applied sciences

MDPI

Article

Sharp Interface Capturing in Compressible Multi-Material Flows with a Diffuse Interface Method

Shambhavi Nandan [1,*], Christophe Fochesato [1], Mathieu Peybernes [1,*], Renaud Motte [2] and Florian De Vuyst [3]

[1] CEA, DES, IRESNE, DTN, Cadarache, 13108 Saint-Paul-lez-Durance, France; christophe.fochesato@cea.fr
[2] CEA, DAM, DIF, 91297 Arpajon, France; Renaud.MOTTE@cea.fr
[3] Laboratoire de Mathématiques Appliquées de Compiègne (EA 2222), Université de Technologie de Compiègne, Sorbonne-Université, CEDEX, 60203 Compiègne, France; florian.de-vuyst@utc.fr
* Correspondence: shambhavi.nandan@cea.fr (S.N.); mathieu.peybernes@epfl.ch (M.P.)

Abstract: Compressible multi-materialflows are encountered in a wide range of natural phenomena and industrial applications, such as supernova explosions in space, high speed flows in jet and rocket propulsion, underwater explosions, and vapor explosions in post accidental situations in nuclear reactors. In the numerical simulations of these flows, interfaces play a crucial role. A poor numerical resolution of the interfaces could make it difficult to account for the physics, such as material separation, location of the shocks and contact discontinuities, and transfer of the mass, momentum and heat between different materials/phases. Owing to such importance, sharp interface capturing remains an active area of research in the field of computational physics. To address this problem in this paper we focus on the *Interface Capturing (IC)* strategy, and thus we make use of a newly developed *Diffuse Interface Method (DIM)* called *Multidimensional Limiting Process-Upper Bound (MLP-UB)*. Our analysis shows that this method is easy to implement, can deal with any number of material interfaces, and produces sharp, shape-preserving interfaces, along with their accurate interaction with the shocks. Numerical experiments show good results even with the use of coarse meshes.

Keywords: multi-material compressible flows; *Diffuse Interface Method (DIM)*; MUSCL reconstruction; compressive-shape preserving limiter; gradient reconstruction; MLP-UB method

check for updates

Citation: Nandan, S.; Fochesato, C.; Peybernes, M.; Motte, R.; De Vuyst, F. Sharp Interface Capturing in Compressible Multi-Material Flows with a Diffuse Interface Method. *Appl. Sci.* **2021**, *11*, 12107. https://doi.org/10.3390/app112412107

Academic Editor: Philip A. Rubini

Received: 10 November 2021
Accepted: 15 December 2021
Published: 19 December 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Copyright: © 2021 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

Multi-phase compressible flows are present in a wide and ever expanding range of natural and industrial applications. Among them we can mention supernova explosions [1], high speed flows in jet and rocket propulsion [2], underwater explosions [3,4], and the scenario of vapor explosions in post accidental situations in nuclear reactors [5,6]. To compute these flows, two main families of numerical methods are used: Lagrangian and Eulerian. In the Lagrangian approach, the objective is to describe the phenomena that govern the state of an individual fluid element, while in the Eulerian approach, the objective is to describe the phenomena that govern the state of the fluid in a fixed volume element. In these modeling approaches, the main concerns are the treatment of the interfaces and the phenomena associated with them.

In the case of a Lagrangian approach, the interfaces are followed by moving and deforming the mesh [7,8]. For the Eulerian approach, it is possible to capture or reconstruct the interface by adding a dedicated equation. This is the approach adopted in the methods such as Volume Of Fluid (VOF) [9–11] and Level Set [12,13]. Front tracking methods [14,15] also exist, coupling some advantages of the Lagrangian and Eulerian representations. These methods consider the interface as a local discontinuity. However, algorithms dedicated for the determination of the position of interfaces are often expensive. Moreover, strong topology changes and formation/disappearance of the interfaces are particularly complex to deal with, especially with the increase in the number of phases present.

There is another class of Eulerian methods called *algebraic* methods, which belong to the class of sharp Interface Capturing (IC) approach [16] . These methods are often referred to as *Diffuse Interface Methods (DIM)* [17], an exhaustive up-to-date review of which can be found in [18]. In this family of methods, the diffusion of the interface is allowed. The interfaces are not subjected to any particular geometrical treatment, which makes their implementation easier. Moreover, these methods can handle the dynamic formation, disappearance and large deformation of the interfaces in flows with several phases. The propagation of the acoustic waves can also be naturally taken into account. However, one of the drawbacks of these methods is the smearing of the already diffused interfaces, due to the numerical diffusion. Significant smearing of the interfaces can then result in an extended zone of a pure numerical (non-physical) mixture of the different materials in the vicinity of the interfaces. Thus, the accurate description of any physical phenomena at the interface may be deteriorated by this wide diffusion zone.

For our own end-use applications, we have decided to focus in this paper on the IC approach to solve the multi-material compressible problem. To ensure the sharp interface capturing in the solution of a compressible multi-phase model, a second-order multidimensional MUSCL [19] reconstruction is used with a recently developed compressive, shape-preserving gradient limiter, referred to as *Multidimensional Limiting Process–Upper Bound (MLP–UB)* [18], which is an updated variant of the classical *MLP* approach demonstrated in [20]. Sharp interface capturing methods are generally Finite Volume (FV) methods, either based on the high order flux evaluations or on the slope reconstruction, but in all the cases they make use of the limiters (flux or slope). Besides the MLP–UB approach adopted here, other popular methods/schemes found in the literature are: limited downwind scheme [21], THINC scheme [22] or the method using correction algorithms [23].

The one-pressure, compressible system of the continuous equations considered in this work is focused on the phase volume fractions together with the evolution of the mean mass, momentum and energy of each phase [24–26]. This system is a well-known problem allowing the assessment of the good behavior of our IC approach together with the use of a compressive, shape-preserving gradient limiter for an arbitrary number of phases. This study is the sequel to the work done in [18], which was restricted to the cases modeled with a pure advection equation. Thus, the starting point model proposed in this paper is intended to evolve using the appropriate closure terms in order to generate the models of industrial relevance. These added terms could be obtained by applying a statistical averaging operator on the corresponding elementary conservation laws written for each fluid. This produces many fluctuation correlation terms that must be closed, usually by introducing the supplementary physical properties of the system.

This paper is organized as follows: after this introduction, we present in Section 2, modelling and governing equations for the compressible multi-phase flows considered in this work. Next, Sections 3–5 introduce the time discretization, and interface sharpening strategy within the framework of space discretization. Following this we present the numerical results obtained for the two-material flows in Section 6, and for the multi-material flows in Section 7. Finally, the conclusions are drawn and scope for the future work is briefly discussed.

2. Physical Model and Governing Equations

Considering a compressible multi-phase flow with N phases, where each phase is indexed by $\phi \in \{1, ..., N\}$, and is modelled as a compressible medium with a thermodynamic state defined by its:

1. mass density ρ_ϕ;
2. specific internal energy e_ϕ;
3. temperature T_ϕ;
4. pressure p_ϕ.

Further, each phase ϕ obeys a coupled system of governing equations, composed of the balance equations of mass, momentum and energy for each phase. We consider here the situation where the pressures are at the instantaneous equilibrium everywhere, so that we have:

$$p_\phi = p \qquad \forall \phi \in \{1, ..., N\}. \tag{1}$$

where p is an equilibrium pressure. We also consider that every control volume (in the terminology of the Eulerian framework) is filled by different phases, such that:

$$\sum_{\phi=1}^{N} \alpha_\phi = 1, \tag{2}$$

with $\alpha_\phi \in [0,1]$ is the volume fraction of any phase ϕ. Additionally, the system is completed with two equations of state (EOS), relating the four thermodynamic variables. The well-known complete set of volume-averaged equations for any phase ϕ is written here in terms of the internal energies, as shown below:

$$\frac{\partial(\rho_\phi \alpha_\phi)}{\partial t} + \nabla \cdot (\rho_\phi \alpha_\phi \mathbf{u}_\phi) = 0, \tag{3}$$

$$\frac{\partial(\rho_\phi \alpha_\phi \mathbf{u}_\phi)}{\partial t} + \nabla \cdot (\rho_\phi \alpha_\phi \mathbf{u}_\phi \otimes \mathbf{u}_\phi) = -\alpha_\phi \nabla(p_\phi) + \mathbf{F}_\phi, \tag{4}$$

$$\frac{\partial(\rho_\phi \alpha_\phi e_\phi)}{\partial t} + \nabla \cdot (\rho_\phi \alpha_\phi e_\phi \mathbf{u}_\phi) = -p_\phi \left[\frac{\partial(\alpha_\phi)}{\partial t} + \nabla \cdot (\alpha_\phi \mathbf{u}_\phi) \right], \tag{5}$$

$$p_\phi = p \qquad \forall \phi \in \{1, ..., N\}, \tag{6}$$

$$\sum_\phi \alpha_\phi = 1, \tag{7}$$

$$\mathrm{EOS}_1(\rho_\phi, e_\phi, T_\phi, p_\phi) = 0, \tag{8}$$

$$\mathrm{EOS}_2(\rho_\phi, e_\phi, T_\phi, p_\phi) = 0 \qquad \forall \phi \in \{1, ..., N\}, \tag{9}$$

The above system is identified by the unknowns $[\mathbf{u}_\phi, \rho_\phi, \alpha_\phi, e_\phi, T_\phi, p_\phi]$. $\mathbf{u}_\phi$ is the phase velocity and we denote the components of the vector $\mathbf{u}_\phi = (u_\phi, v_\phi)$ in 2D and $\mathbf{u}_\phi = (u_\phi, v_\phi, w_\phi)$ in 3D. EOS$_i$ stands for the Equation Of State number i; $i = 1,2$. The form of EOS$_i$ mentioned above is generic. The specific definitions are given in Sections 6 and 7.

