}_{ij}$ (the barycenters of K_i and S_{ij} respectively), so that we can compute backward and forward points $\boldsymbol{H}_{ij}^-$ and $\boldsymbol{H}_{ij}^+$. Then second-order consistent approximations of any reconstructed variable W can be computed at these points using barycentric interpolations, namely $W_{H_{ij}^-}$ and $W_{H_{ij}^+}$. The procedure is then very close to the classical MUSCL approach in one dimension. Both a backward and a forward slope are computed, and then used to compute the interpolation W_{ij} at the face barycenter:

$$p_{ij}^- = \frac{W_i - W_{H_{ij}^-}}{\|\boldsymbol{B}_i\boldsymbol{H}_{ij}^-\|}, \quad p_{ij}^+ = \frac{W_{H_{ij}^+} - W_i}{\|\boldsymbol{B}_i\boldsymbol{H}_{ij}^+\|}, \quad W_{ij} = W_i + p_{ij}^+ \, \varphi\left(r_{ij}, \eta_{ij}^-, \eta_{ij}^+\right) \|\boldsymbol{B}_i\boldsymbol{M}_{ij}\|, \tag{5}$$

where φ is a non-linear limiter function whose purpose is to provide the second-order accuracy in smooth regions while preserving the monotonicity near discontinuities. For the scheme stability (a proof of which is provided for the linear advection equation in [13]), this limiter function has to be expressed in terms of the slope ratio $r_{ij} = p_{ij}^-/p_{ij}^+$ but also in terms of geometrical parameters $\eta_{ij}^- = \|\boldsymbol{B}_i\boldsymbol{H}_{ij}^-\|/\|\boldsymbol{B}_i\boldsymbol{M}_{ij}\|$ and $\eta_{ij}^+ = \|\boldsymbol{B}_i\boldsymbol{H}_{ij}^+\|/\|\boldsymbol{B}_i\boldsymbol{M}_{ij}\|$.

This is the standard procedure for the interpolation of a given variable W. But in practice, when solving a particular system of equations, we also need to specify the set of variables on which to perform the interpolations. This is not a trivial issue and may have an impact on accuracy and robustness. For instance, it is well-known than interpolating primitive rather than conservative variables is a safer choice. Another issue of the same kind is how to interpolate vectorial variables, such as the velocity (this topic is addressed in another contribution to the 2022 ECCOMAS conference [14]). The result is that we may have to adopt a limitation procedure adapted to each variable and / or depending on the flow region. This is typically how the compressive limitation technique operates.

3.2 Compressive limitation in the multislope MUSCL framework

In spite of their advantages to deal with compressible multiphase flows, diffuse interface models also have the drawback to entail excessive numerical diffusion of interfaces when using classical second-order schemes. This is why so-called compressive limitation techniques have been proposed in the literature [15–17]. In this paper, we present such a technique adapted

to the multislope MUSCL method introduced in [13]. Inspired by [17], in which a cell-based interface sensor ξ_i was defined in a monoslope context using the volume fraction gradient, we introduce in the multislope context a face-based interface sensor ξ_{ij} as:

$$\forall K_i \in \Omega, \quad \forall S_{ij} \in \mathcal{V}_i, \quad \xi_{ij} = \begin{cases} \text{true} & \text{if} \quad ||\boldsymbol{B}_i\boldsymbol{H}_{ij}^+||p_{ij}^+ = \alpha_{H_{ij}^+} - \alpha_i > \epsilon_\alpha \\ \text{false} & \text{else} \end{cases} , \tag{6}$$

where ϵ_α is a threshold (typical value $\epsilon_\alpha = 10^{-3}$), and with α_i and $\alpha_{H_{ij}^+}$ the volume fractions at points $\boldsymbol{B}_i$ and $\boldsymbol{H}_{ij}^+$. Then we compute slope compression coefficients for each cell:

$$\gamma_i = \max\left[0, \min\left(\gamma_i^{min}, \gamma_i^{max}\right)\right], \; \gamma_i^{min} = \frac{\alpha_i - \min\left(\alpha_i, \epsilon\right)}{\max\left(\alpha_i - \alpha_i^{min}, \epsilon\right)}, \; \gamma_i^{max} = \frac{\max\left(\alpha_i, 1-\epsilon\right) - \alpha_i}{\max\left(\alpha_i^{max} - \alpha_i, \epsilon\right)}, \tag{7}$$

where $\epsilon \approx 10^{-20}$ and α_i^{min}, α_i^{max} are the minimum and maximum values of the volume fraction in the vertex-based neighborhood of the cell K_i:

$$\begin{array}{rclcrcl} \alpha_i^{min} & = & \min\left(\alpha_i, \min\limits_{j\in\mathcal{V}_i} \alpha_{ij}^{min}\right) & \text{with:} & \alpha_{ij}^{min} & = & \min\left(\alpha_{H_{ij}^-}, \alpha_{H_{ij}^+}, \alpha_{H_{ji}^-}, \alpha_{H_{ji}^+}\right) \\ \alpha_i^{max} & = & \max\left(\alpha_i, \max\limits_{j\in\mathcal{V}_i} \alpha_{ij}^{max}\right) & & \alpha_{ij}^{max} & = & \max\left(\alpha_{H_{ij}^-}, \alpha_{H_{ij}^+}, \alpha_{H_{ji}^-}, \alpha_{H_{ji}^+}\right) \end{array} . \tag{8}$$

This enables us to perform the slope compression as follows:

$$\forall K_i \in \Omega, \quad \forall S_{ij} \in \mathcal{V}_i \mid \xi_{ij} = \text{true}, \quad \overline{p_{ij}^-} = \gamma_i p_{ij}^- , \quad \overline{p_{ij}^+} = \gamma_i p_{ij}^+ , \tag{9}$$

where $\overline{p_{ij}^-}$ and $\overline{p_{ij}^+}$ are the compressed slopes for the volume fraction. The next step afterwards is to adapt the limitation procedure for every face, depending on whether it belongs to the interface or not. For the faces localized outside the interface ($\xi_{ij} = \text{false}$), we just interpolate all the usual variables with our favorite standard limiter. For the faces localized inside the interface on the other hand ($\xi_{ij} = \text{true}$), the procedure is as follows: first interpolate the volume fractions α_{ij} with a hyper-compressive limiter (see below), then interpolate the other variables with a standard limiter, and finally update the variables required to compute the numerical fluxes from the interpolated variables. The final step is to derive a hyper-compressive limiter dedicated to the interpolation of the volume fraction within the interface. To do so, we use a so-called Overbee limiter adapted to the multislope formalism:

$$\varphi(r_{ij}, \eta_{ij}^-, \eta_{ij}^+) = \max\left[0, \min\left(\eta_{ij}^- r_{ij}, \eta_{ij}^+\right)\right] . \tag{10}$$

This limiter matches the boundaries of the stability zone without ensuring the second-order consistency property ($\phi(r) = 1$). As a result, it tends to transform the solutions that vary linearly into discontinuities, which is precisely the behavior we seek here in order to maintain a discontinuous profile for the volume fraction as long as possible. With this limiter, interpolations of the volume fractions can be recast as:

$$\alpha_{ij} = \alpha_i + \left(\alpha_{H_{ij}^+} - \alpha_i\right) \max\left(0, \min\left[1, (\alpha_i - \alpha_{H_{ij}^-})/(\alpha_{H_{ij}^+} - \alpha_i)\right]\right) , \tag{11}$$

that is to say as a convex combination of α_i and $\alpha_{H_{ij}^+}$, thereby preserving the domain of definition of the volume fraction.

The effect of this compressive limitation technique is clearly shown in figure 1, which represents a classical sloshing test case under gravity performed with the 4-equation model. It can be seen that the liquid-gas interface undergoes a very strong numerical diffusion with classical second-order schemes, whereas the interface is much more preserved thanks to the limitation technique.

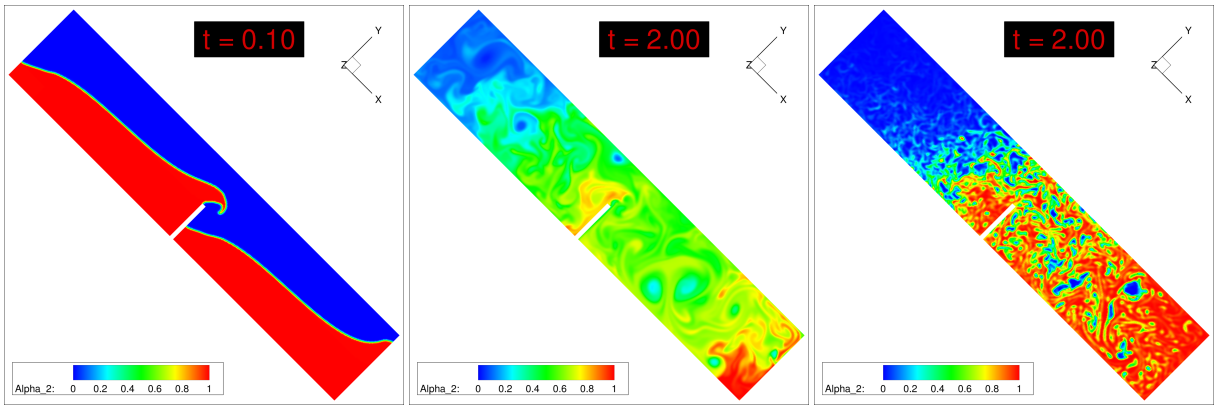


Figure 1: Effect of the compressive limitation technique on a water tank sloshing test case under gravity with the 4-equation model. Left: volume fraction field just after the initial time. Middle: final solution with a standard second-order scheme. Right: final solution with the compressive method.