Furthermore, in the present study, as evident from the above system of equations, we have separate momentum equations for each phase with independent velocities per phase. It is known that without the velocity relaxation terms, this system is conditionally hyperbolic with the loss of hyperbolicity under certain conditions [27,28]. This issue can be fixed by using a large friction source term $\mathbf{F}_\phi$ for balancing velocities between the phases.

In this work we deal with the situation where there is a full mechanical equilibrium. This implies that the phases share a common velocity, which is achieved through the use of a term $\mathbf{F}_\phi$. The form of which is:

$$\mathbf{F}_\phi = D \left[\sum_{\phi^* \neq \phi}^{N} \mathbf{u}_{\phi^*} - (N-1)\mathbf{u}_\phi \right]. \tag{10}$$

where D is a friction constant which determines the rate at which velocities equilibrate, and ϕ^* is a conjugate phase i.e., $\phi^* \neq \phi$.

3. Time and Space Discretization

3.1. Time Discretization

The time segment $[0, T]$ is split into successive time-steps $[t^n, t^{n+1}]$ with $\Delta t \equiv \Delta t^n = t^{n+1} - t^n > 0$ and $n \in \mathbb{N}$. We further identify any variable or parameter ψ at time t^n as ψ^n. In this study we first consider a semi-discrete, first-order, semi-implicit time discretization scheme, where discrete spatial operators are written in an explicit form, except in the pressure terms. The pressure terms are made implicit for numerical stability reasons and to make use of larger time steps, choice of which is restricted by CFL-like [29] conditions based on material velocities. This gives rise to a coupled, nonlinear system of semi-discrete equations:

$$\rho_\phi^{n+1}\alpha_\phi^{n+1} = \rho_\phi^n\alpha_\phi^n - \Delta t\,\nabla\cdot(\rho_\phi^n\alpha_\phi^n\mathbf{u}_\phi^n), \tag{11}$$

$$\rho_\phi^n\alpha_\phi^n\,(\mathbf{u}_\phi^{n+1} - \mathbf{u}_\phi^n) = -\Delta t\,\rho_\phi^n\alpha_\phi^n\,\mathbf{u}_\phi^n\cdot\nabla(\mathbf{u}_\phi^n) - \Delta t\,\alpha_\phi^n\,\nabla p_\phi^{n+1} + \Delta t\,\mathbf{F}_\phi^{n+1}, \tag{12}$$

$$\rho_\phi^{n+1}\alpha_\phi^{n+1}e_\phi^{n+1} = \rho_\phi^n\alpha_\phi^n e_\phi^n - \Delta t\,\nabla\cdot(\rho_\phi^n\alpha_\phi^n e_\phi^n\mathbf{u}_\phi^n) - \Delta t\,p_\phi^{n+1}\left[\frac{\alpha_\phi^{n+1}-\alpha_\phi^n}{\Delta t} + \nabla\cdot(\alpha_\phi^n\mathbf{u}_\phi^n)\right], \tag{13}$$

$$p_\phi^{n+1} = p^{n+1}, \tag{14}$$

$$\sum_\phi \alpha_\phi^{n+1} = 1, \tag{15}$$

$$\mathrm{EOS}_i(\rho_\phi^{n+1}, e_\phi^{n+1}, T_\phi^{n+1}, p_\phi^{n+1}) = 0, \quad i = 1,2. \tag{16}$$

Here superscripts n and $n+1$ denote the term at previous discrete time t^n (explicit terms) and next discrete time t^{n+1} (implicit terms), respectively. The above system of partial differential Equations (11)–(16) is in the semi-implicit form in time. With regard to the time scheme, we have taken a two-step approach for the numerical solution of the system (11)–(16). These are:

1. Explicit convection/advection step: This step deals with the numerical calculation of all explicit terms in the system (11)–(16). This step is also called a convection/advection step (referred to as advection step hereafter) because these explicit terms can be determined by simply solving the advection problems for $\mathbf{u}_\phi$, α_ϕ, $\alpha_\phi\rho_\phi$ and $\alpha_\phi\rho_\phi e_\phi$, as shown below:

Time discrete system for the explicit advection step

$$(\mathbf{u}_\phi)^{c,n+1} = \mathbf{u}_\phi^n - \Delta t\,\mathbf{u}_\phi^n\cdot\nabla\mathbf{u}_\phi^n, \tag{17}$$

$$(\alpha_\phi)^{c,n+1} = \alpha_\phi^n - \Delta t\,\nabla\cdot(\alpha_\phi^n\mathbf{u}_\phi^n), \tag{18}$$

$$(\rho_\phi\alpha_\phi)^{c,n+1} = \rho_\phi^n\alpha_\phi^n - \Delta t\,\nabla\cdot(\rho_\phi^n\alpha_\phi^n\mathbf{u}_\phi^n), \tag{19}$$

$$(\rho_\phi\alpha_\phi e_\phi)^{c,n+1} = \rho_\phi^n\alpha_\phi^n e_\phi^n - \Delta t\,\nabla\cdot(\rho_\phi^n\alpha_\phi^n e_\phi^n\mathbf{u}_\phi^n). \tag{20}$$

where superscript c denotes the advected quantities.

Remark 1. *In Section 5 we will focus on the application of the sharp interface capturing strategy for the space discretization of divergence operators in Equations (18)–(20).*

2. Implicit step: This step deals with the numerical solution of only implicit terms in the system (11)–(16) after calculating the explicit ones by solving the explicit advection step. To rewrite the system (11)–(16) in form of only implicit quantities, we perform the following substitution: (19) in (11); (17) in (12); (20) and (18) in (13). We consequently get the system of equations containing only implicit terms at time t^{n+1}:

$$\rho_\phi^{n+1}\alpha_\phi^{n+1} = (\rho_\phi\alpha_\phi)^{c,n+1}, \tag{21}$$

$$\rho_\phi^n\alpha_\phi^n\,(\mathbf{u}_\phi^{n+1} - (\mathbf{u}_\phi)^{c,n+1}) = -\Delta t\,\alpha_\phi^n\,\nabla(p_\phi^{n+1}) + \Delta t\,\mathbf{F}_\phi^{n+1}, \tag{22}$$

$$\rho_\phi^{n+1}\alpha_\phi^{n+1}e_\phi^{n+1} = (\rho_\phi\alpha_\phi e_\phi)^{c,n+1} - p_\phi^{n+1}\,(\alpha_\phi^{n+1} - (\alpha_\phi)^{c,n+1}), \tag{23}$$

$$p_\phi^{n+1} = p^{n+1}, \tag{24}$$

$$\sum_{\phi} \alpha_{\phi}^{n+1} = 1, \tag{25}$$

$$EOS_i(\rho_{\phi}^{n+1}, e_{\phi}^{n+1}, T_{\phi}^{n+1}, p_{\phi}^{n+1}) = 0, \quad i = 1,2. \tag{26}$$

At this point it is worth mentioning that the time scheme presented here is a modified version of the so-called Implicit Continuous Eulerian (ICE) scheme [30,31]. But unlike ICE, where velocity variables ($\mathbf{u}_{\phi}$) are made implicit in the advection equations of volume fractions (α_{ϕ}), partial masses ($\rho_{\phi}\alpha_{\phi}$) and partial energies ($\rho_{\phi}\alpha_{\phi}e_{\phi}$), we have used instead explicit time discretization. Indeed, in this first approach, we made the choice to solve only pressure terms in an implicit way because, as explained in [32] for a two-phase flow system, the characteristic time scale for the pressure propagation is much smaller than that of advection. Thus, choosing an implicit time discretization for pressure will avoid the choice of very small time steps for its numerical resolution.

Next, rewriting Equations (21)–(26) in the form of phase volumes ($V_{\phi} = \alpha_{\phi} V$, where V denotes the sum of volumes of all the phases) and mass (m_{ϕ}) is preferred:

$$m_{\phi}^{n+1} = \rho_{\phi}^{n+1} V_{\phi}^{n+1} = V\,(\rho_{\phi}\alpha_{\phi})^{c,n+1}, \tag{27}$$

$$m_{\phi}^{n}\,(\mathbf{u}_{\phi}^{n+1} - (\mathbf{u}_{\phi})^{c,n+1}) = -\Delta t\, V_{\phi}^{n}\, \nabla(p_{\phi}^{n+1}) + \Delta t\, \mathbf{F}_{V,\phi}^{n+1}, \tag{28}$$

where $m_{\phi}^{n} = V(\rho_{\phi}\alpha_{\phi})^{n}$ and $\mathbf{F}_{V,\phi}^{n+1} = V\, \mathbf{F}_{\phi}^{n+1}$,

$$m_{\phi}^{n+1} e_{\phi}^{n+1} = (E_{\phi})^{c,n+1} - p_{\phi}^{n+1}\,(V_{\phi}^{n+1} - (V_{\phi})^{c,n+1}), \tag{29}$$

where $(E_{\phi})^{c,n+1} = V\,(\rho_{\phi}\alpha_{\phi}e_{\phi})^{c,n+1}$ and $(V_{\phi})^{c,n+1} = V\alpha_{\phi}^{c,n+1}$,

$$p_{\phi}^{n+1} = p^{n+1}, \tag{30}$$

$$\sum_{\phi} V_{\phi}^{n+1} = V, \tag{31}$$

$$\mathrm{EOS}_i(m_{\phi}^{n+1}/V_{\phi}^{n+1}, e_{\phi}^{n+1}, T_{\phi}^{n+1}, p_{\phi}^{n+1}) = 0, \quad i = 1,2. \tag{32}$$

The first relation given by Equation (27) will be solved after solving the following ones in order to compute partial mass density for each phase (see Section 4). Notice that the operation $m_{\phi}^{n+1}/V_{\phi}^{n+1}$ in Equation (32) to get ρ_{ϕ}^{n+1} forbids the use of a zero phase volume.

Thus, we summarize the final time discrete system for the implicit step with the new unknowns $\left[\mathbf{u}_{\phi}^{n+1}, m_{\phi}^{n+1}, V_{\phi}^{n+1}, e_{\phi}^{n+1}, T_{\phi}^{n+1}, p_{\phi}^{n+1}\right]$:

Time discrete system for the implicit step

$$m_{\phi}^{n}\,(\mathbf{u}_{\phi}^{n+1} - (\mathbf{u}_{\phi})^{c,n+1}) = -\Delta t\, V_{\phi}^{n}\, \nabla p_{\phi}^{n+1} + \Delta t\, \mathbf{F}_{V,\phi}^{n+1}, \tag{33}$$

$$m_{\phi}^{n+1} e_{\phi}^{n+1} = (E_{\phi})^{c,n+1} - p_{\phi}^{n+1}\,(V_{\phi}^{n+1} - (V_{\phi})^{c,n+1}), \tag{34}$$

$$p_{\phi}^{n+1} = p^{n+1} \tag{35}$$

$$\sum_{\phi} V_{\phi}^{n+1} = V, \tag{36}$$

$$\mathrm{EOS}_i(m_{\phi}^{n+1}/V_{\phi}^{n+1}, e_{\phi}^{n+1}, T_{\phi}^{n+1}, p_{\phi}^{n+1}) = 0, \quad i = 1,2. \tag{37}$$

This time discrete system in this implicit step is coupled and non-linear. We generically denote it for any phase ϕ by:

$$\mathbf{Q}^{time}\left(\mathbf{u}_{\phi}^{n+1}, m_{\phi}^{n+1}, V_{\phi}^{n+1}, e_{\phi}^{n+1}, T_{\phi}^{n+1}, p_{\phi}^{n+1}\right) = 0. \tag{38}$$

where $\mathbf{Q}^{time}$ represents the system of Equations (33)–(37) in the implicit step, and discretized only with respect to time.

3.2. Space Discretization

In our study we deal with the Finite Volume method (FVM) [33] discretization of Equations (33)–(37). It is important to remark at this point that here in this section, for the sake of conciseness, we prefer not to highlight the numerical strategy for the FV discretization of the divergence operators in the advection problems' set (17)–(20), we reserve that analysis for Section 5. Further, the two-dimensional (2D) and three-dimensional (3D) versions have been implemented, but at this step, only a 2D version has been validated and thus presented in this paper. We thus consider a two-dimensional (2D) uniform Cartesian mesh [34] with staggered variables for space discretization (in the spirit of Raviart–Thomas finite elements), in order to avoid the pressure oscillations due to the checkerboard problem [33]. Thus, as usual, the normal velocity variables are stored at faces of control volumes, whereas all other scalars (volume fractions and thermodynamic variables, generically noted by Π hereafter) are stored at cell centers, as shown in Figure 1 [35].

After performing a finite volume space discretization of time discrete model (33)–(37), we get a non-linear and coupled algebraic model, which generically can be written as:

$$\mathbf{Q}^{time}_{space}\left(u^{n+1}_{\phi,face}, v^{n+1}_{\phi,face}, m^{n+1}_{\phi,center}, V^{n+1}_{\phi,center}, e^{n+1}_{\phi,center}, T^{n+1}_{\phi,center}, p^{n+1}_{\phi,center}\right) = 0. \tag{39}$$

where $\mathbf{Q}^{time}_{space}$ represents the system of Equations (33)–(37) in the implicit step, and discretized with respect to time and space.

Further, to find the numerical solution of the above model on 2D Cartesian meshes, we have used the *Newton's method* [36] from PETSc library (see Section 4), and thus for any phase ϕ we can express it in the general form of a solvable *linear system* at time t^{n+1}:

$$\mathbf{J}(\mathbf{X}^k)(\mathbf{X}^{k+1} - \mathbf{X}^k) = -(\mathbf{Q}^{time}_{space})(\mathbf{X}^k), k = 0, 1, ..., \tag{40}$$

where

$$\mathbf{X}^k = \begin{bmatrix} u^{n+1,k}_{\phi,face} \\ v^{n+1,k}_{\phi,face} \\ m^{n+1,k}_{\phi,center} \\ V^{n+1,k}_{\phi,center} \\ e^{n+1,k}_{\phi,center} \\ T^{n+1,k}_{\phi,center} \\ p^{n+1,k}_{\phi,center} \end{bmatrix}$$

with superscript k refering to the k^{th} Newton iteration, and $\mathbf{X}^0$ is an initial approximation given by the values: $(u^n_{\phi,face}, v^n_{\phi,face}, m^n_{\phi,center}, V^n_{\phi,center}, e^n_{\phi,center}, T^n_{\phi,center}, p^n_{\phi,center})$. Matrix $\mathbf{J} = (\mathbf{Q}^{time}_{space})'$ is a *block* and *sparse* **Jacobian** matrix of the non-linear and coupled system (39). Again, *block* nature of the Jacobian matrix highlights the fact that the algebraic system (39) is coupled.

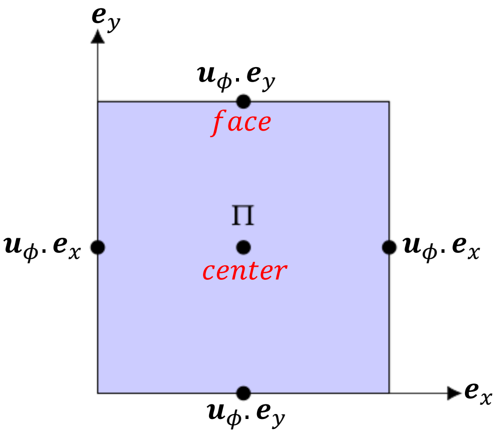


Figure 1. A pictorial depiction of a staggered variable layout with normal phase velocities stored at the faces, and all scalars (Π) stored at the cell centers.

4. Software Architecture for Numerical Resolution of The Model

To compute the solution of the nonlinear, coupled multi-phase model, we use our in-house high-performance software platform MMICADO: Mesh based Multi-phase Interaction Capabilities for severe Accident Dedicated cOdes. MMICADO is a CEA's proprietary software designed by following the Object-Oriented Programming (OOP) principles in the modern C++. MMICADO provides a flexible framework for meshes, parallelism (using DUNE library [37]), numerical methods, input data parsers (TinyXML library [38]), solvers (mainly with PETSc library [39]), primarily for developing any new multi-phase application. An overview of such a platform is shown in Figure 2.