3.3 Low-Mach number corrections in HLLC Riemann solvers

An important feature of multiphase flows in energetics is that they may involve a wide scale of Mach numbers in a same application, ranging from low-Mach number (or even zero) for example in steady liquid regions, to supersonic or even hypersonic areas. For these applications, the use of incompressible models is therefore impossible, while the use of compressible models can be a challenge due to their well-known inaccuracy in the low-Mach number limit [18]. This may result in excessive numerical diffusion of material interfaces moving at low speed, but also in strong pressure oscillations, especially when a weakly compressible liquid phase is involved. Much work has been done in the literature to deal with this low-Mach issue in compressible multifluid models. Starting from a reformulation of HLLC Riemann solvers proposed in an unpublished ONERA report by B. Courbet, we introduce here low-Mach corrections as additional terms to the standard upwinding, thus simplifying the implementation of an all-Mach HLLC scheme in a CFD code. Considering the Euler equations without loss of generality, the classical HLLC Riemann solver is rewritten as:

$$F_{HLLC} = \frac{1}{2}(F_L + F_R) + \frac{1}{4}(\sigma_R + \sigma_L)(F_L - F_R) + \frac{1}{4}(\sigma_R - \sigma_L)D, \tag{12}$$

where D is the subsonic upwinding ($D = 0$ in the supersonic case) that reads:

$$D = (F_L^* - F_L) + (F_R^* - F_R) + \sigma_M(F_L^* - F_R^*) = \left[\begin{array}{ccc} D_{\rho Y} & D_{\rho \boldsymbol{u}} & D_{\rho E} \end{array} \right]^t , \tag{13}$$

with F_L, F_L^*, F_R^* and F_R the physical fluxes considered in the different regions of the Riemann problem, and σ_L, σ_R, σ_M the sign of the three waves speeds S_L, S_R, S_M. The subsonic upwinding can be split into convective vs. pressure force terms, i.e. $D = D^c + D^p$ with:

$$D^p = \ \ldots \ + \boxed{(f_R - f_L) \begin{bmatrix} 0 \\ \boldsymbol{n} \\ S_M \end{bmatrix}},$$

$f_L = \rho_L(u_{n,L} - S_L)(S_M - u_{n,L})$ and $f_R = \rho_R(S_R - u_{n,R})(S_M - u_{n,R})$ standing for pressure forces (with $u_{n,L}$ and $u_{n,R}$ the left and right normal velocities), while $\boldsymbol{n}$ is the unit vector normal to the face. The term in the box is actually the one to blame for the over dissipation and inaccuracies at low-Mach. In order to compensate for it, we add a contribution D^{sup} to the standard upwind term D:

$$D^{sup} = (1-\beta)\left[\rho_L^* S_L(u_{n,L} - S_M) - \rho_R^* S_R(S_M - u_{n,R})\right] \begin{bmatrix} 0 \\ \boldsymbol{n} \\ S_M \end{bmatrix},$$

where ρ_L^* and ρ_R^* are the densities in the left and right starred region of the Riemann problem. The important part is β, a function of the Mach number at least, and whose role is to tend to unity outside of the low-Mach regime to recover the classical HLLC scheme. Most of low-Mach corrections proposed in the literature can be recast in this way, with particular expressions for the function β. Here we employ an ad hoc formulation proposed by D. Scherrer at ONERA and depending on the left and right Mach numbers, sound speeds, densities and pressures:

$$\beta = \min\left[1, M_L + M_R + \frac{|P_L - P_R|}{\rho_L c_L^2 + \rho_R c_R^2 + \epsilon} + 10^{-2}\right], \quad \epsilon = 10^{-10}. \tag{14}$$

Figure 2 shows the contribution of this low-Mach correction on a simulation performed with the 4-equation model (equipped with a $k-\omega$ RANS turbulence model) during the internship project of W. Haegeman. It consists of a high-velocity gas jet impacting a free surface of liquid water initially at rest. Without the low-Mach correction, spurious pressure fluctuations of high amplitude are triggered just below the liquid surface as soon as the pressure wave generated by the gas jet impacts on it (even before the impact of the jet itself), which eventually lead to the failure of the computation. In this very low-Mach liquid zone, the misbehavior of the HLLC Riemann solver is more than a dissipation issue, it is primarily a robustness issue. With the low-Mach correction detailed above, these pressure shoots are avoided and the simulation can be carried out until convergence.