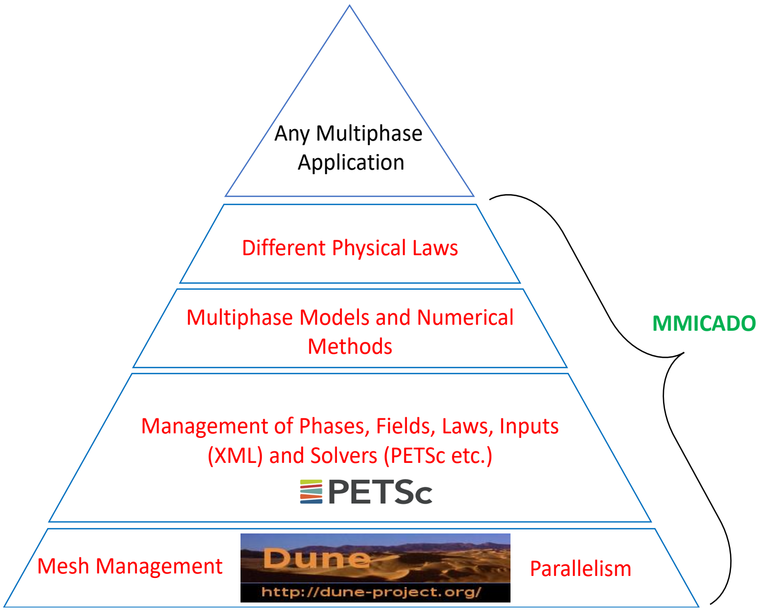

Any Multiphase Application
Different Physical Laws
Multiphase Models and Numerical Methods
Management of Phases, Fields, Laws, Inputs (XML) and Solvers (PETSc etc.)
PETSc
Mesh Management
Dune
http://dune-project.org/
Parallelism
MMICADO

Figure 2. An overview of the MMICADO software.

For example, to solve system (39) we have used PETSc's Linear Search Newton (SNESNEWTONLS) [40] method (a non-linear solver) within the framework of MMICADO, where the Jacobian matrix is computed numerically using a finite difference approximation. This is followed by the use of PETSc's linear solver KSPPREONLY [41] for solving the final global linear matrix, which uses the LU factorization (specifically the PETSc's PCLU [41]) as a pre-conditioner only once. Having said that, we now summarize the time algorithm in Figure 3 showing the sequence in which the model is solved in MMICADO.

Remark 2. *To get the numerical solution of the system* (39)*, the following CFL condition based on the material velocities should be fulfilled:*

$$\frac{\Delta t}{h} \max_{\phi} \|\boldsymbol{u}^n_\phi\|_\infty \leq \eta,$$

where $\eta \leq \frac{1}{2}$, Δt *is the discrete time step and* h *is the constant distance between cell centers in a 2D Cartesian mesh.*

In the next section we will show the details of the *interface sharpening* approach for the *advection step* shown in Figure 3.

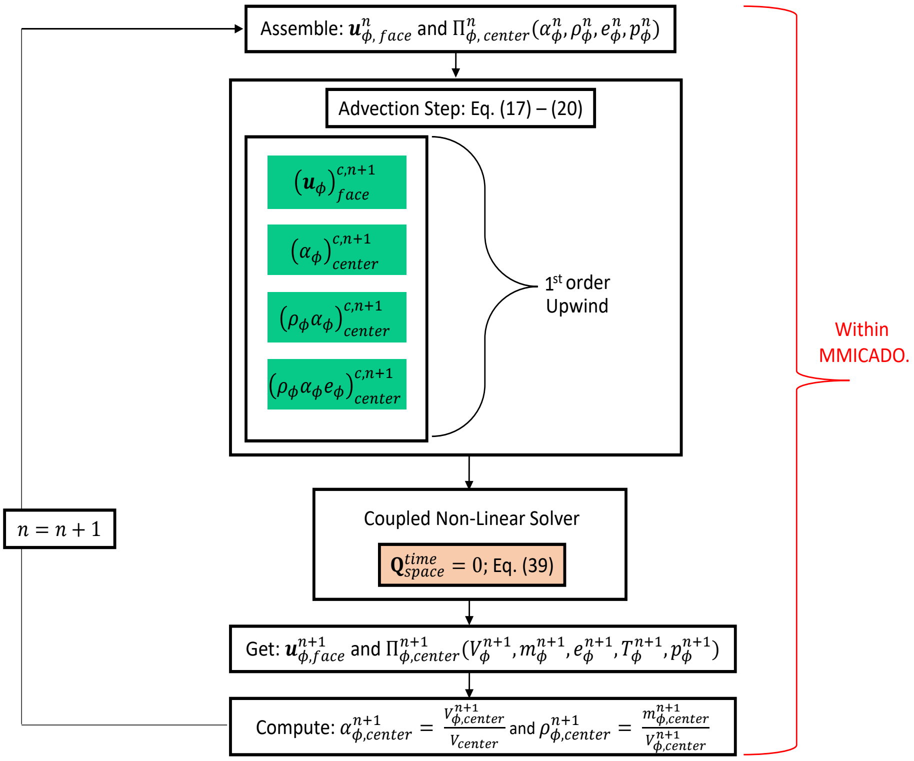

Assemble: $\boldsymbol{u}^n_{\phi,\,face}$ and $\Pi^n_{\phi,\,center}(\alpha^n_\phi, \rho^n_\phi, e^n_\phi, p^n_\phi)$
Advection Step: Eq. (17) – (20)
$(\boldsymbol{u}_\phi)^{c,n+1}_{face}$
$(\alpha_\phi)^{c,n+1}_{center}$
$(\rho_\phi \alpha_\phi)^{c,n+1}_{center}$
$(\rho_\phi \alpha_\phi e_\phi)^{c,n+1}_{center}$
1st order Upwind
Within MMICADO.
$n = n + 1$
Coupled Non-Linear Solver
$\mathbf{Q}^{time}_{space} = 0$; Eq. (39)
Get: $\boldsymbol{u}^{n+1}_{\phi,face}$ and $\Pi^{n+1}_{\phi,center}(V^{n+1}_\phi, m^{n+1}_\phi, e^{n+1}_\phi, T^{n+1}_\phi, p^{n+1}_\phi)$
Compute: $\alpha^{n+1}_{\phi,center} = \frac{V^{n+1}_{\phi,center}}{V_{center}}$ and $\rho^{n+1}_{\phi,center} = \frac{m^{n+1}_{\phi,center}}{V^{n+1}_{\phi,center}}$

Figure 3. The time algorithm showing the sequence in which the model is solved in MMICADO (details of the interface sharpening method for the advection step are given in Section 5). In this figure, $\Pi^n_{\phi,\text{center}}$ refers to the set of scalars $\alpha^n_\phi, \rho^n_\phi, e^n_\phi, p^n_\phi$ at a discrete time t^n, and computed at cell center. Similarly, $\Pi^{n+1}_{\phi,\text{center}}$ refers to the set of scalars $V^{n+1}_\phi, m^{n+1}_\phi, e^{n+1}_\phi, T^{n+1}_\phi, p^{n+1}_\phi$ computed at cell centers at a discrete time t^{n+1}.

5. Modification of the Advection Step: Sharp Interface Capturing (IC) Approach

5.1. Interface Capturing (IC) Strategy: Multidimensional Limiting Process with the Upper Bound Limiter (MLP–UB)

To ensure the sharp interface capturing in the solution of the compressible multi-material model, the reconstruction of the scalars $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$ in each finite volume cell during the advection step depicted in time algorithm (see Figure 3) is done using a second-order multidimensional MUSCL slope reconstruction strategy [19], where the predicted slope/gradient is limited using a recently developed, compressive, shape-preserving limiter, referred to as *Multidimensional Limiting Process–Upper Bound (MLP–UB)* (see [18]), which is an updated version of the *MLP* methodology detailed in [20]. We briefly elaborate this new MLP–UB method for a generic scalar variable $z(\mathbf{x}, t)$ advected in a medium with velocity $\mathbf{u}$:

$$\frac{\partial z}{\partial t} + \nabla \cdot (z\mathbf{u}) = 0. \tag{41}$$

Let us use K as a generic notation for a given finite volume cell, A as an edge (or a face in 3D) of this volume K, and a vector $\boldsymbol{\nu}_A$ being a normal unit vector to A pointing out of the cell K (see Figure 4). We look for the semi-discrete finite volume schemes of the form:

$$\frac{dz_K}{dt} = -\frac{1}{|K|} \sum_{A \subset \partial K} |A| \, z_A \, \boldsymbol{u}_A \cdot \boldsymbol{\nu}_A. \tag{42}$$

where z_K is the value of z defined at the center of a cell K, and z_A (resp. $\boldsymbol{u}_A$) denotes a certain value of z (resp. $\boldsymbol{u}$) at the midpoint $\boldsymbol{x}_A$ of an edge A.

As usual for a finite volume scheme, flux is linearly reconstructed in each cell, and a slope limiter is applied to limit this reconstructed gradient. The general expression of a MUSCL [19] linear reconstruction of a discrete field z_K in cell K is:

$$\mathscr{I}_K^h z(\boldsymbol{x}) = z_K + \Theta_K \, (\nabla^h z)_K \cdot (\boldsymbol{x} - \boldsymbol{x}_K). \tag{43}$$

where $\boldsymbol{x}_K$ is the center of mass of cell K, $(\nabla^h z)_K$ is a predicted estimate of the gradient, and Θ_K is a positive scalar intended for limiting this gradient. Next the reconstructed value z_A at the midpoint $\boldsymbol{x}_A$ of an edge A in cell K is given by:

$$z_A = \mathscr{I}_K^h \, z(\boldsymbol{x}_A) = z_K + \Theta_K \, (\nabla^h z)_K \cdot (\boldsymbol{x}_A - \boldsymbol{x}_K).$$

which, in case of a Cartesian mesh with h as a constant distance between cell centers, can be simplified as:

$$z_A = z_K + \frac{h}{2} \, \Theta_K \, (\nabla^h z)_K \cdot \boldsymbol{\nu}_A.$$

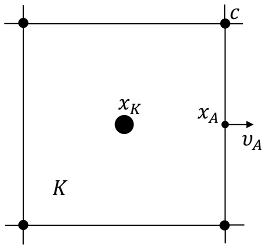


Figure 4. Geometric entities of a finite volume cell K: mass center $\boldsymbol{x}_K$, edge A, edge center $\boldsymbol{x}_A$, normal unit vector $\boldsymbol{\nu}_A$, corner c.

Main steps for the calculation of gradient limiter (Θ_K):

Here we summarize main steps of the MLP-UB method. Detailed analysis of these steps can be found in [18].

Step 1: Prediction of the gradient

In this study, gradient is predicted with a 8-point formula (see [18]). This 8-point formula for calculating the gradient of a scalar z_K in cell K of a 2D Cartesian mesh (see Figure 5) is:

$$(\nabla^h z)_K = \frac{1}{h} \begin{pmatrix} \frac{1}{12}(z_{K_8} - z_{K_6}) + \frac{1}{3}(z_{K_5} - z_{K_4}) \\ +\frac{1}{12}(z_{K_3} - z_{K_1}) \\ \\ \frac{1}{12}(z_{K_8} - z_{K_3}) + \frac{1}{3}(z_{K_7} - z_{K_2}) \\ +\frac{1}{12}(z_{K_6} - z_{K_1}) \end{pmatrix}. \tag{44}$$

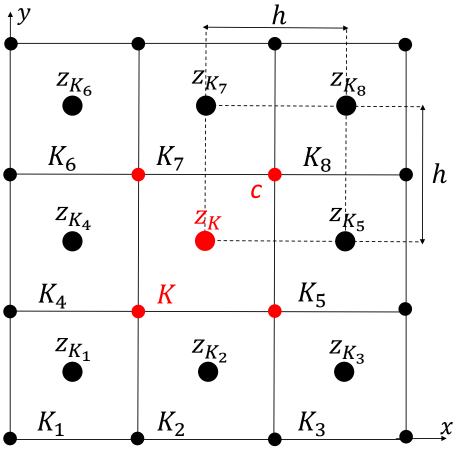


Figure 5. Layout of a 2D Cartesian mesh, where finite volume cells are represented by K, corners by c, value of a scalar z at cell centers by z_K, and constant distance between cell centers by h.

Step 2: Calculate limiters at cell corners (c)

In order to define the scalar limiting functions Θ_c for each corner of a cell K, we compute a *non-limited* (NL) reconstructed value with the predicted centered gradient:

$$\mathscr{I}_K^{NL} z(\boldsymbol{x}) = z_K + (\nabla^h z)_K \cdot (\boldsymbol{x} - \boldsymbol{x}_K).$$

We denote it by $\hat{z}_K(\boldsymbol{x}) = \mathscr{I}_K^{NL} z(\boldsymbol{x})$ for convenience in the next formulae, and we define the extremum value z_c^{extr} as:

$$z_c^{extr} = \begin{cases} \bar{z}_c & \text{if } \hat{z}_K(\boldsymbol{x}_c) > z_K, \\ \underline{z}_c & \text{otherwise.} \end{cases}$$

depending on whether the non-limited reconstructed value is greater or smaller than the cell value. Here, $\bar{z}_c$ and $\underline{z}_c$ are the local extremum values at each corner (c) determined from the different cell neighbors around a cell corner (c) (see Figure 5) by the formulae:

$$\bar{z}_c = \max_{K' \in \text{Cells}(c)} (z_{K'}), \quad \underline{z}_c = \min_{K' \in \text{Cells}(c)} (z_{K'}),$$

where $\text{Cells}(c)$ refers to the set of neighbor cells around a cell corner c. The formula for the corner limiting function is then given as (see [18]):

$$\Theta_c = \min\left(\beta, \frac{z_c^{extr} - z_K}{\hat{z}_K(\boldsymbol{x}_c) - z_K} \right). \tag{45}$$

In case of a zero denominator in the above formula, Θ_c is set to zero, which corresponds to zero gradient. In order to provide the method an ability to limit compressive effects, a threshold value of β is introduced (see [18]), such that the centered gradient can be multiplied at most by β. If $\beta = 1$, no compressive effect is allowed, otherwise $\beta \in (1, 2]$, and some compressive effect is allowed; the value $\beta = 2$ corresponds to the most compressive effect.

Remark 3. *$\beta \leq 2$ bound can be interpreted as the multidimensional equivalent of Ultrabee [42] stability bounds in 1D.*

Step 3: Calculate limiters at cell centers (K)

Finally, limiters at the cell centers are given by the following formula (see [18]):

$$\Theta_K = \min_{c \in \text{Corners}(K)} (\Theta_c). \tag{46}$$

5.2. Reformulation of the Advection Step with the MLP-UB Method for the Sharp Interface Capturing

In this section we describe the methodology for the application of the above discussed MLP-UB [18] method to the advection Equations (18)–(20) of scalars $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$. The main points are discussed in the following sub-sections.

5.2.1. Reconstruction of Primitive Form of Scalars Instead of Their Conservative Form

In the present study, although we work with the conservative form of advection equations for $\alpha_\phi, \rho_\phi, e_\phi$ (see Equations (19) and (20)), we perform the reconstruction of primitive scalars $\alpha_\phi, \rho_\phi, e_\phi$ with MUSCL followed by the gradient limiting with MLP-UB to get their convective fluxes. This is mainly done to serve two purposes:

1. As shown by [27,43,44], the application of the MUSCL on the primitive form of scalars leads to a more accurate reconstruction at the interface.
2. Reconstructing $\alpha_\phi, \rho_\phi, e_\phi$ in their primitive form provides us the freedom to use different types of gradient limiters for α_ϕ and ρ_ϕ, e_ϕ. Here, we prefer to use a compressive flux limiter (MLP-UB in present case) for α_ϕ, and a less compressive one for ρ_ϕ and e_ϕ individually. For instance we have used MINMOD [45] type limiter for ρ_ϕ and e_ϕ.

Further elaborating on point 2 given above, we choose to do so because regular wave solutions shown by ρ_ϕ and e_ϕ in multi-phase test problems are prone to suffer from *staircase effect* (demonstrated in Section 5.2.2) when reconstructed with compressive limiters. However, the evolution of α_ϕ maintains the initial discontinuous solution, and thus to preserve this sharp discontinuity, we use a compressive limiter like MLP-UB.

5.2.2. Modification of MLP-UB Method to Include the Functionality of the Minmod Limiter: 1D Analysis

Here we show a 1D analysis which shows the modification in Formula (45) in order to integrate both MINMOD and MLP-UB methods into a single formula. The 1D version of (45) reads:

$$\Theta_A = \min\left(\beta,\ \frac{z_A^{extr} - z_K}{\hat{z}_K(\boldsymbol{x}_A) - z_K}\right). \tag{47}$$

We introduce a new multiplicative parameter θ in the above formula which results in:

$$\Theta_A = \min\left(\beta,\ \theta\frac{z_A^{extr} - z_K}{\hat{z}_K(\boldsymbol{x}_A) - z_K}\right). \tag{48}$$

In our analysis, we have tested several values of β and θ, and have observed that $\beta = 1$ and $\theta = 0.5$ corresponds to MINMOD limiter. In Figure 6 we can see that a combination of a smooth and hat function is chosen as a initial condition, and subsequently it is linearly advected, where for gradient limiting after MUSCL reconstruction in each finite volume cell, we use relation (48) with two set of values for $(\beta, \theta) \in \{(1, 0.5), (2, 1)\}$. The time scheme used is an explicit second order accurate Heun's RK2 method [46]. We see that the curve produced with MINMOD limiter overlaps with the curve obtained by using relation (48) with $\beta = 1$ and $\theta = 0.5$.

It is also interesting to see that the advection of smooth function with a compressive limiter (using $\beta = 2$ and $\theta = 1$ in relation (48)) leads to the *staircase effect* (see green curve on left-hand side in Figure 6). Thus, we choose to only apply MLP-UB ($\beta = 2$ and $\theta = 1$) on α_ϕ, which initially is discontinuous, and it is necessary to maintain this discontinuity as it gets advected at later times. However, for ρ_ϕ, e_ϕ, we perform their individual reconstruction by using MINMOD limiter (here MLP-UB with $\beta = 1$ and $\theta = 0.5$).

Further, we also extend this analysis to 2D, and subsequently modify the advection step (see Figure 3) with the application of MLP-UB on scalars $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$, as shown in Figure 7.