4 FOCUS ON A SHOCK-DROPLET INTERACTION TEST-CASE

4.1 Description of the case

Similar test cases have been widely studied in the literature (see [19–21] for example) and aim at reproducing the deformation and breakup of a water droplet (or water column [19]) exposed to a supersonic flow. For the validation of the numerical strategy presented in this paper, both experimental and numerical studies are used as reference [21]. This test case consists in a

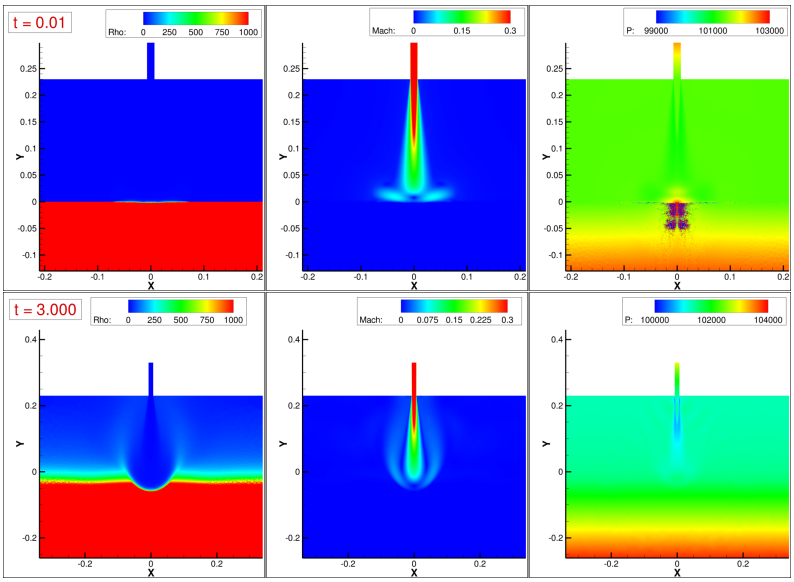
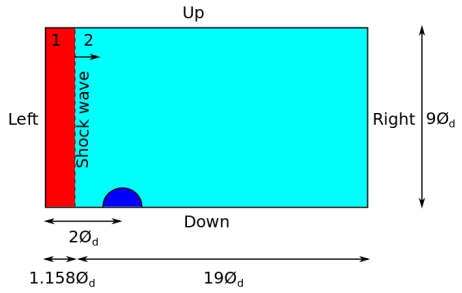


Figure 2: Top: solution at $t = 0.01$ s without low-Mach correction, a few time before the computation fails due to spurious pressure fluctuations. Bottom: converged solution at $t = 3$ s obtained with the low-Mach correction (slightly zoomed out). From left to right: density, Mach number and pressure fields.

spherical water droplet of diameter $\varnothing_d = 1.135$ mm located in ambiant air in which propagates a planar shock wave. In order to reduce computational cost, a 2D axisymmetrical geometry is simulated. The numerical domain and the initial state of the simulation are described in figure 3. Red values represent input data extracted from the reference [21], while other data are computed mostly thanks to Rankine-Hugoniot jump conditions.



Quantity	Zone 1	Zone 2	Droplet	Shock
ρ [kg.m^{-3}]	5.535	1.179	1e3	-
P [Pa]	2.122334e6	1.02e5	1e5	-
T [K]	1336	301	278	-
V [m.s^{-1}]	1160	0	0	1474
c [m.s^{-1}]	733	348	1415	-
Ma [-]	1.58	0	0	4.24

Figure 3: Geometry and initialization data for the simulation of shock-droplet interaction.

The numerical mesh consists of two zones of quadrangles with two different characteristic sizes. The first zone corresponds to the lower part of the domain and extends up to $3\varnothing_d$ from the symmetry axis. In this zone, a uniform cell characteristic size of 90 cells per initial droplet diameter is used. Due to the high Weber ("catastrophic") regime considered in this application ($We \sim 1.2e5$), such refinement is clearly not sufficient to capture all final liquid fragments but is considered to be adapted for a macroscopic analysis of the early droplet breakup dynamics. In the second zone, which corresponds to the rest of the domain, the mesh is progressively coarsened

in vertical direction down to a cell characteristic size of 30 cells per initial droplet diameter at the upper boundary condition in order to reduce computational cost. Besides, viscous stresses, heat diffusion and capillary effects are neglected since such mechanisms do not participate to the early droplet breakup [21]. Finally, the left and right boundary conditions are supersonic air inflow ($P = 2.122\,334 \times 10^6$ Pa, $T = 1336$ K and $V = 1160$ m.s^{-1}) and outflow respectively, while the up and down boundary conditions correspond to a wall and the symmetry axis. In the following section, the initial droplet diameter $\varnothing_d$ and Rayleigh time $t_c = \frac{\varnothing_d}{V_1}\sqrt{\frac{\rho_d}{\rho_1}}$ are used to get dimensionless space and time quantities, with ρ_1 and V_1 the upstream air density and velocity, and ρ_d the water droplet density.