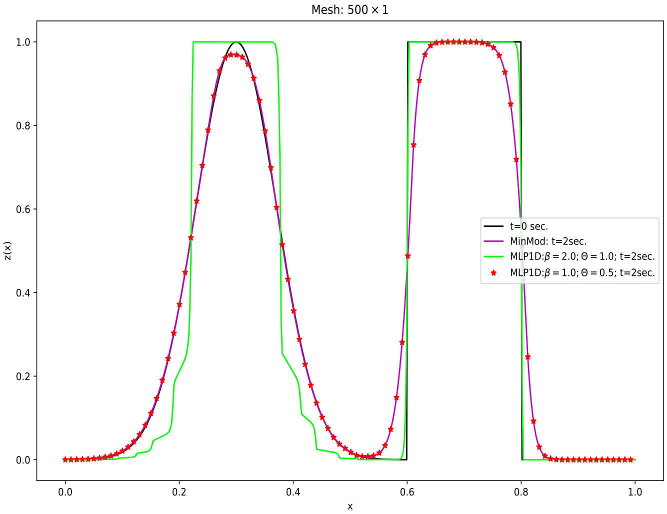


Figure 6. Application of Formula (48) to 1D linear advection of a smooth + hat initial condition with $\beta = 1$ and $\theta = 0.5$ (MINMOD type) and $\beta = 2$ and $\theta = 1$ (MLP-UB).

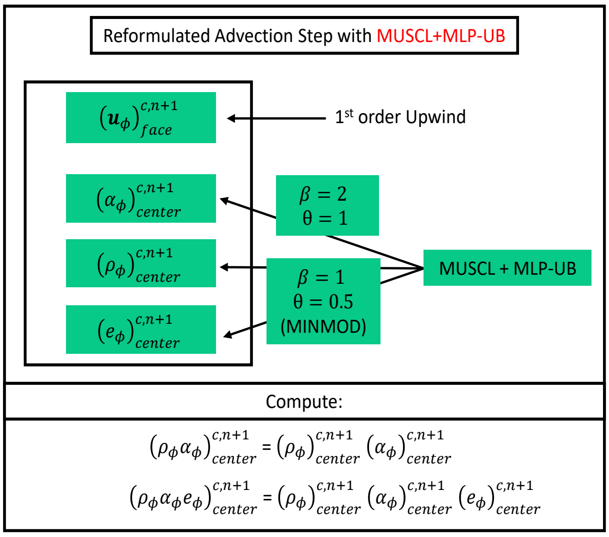

Reformulated Advection Step with MUSCL+MLP-UB
$(u_\phi)^{c,n+1}_{face}$
1st order Upwind
$(\alpha_\phi)^{c,n+1}_{center}$
$(\rho_\phi)^{c,n+1}_{center}$
$(e_\phi)^{c,n+1}_{center}$
β = 2
θ = 1
β = 1
θ = 0.5
(MINMOD)
MUSCL + MLP-UB
Compute:
$(\rho_\phi\alpha_\phi)^{c,n+1}_{center} = (\rho_\phi)^{c,n+1}_{center}\,(\alpha_\phi)^{c,n+1}_{center}$
$(\rho_\phi\alpha_\phi e_\phi)^{c,n+1}_{center} = (\rho_\phi)^{c,n+1}_{center}\,(\alpha_\phi)^{c,n+1}_{center}\,(e_\phi)^{c,n+1}_{center}$

Figure 7. Reformulated advection step with the application of MUSCL+MLP-UB on primitive form of the scalars $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$.

6. Numerical Experiments: Two-Material Test Cases

6.1. Two-Dimensional (2D) Air-Helium Shock-Bubble Interaction: Experimental Validation

We focus here on the comparison of the numerical results produced with our method for the case of an air-helium shock-bubble interaction with one of the classical experiments shown in [47]. The goal, however, is not to have a precise quantitative comparison, but a qualitative one, to show that the method provides physically relevant results with a good resolution of the interfaces, especially when the coarse meshes are used. To study the case, a suitable computational domain is designed (see Figure 8). The dimensions of the computational domain are given in Table 1.

In order to simulate a Mach 1.22 shock wave similar to the one generated in the experiment, we consider the initial conditions shown in Table 2. We also assume that both air and helium are governed by a perfect gas EOS:

$$p_\phi = \frac{m_\phi}{V_\phi} (\gamma_\phi - 1)\, e_\phi, \tag{49}$$

where

$$e_\phi = C_{v_\phi}\, T_\phi. \tag{50}$$

Table 1. Dimensions used for the 2D air-helium shock-bubble interaction test case.

L_x	l_x	l_{xx}	L_y	r
		(m)		
0.312	0.1	0.05	0.0445	0.025

Table 2. Initial conditions used for the 2D air-helium test case.

	ρ (kg/m^3)	u (m/s)	v (m/s)	p (Pa)	γ $(-)$	C_v (J/kgK)
zone 1	1.686	−156.26	0	250,638	1.4	720
zone 2	1.225	0	0	101,325	1.4	720
zone 3	0.228	0	0	101,325	1.648	2440

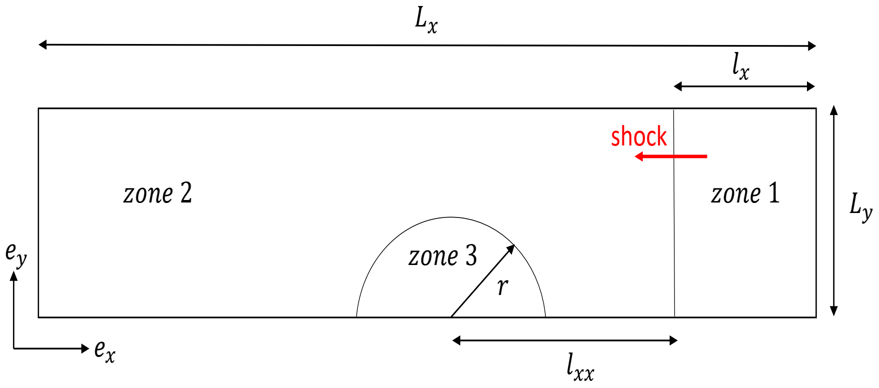
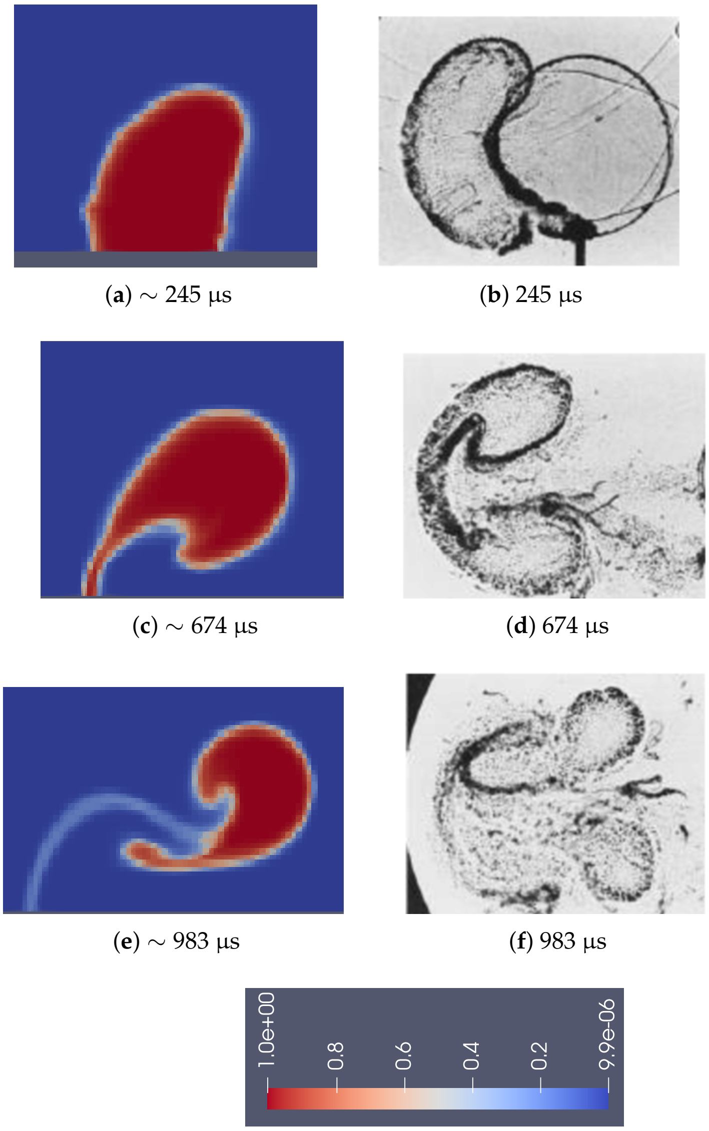


Figure 8. 2D computational domain (not to scale) for the air-helium shock-bubble interaction test case.

Further, the CFL number for this problem is 0.2. We have used a coarse Cartesian mesh of size 294 × 42. Homogeneous Neumann boundary conditions are used for scalars (Π), and slip boundary condition is used for velocities. It the present case we have assumed symmetry in y-direction, thus only the upper half of the computational domain is considered.

Focusing on the numerical results shown in Figure 9, we observe the interaction of a Mach 1.22 shock wave with helium bubble while the shock travels from right to left in the computational domain. The numerical results shown in Figure 9a,c,e are at the three interaction times.These approximately correspond to the same time when shadow-photographs (see Figure 9b,d,f) are taken in the experiment.

Qualitatively, the numerical results correctly depict the deformation of the lighter helium bubble by an air-shock until the end of simulation time. It is interesting to see that even for a relative coarse mesh of size 294 × 42, we get a satisfactory phase separation. This is mainly because of the good interface sharpening effect produced by using MLP-UB method [18]. The results are also free from any interface oscillations or artifacts. We also observe that the values of cell volume fraction (α) for both air and helium are in the range $[0, 1]$ and thus bounded. Some slight differences in the topology could be observed between the numerical results and experimental graphs. This difference can be explained by the absence of a surface tension force in the current numerical model.



(**a**) $\sim$ 245 µs (**b**) 245 µs

(**c**) $\sim$ 674 µs (**d**) 674 µs

(**e**) $\sim$ 983 µs (**f**) 983 µs

(**g**)

Figure 9. 2D numerical results for an air-helium shock-bubble interaction test case. Qualitative comparison of the cell volume fraction (α) of Helium bubble between the modeling (Figures (**a**,**c**,**e**)) and experimental results (Figures (**b**,**d**,**f**)) [47]. The scale of (α) is shown in Figure (**g**).

6.2. Two-Dimensional (2D) Triple Point Case: Two Materials and Three Phases

In this section we show a 2-material and 3-phase triple point test case. This is a standard test case corresponding to a 2D Riemann problem in a rectangular computational domain, which together with the dimensions and initial conditions is displayed in Figure 10. We also consider that the two materials are governed by an ideal gas EOS given by Equation (49). We choose a coarse Cartesian mesh of size 140 × 60. The boundary conditions for this case are Homogeneous Neumann for scalars (Π), and slip for velocities.

Our analysis for this test case primarily focuses on the comparison between the numerical results produced by using two different numerical recipes for the advection of scalars $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$:

1. MUSCL+MLP-UB for α_ϕ, and MINMOD (MUSCL+MLP-UB with $\beta = 1; \theta = 0.5$) for $\rho_\phi, e\phi$.
2. MINMOD for all $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$.

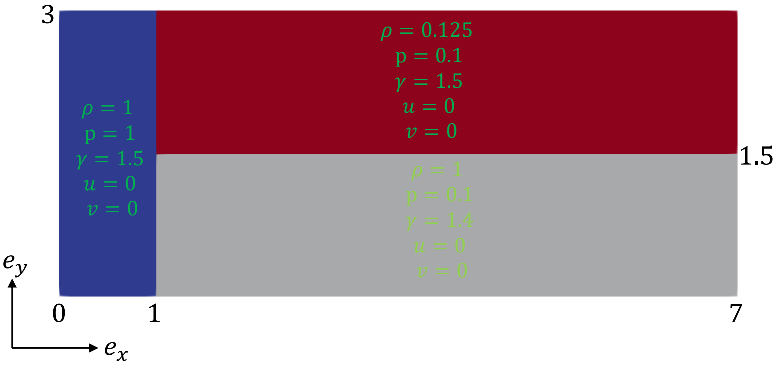
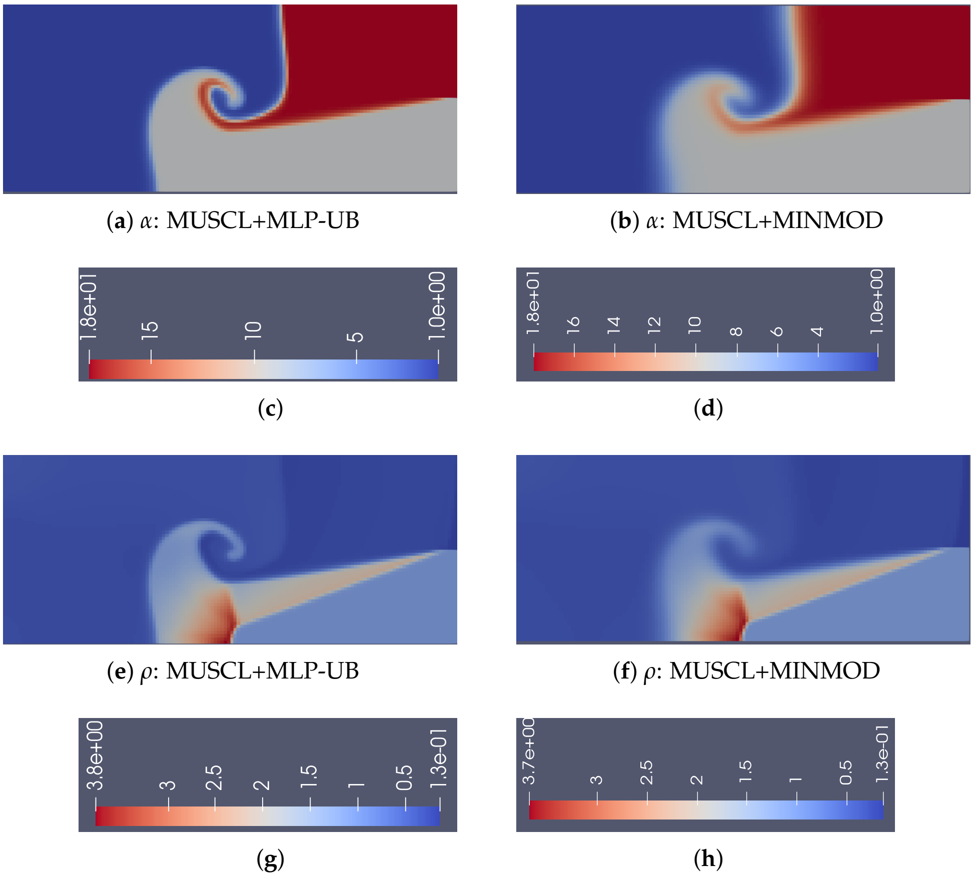


Figure 10. 2D computational domain for the 2-material, 3-phase triple point test case. The units of ρ, p, u, v and dimensions are in S.I. unit system.


(**a**) α: MUSCL+MLP-UB

(**b**) α: MUSCL+MINMOD

(**c**)

(**d**)

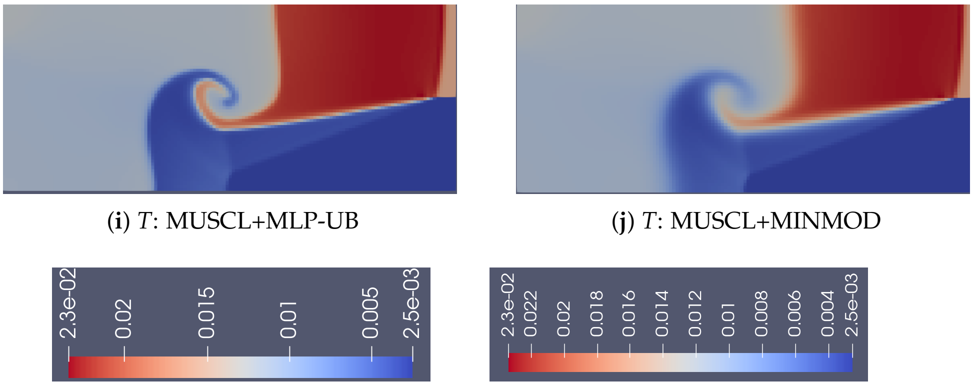
(**e**) ρ: MUSCL+MLP-UB

(**f**) ρ: MUSCL+MINMOD

(**g**)

(**h**)

Figure 11. *Cont.*



(**i**) T: MUSCL+MLP-UB

(**j**) T: MUSCL+MINMOD

(**k**)

(**l**)

Figure 11. 2D numerical results for the 2-material and 3-phase triple point test case using a Cartesian mesh of size 140 × 60. Values of α for all the three phases are presented together, and the same is done for ρ and T. Thus, their scales (see Figures (**c**,**d**,**g**,**h**,**k**,**l**)) are scaled up accordingly.

Firstly, we can observe in Figure 11 that two shocks propagate with different speeds due to the initial density difference between the left, top-right and top-bottom domains (see Figure 11). This creates the shear along the initial phase interfaces, and results in the formation of a vorticity. Regarding the coarse mesh used to compute this vorticity, we observe in Figure 11b,f,j that the numerical solution is diffused due to the use of MINMOD limiter in the numerical advection step of scalars $\Pi(\alpha_\phi, \rho_\phi, e_\phi)$. However, if we focus on Figure 11a,e,i; a significantly better capture of this vorticity can be seen. Numerical results also show an accurate capture of the shocks and multi-phase interfaces. Again, like the previous test case, we observe that the values of cell volume fraction (α) for all the phases are in the range $[0, 1]$.