4.2 Results with the 4-equation model

Figure 4 represents the liquid volume fraction and numerical Schlieren fields at different times after the shock impacts the droplet. In this figure, the strongest air density gradients extracted from the numerical simulation of reference [21] (red dots) are used to visualize bow and wake shocks positions as well as parts of the liquid-gas interface. This first comparison tends to demonstrate the capability of the 4-equation model to reproduce the displacement and deformation of bow and wake shocks as well as the droplet breakup dynamics. To demonstrate that the numerical methods presented in section 3 play a major role in the reproduction of the shock-droplet interaction, the same analysis is performed in figure 5 for simulations carried out either without compressive limitation (left) or without low-Mach number correction (right). Without compressive simulation, an excessive diffusion of the interface is noticed at the droplet stagnation point, leading to a significant error on the bow shock location. Regarding the low-Mach number correction, some may think that no effect should be observed in such a test case due to the overall supersonic nature of the flow. However, as its name suggests, the droplet stagnation point is associated with a very low Mach number. At this particular location, the HLLC scheme thus leads to an excessive diffusion of the solution, which results once again in a significant error on the bow shock location. This demonstrates that both compressive limitation and low-Mach number correction participate to enhance the precision of such multifluid simulations.

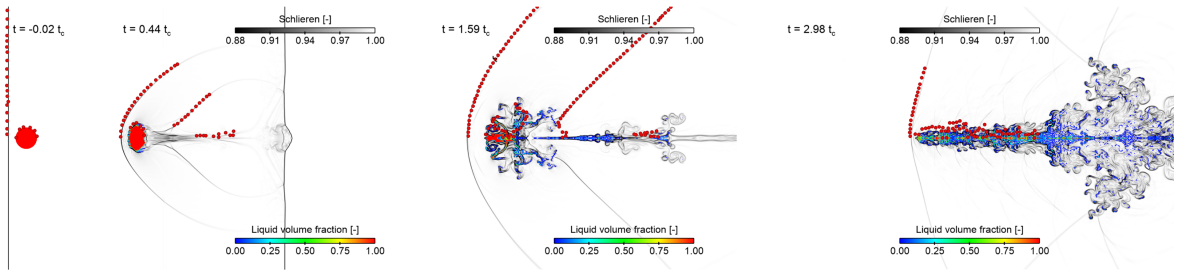


Figure 4: Liquid volume fraction and Schlieren fields at different times. Red dots: strongest air density gradients from reference simulation [21].

Further analyses are performed hereafter to pursue the validation of the multifluid strategy in Cedre. First, displacements of the droplet center of mass (DCM) and the droplet stagnation point (DSP) are measured. As mentioned in the literature, the DCM displacement can be used

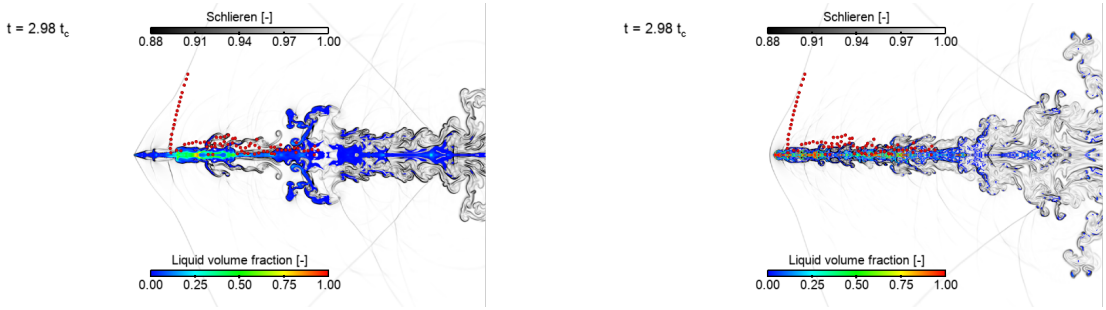


Figure 5: Liquid volume fraction and Schlieren fields after droplet breakup ($t = 2.98t_c$). Red dots: strongest air density gradients from reference simulation [21]. Left: without compressive limitation. Right: without low-Mach number correction.

to calculate the drag coefficient of the deforming droplet, which is an important feature to model the liquid acceleration [19], while the DSP displacement is a key feature to evaluate the bow shock position in time. The DCM coordinates $\boldsymbol{X}_{DCM}$ can be calculated in numerical simulations as $\boldsymbol{X}_{DCM} = \frac{\int_\Omega \alpha_l \rho_l \boldsymbol{X} dV}{\int_\Omega \alpha_l \rho_l dV}$, where the integrated volume Ω is the entire computational domain, $\boldsymbol{X}$ corresponds to each computational cell coordinates and dV is their volume. As shown in figure 6, a good agreement is found between reference and CEDRE results as far as it concerns DCM displacement. Regarding DSP, measurements are much more uncertain with a diffuse interface method than with the Volume Of Fluid method used in reference [21]. Indeed, diffuse interface models do not involve any interface reconstruction, leading to an ambiguity in the exact interface location and thus in the DSP position. Therefore, two measurements of the DSP displacement are performed and displayed on left part of figure 6 with two different threshold liquid volume fractions α_{th} *viz.* 0 and 0.1 (any mesh cell with a liquid volume fraction $\alpha_l \geq \alpha_{th}$ is considered part of the liquid structures). According to figure 6, the threshold value 0.1 seems to be a better tracer of the interface since it leads to a perfect match with the reference DSP. The time evolution of bow shock position along the symmetry axis plotted on the middle part of figure 6 tends to confirm this conclusion. Indeed, as mentioned before, the DSP and bow shock displacements are strongly related. Finally, right part of figure 6 shows the evolution of the dimensionless distance between bow and wake shocks at $2.5\varnothing_d$ from the symmetry axis. A good agreement is found with numerical results of reference, with however slight differences whose magnitude is the same as between experimental and numerical results of reference. This last result participates to demonstrate the good reproduction of the macroscopic droplet breakup dynamics since the position of the wake shock strongly depends on the formation of vortices in the droplet's wake [19]. All these results show that thanks to the numerical methods presented in section 3, the 4-equation model is able to reproduce complex breakup mechanisms in an environment experiencing a wide range of Mach numbers, from locally sub- to supersonic flows.

5 TOWARDS THE USE OF A TWO-TEMPERATURE MODEL IN CEDRE

Being a mirror of the compressible Navier-Stokes equations, the 4-equation model can be easily implemented in advanced industrial CFD codes like CEDRE, which makes it attractive. Indeed, these codes more easily include the whole complexity required for industrial applica-

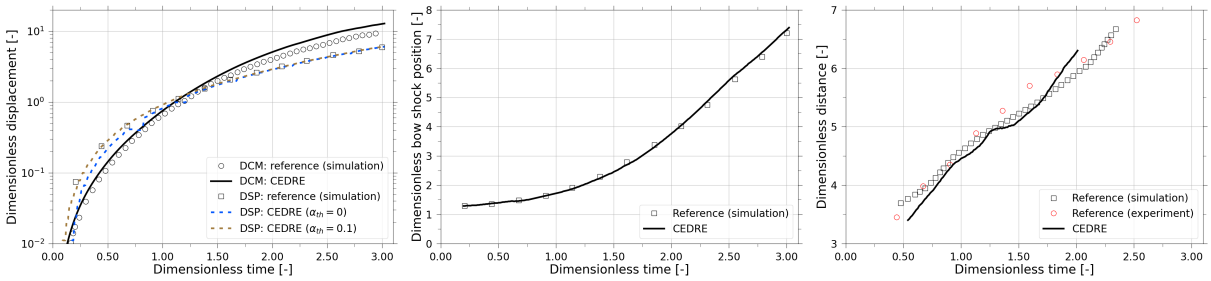


Figure 6: Left: DCM and DSP displacements along the shock propagation axis. Middle: bow shock position along the symmetry axis. Right: distance between bow and wake shocks at $2.5\varnothing_d$ from the symmetry axis. Reference results: see [21]

tions (e.g. unstructured meshes and advanced numerical methods, general thermodynamics or comprehensive physical modelling) than specialized codes. Nevertheless, the use of more refined multifluid models, giving locally access e.g. to the temperature and the velocity of each fluid, as well as to some geometrical variables describing the interface topology at the subgrid scale level, seems unavoidable if we want to progress towards more predictive multifluid simulations for industrial applications. This is why we are interested in bringing a more accurate multifluid model, namely the 6-equation model with instantaneous pressure relaxation, to industrial maturity. This model has been first introduced by [6], and then revisited by some authors [7–9]. In this model, each fluid is locally described with its own temperature. This entails a more robust thermodynamic closure than the 4-equation model when strong temperature gradients between fluids are involved [22], and offers a more solid basis to model heat transfers and phase changes. The version of this model that has been implememented in Cedre is the one derived during the PhD of P. Cordesse [9], carried out at ONERA in collaboration with M. Massot's team at CMAP - Ecole Polytechnique. The comprehensive version of this model includes transported equations on subscale geometrical variables (surface density area, mean curvature) derived from the principle of least action, accounting in particular for capillary effects at the subscale level. Without considering here the geometrical variables equations for the sake of simplicity, this system reads (for the case of two fluids, $k = 1, 2$):

$$\partial_t \boldsymbol{Q} + \nabla \cdot (\boldsymbol{f_C} + \boldsymbol{f_D} + \boldsymbol{f_\sigma}) + \boldsymbol{S_D} + \boldsymbol{K} \nabla \cdot \boldsymbol{u} = \frac{\boldsymbol{R}}{\epsilon^p} + \boldsymbol{S}\,, \tag{15}$$

where $\boldsymbol{Q} = (\alpha_k \rho_k y_{j,k} \;\; \rho \boldsymbol{u} \;\; \rho E \;\; \alpha_k \rho_k \epsilon_k \;\; \alpha_1)^t$ is the vector of conserved or transported variables, comprising species partial densities of each fluid (j is the index for species within the fluid k), momentum and total energy of the whole mixture, plus internal energies and volume fractions of each fluid. From $\boldsymbol{Q}$, one can derive a set of primitive variables, e.g. $\boldsymbol{U} = (\boldsymbol{u} \;\; \alpha_k \;\; y_{j,k} \;\; P_k \;\; T_k)^t$, i.e. the mixture mean velocity, plus for each fluid the volume fraction, the mass fractions of its species (with respect to the fluid), the pressure and the temperature. As each fluid is described with its own T_k and P_k (before applying the instantaneous pressure relaxation), this makes the thermodynamic closure independant for each fluid:

$$\epsilon_k = \epsilon_k^{EOS}\left(P_k, T_k, y_{j,k}\right)\,, \quad P_k = P_k^{EOS}\left(\rho_k, \epsilon_k, y_{j,k}\right)\,, \quad T_k = T_k^{EOS}\left(\rho_k, \epsilon_k, y_{j,k}\right)\,, \quad \text{etc.} \tag{16}$$

It is therefore possible to use any equation of state for each fluid (see section 2.2). A mixture pressure is defined as $P = \sum \alpha_k P_k$, and becomes the unique pressure of the fluids as soon as the pressure relaxation is applied. Moreover, $\boldsymbol{f_C}$ and $\boldsymbol{K}\nabla \cdot \boldsymbol{u}$ are respectively the conservative and non conservative parts of convective fluxes, $\boldsymbol{f_D}$ and $\boldsymbol{S_D}$ comprise the diffusive terms (see [23]), $\boldsymbol{f_\sigma}$ stands for surface tension, $\frac{\boldsymbol{R}}{\epsilon^p}$ is the instantaneous pressure relaxation term ($\epsilon^p \to 0$), while $\boldsymbol{S}$ gathers the source terms, e.g. body forces such as gravity. Under the assumption of instantaneous pressure relaxation, the solutions of the 6-equation model converge towards that of the classical 5-equation model [5]. Even though the equation on the total energy of the mixture is redundant with the transport equations on internal energies, it is still solved numerically in order to ensure the conservation of the total energy. Indeed, the latter is not guaranteed due to the numerical discretization of the non conservative terms in the internal energies equations [6].

The numerical discretization of this 6-equation model in Cedre is made in the same way as for the 4-equation model. In particular, both numerical methods described in this paper (compressive limitation and low-Mach corrections) have been easily adapted to the 6-equation model. However, while the 4-equation model in Cedre has reached the level of maturity required for numerical simulations on industrial configurations (it has already been used for many complex applications in various fields, such as liquid atomization in jet or rocket engines, supercritical combustion, phase change in heat exchangers, spray / acoustics interaction with space launch vehicles, etc.), the simulations with the 6-equation model have not yet achieved the same level of maturity in Cedre. The main difficulty currently occurs when shocks interact with the diffuse interface, like e.g. in the shock-droplet interaction test case presented in section 4. The numerical discretization of the non conservative terms in the volume fraction and internal energy equations may lead to small discrepancies when updating the fluid densities in the shock layer, resulting in strong discrepancies when updating the fluid pressures due to the weak compressibility of the liquid. This occurs after the convective step and before the pressure relaxation procedure. To ensure the robustness of the computation, especially when dealing with general EOS, it may be necessary to use clipping patches, but then at the expense of a detrimental loss of conservativity. These difficulties seem to be kwown in the literature, but the potential solutions such as the artificial heat exchange mentioned by [6] are not compatible with practical simulations on an industrial scale. We currently envision different ways to remedy this problem, e.g. using an alternative equation of state for the liquid phase, less sensitive to the pressure and density discrepancies. Otherwise it will be necessary to further study the numerical discretization methods employed, or maybe consider an alternative version of the model such as the 6-equation model of [7] based on the total energies of the fluids.

Apart from this shock / interface interaction issue, satisfactory simulations with the 6-equation model have been performed on several cases, such as those presented in section 3 with the 4-equation model. In particular, challenging simulations of high-temperature and high-speed gas jets impacting water surfaces have been conducted with the 6-equation model, exhibiting good accuracy and robustness, even better than with the 4-equation model due to the strong temperature gradients involved. When these numerical robustness issues are solved, our main focus will be on the formulation of advanced physical submodels for the 6-equation model (subscale geometrical variables, general thermodynamics and phase change, turbulence and combustion...).

REFERENCES

[1] Baer, M. R. & Nunziato, J. W. *Journal of Multiphase flows* (1986). 12:861–889.

[2] Saurel, R. & Abgrall, R. *Journal of Computational Physics* (1999). 150(2):425–467. ISSN 00219991. doi:10.1006/jcph.1999.6187.

[3] Le Touze, C. et al. *Applied Mathematical Modelling* (2020). 84:265–286. ISSN 0307-904X. doi:10.1016/j.apm.2020.03.028.

[4] Rutard, N. et al. *International Journal of Multiphase Flow* (2020). 122:103144. ISSN 03019322. doi:10.1016/j.ijmultiphaseflow.2019.103144.

[5] Kapila, A. K. et al. *Physics of Fluids* (2001). 13(10):3002–3024. doi:10.1063/1.1398042.

[6] Saurel, R. et al. *Journal of Computational Physics* (2009). 228(5):1678–1712. ISSN 00219991. doi:10.1016/j.jcp.2008.11.002.

[7] Pelanti, M. & Shyue, K.-M. *Journal of Computational Physics* (2014). 259:331–357. ISSN 0021-9991. doi:10.1016/j.jcp.2013.12.003.

[8] Schmidmayer, K. et al. *Journal of Computational Physics* (2017). 334:468–496. ISSN 0021-9991. doi:10.1016/j.jcp.2017.01.001.

[9] Cordesse, P. Ph.D. thesis, Université Paris-Saclay (2020). https://tel.archives-ouvertes.fr/tel-02948195.

[10] Refloch, A. et al. *AerospaceLab Journal* (2011). 2:131–140.

[11] Goncalvès, E. *International Journal of Heat and Mass Transfer* (2014). 76(Supplement C):247 – 262. ISSN 0017-9310. doi:https://doi.org/10.1016/j.ijheatmasstransfer.2014.04.065.

[12] Ghidaglia, J.-M. & Mrabet, A. A. *Journal of Applied Analysis and Computation* (2018). 8(3):675–689. doi:10.11948/2018.675.

[13] Le Touze, C. et al. *Journal of Computational Physics* (2015). 284:389–418. ISSN 00219991. doi:10.1016/j.jcp.2014.12.032.

[14] Tételin, A. et al. In *ECCOMAS 2022*. Oslo (2022) .

[15] Chiapolino, A. et al. *Journal of Computational Physics* (2017). 340:389–417. ISSN 0021-9991. doi:10.1016/j.jcp.2017.03.042.

[16] Deng, X. et al. *Journal of Computational Physics* (2018). 371:945–966. ISSN 0021-9991. doi:10.1016/j.jcp.2018.03.036.

[17] Blanchard, G. Ph.D. thesis, Institut Supérieur de l'Aéronautique et de l'Espace (ISAE) (2014). https://hal.archives-ouvertes.fr/tel-01171809.

[18] Guillard, H. & Viozat, C. *Computers & Fluids* (1999). 28(1):63–86. ISSN 00457930. doi:10.1016/S0045-7930(98)00017-6.

[19] Meng, J. C. & Colonius, T. *Shock Waves* (2015). 25(4):399–414. ISSN 0938-1287, 1432-2153. doi:10.1007/s00193-014-0546-z.

[20] Meng, J. C. & Colonius, T. *Journal of Fluid Mechanics* (2018). 835:1108–1135. ISSN 0022-1120, 1469-7645. doi:10.1017/jfm.2017.804.

[21] Hébert, D. et al. *SN Applied Sciences* (2020). 2(1):69. ISSN 2523-3963, 2523-3971. doi:10.1007/s42452-019-1843-z.

[22] Le Touze, C. Ph.D. thesis, Université Nice Sophia Antipolis (2015). https://tel.archives-ouvertes.fr/tel-01250527.

[23] Schmidmayer, K. et al. In *46th AIAA Fluid Dynamics Conference*, AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics (2016). doi:10.2514/6.2016-4269.