7. Numerical Experiments: Three-Material Test Case

Two-Dimensional (2D) Sod-like Shock Tube: 3-Material Shock-Bubble Interaction

In this 3-material test case we have simulated a situation where a bubble of some weakly compressible heavy material is surrounded by a vapor film. We have numerically studied the interaction of a shock, generated in the Sod-like shock tube [48], with this bubble-film configuration. The 2D computational domain for this case is displayed in Figure 12, where *zone* 4 is a bubble of heavy material governed by a stiffened gas EOS:

$$p_\phi = \frac{m_\phi}{V_\phi}(\gamma_\phi - 1)e_\phi - \gamma_\phi p_{ref,\phi}, \tag{51}$$

where

$$e_\phi = C_{v_\phi} T_\phi + \frac{p_{ref,\phi}}{\frac{m_\phi}{V_\phi}}. \tag{52}$$

zone 3 here represents a vapor film which, along with *zone* 1 and *zone* 2, is governed by a perfect gas EOS Equation (49). The initial conditions and dimensions of the problem are shown in Tables 3 and 4, respectively. The boundary conditions for this case are same as the previous ones. The size of Cartesian mesh for this case is 800 × 400.

Table 3. Initial conditions used for the 2D 3-material Sod shock-bubble interaction test case.

	ρ (kg/m^3)	u (m/s)	v (m/s)	p (Pa)	p_{ref} (Pa)	γ $(-)$	C_v (J/kgK)
$zone\,1$	1.0	0	0	1.0	0	1.4	100
$zone\,2$	0.125	0	0	0.1	0	1.4	100
$zone\,3$	0.0125	0	0	0.1	0	1.648	100
$zone\,4$	2.0	0	0	0.1	100	2.0	1000

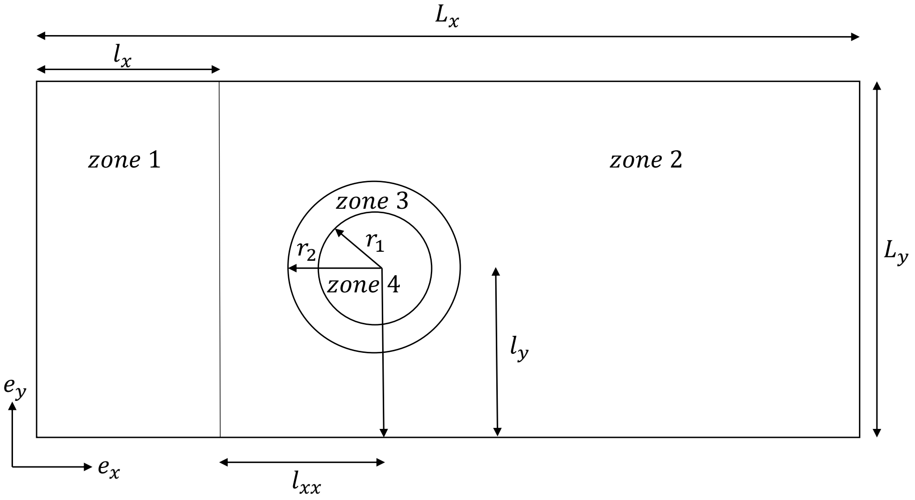


Figure 12. 2D computational domain (not to scale) for the 3-material Sod shock-bubble interaction test case.

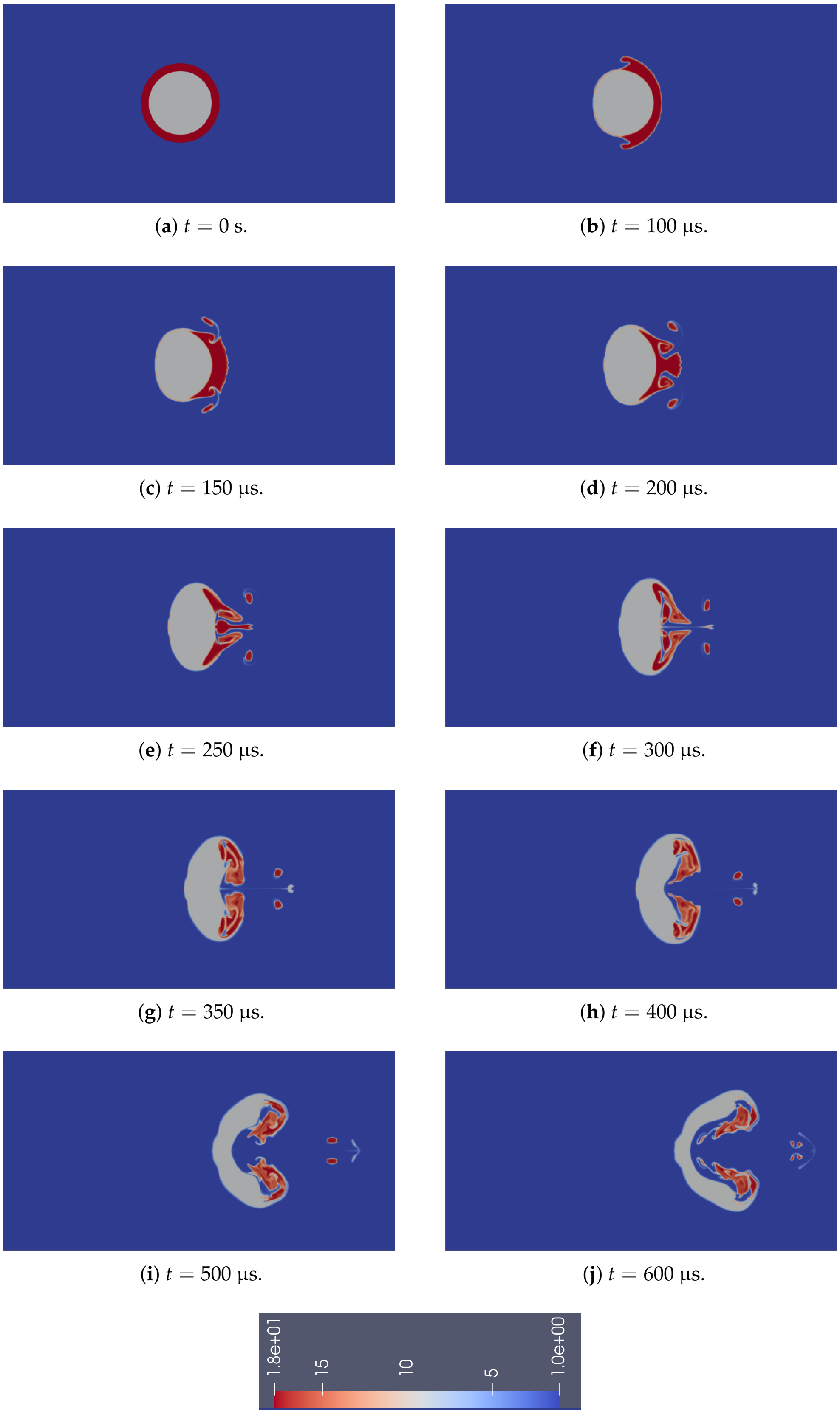
Table 4. Dimensions used for the 2D 3-material Sod shock-bubble interaction test case.

L_x	l_x	l_{xx}	L_y	l_y	r_1	r_2
			(m)			
1.0	0.3	0.15	0.5	0.25	0.08	0.1

From the results shown in Figure 13 we can see that after an interaction with the shock, vapor film detaches from the heavy material bubble, and consequently overturns on the right-hand side of it. It then mixes with the heavy bubble interface as time progresses. We can also see a complete dissociation of two small parts of the vapor film. These two breakaway structures can be seen to have both translational and circular trajectories as they travel towards the right boundary. At the end of simulation they break into two more parts due to their interaction with the reflected shock wave from the right boundary.

In this case, mesh is sufficiently fine to capture the occurrence of the *Richtmyer-Meshkov* [49,50] instabilities between the interface of the light vapor film and heavy material bubble. This is evident from the mixing zone created between the two interfaces at later times. These instabilities occur in the present case because of the perturbed shape of the interface, which is mainly due to the initial representation of a circular interface by a piece-wise constant profile, in each finite volume cell. After a shock interacts with multi-material interfaces, this initial numerical perturbation grows in time in a similar fashion to the growth of perturbations in the physical *Richtmyer-Meshkov* instability.

Thus, in this complex 3-material case as well we see the robustness of our method, which ensures the sharp interface capturing for the intricate evolution of the multi-material interfaces. Furthermore, in this test case, the values of cell volume fraction (α) for all the three materials is observed to be bounded in the range $[0, 1]$.



(**a**) $t = 0$ s.

(**b**) $t = 100$ µs.

(**c**) $t = 150$ µs.

(**d**) $t = 200$ µs.

(**e**) $t = 250$ µs.

(**f**) $t = 300$ µs.

(**g**) $t = 350$ µs.

(**h**) $t = 400$ µs.

(**i**) $t = 500$ µs.

(**j**) $t = 600$ µs.

(**k**)

Figure 13. 2D numerical results for the 3-material shock-bubble interaction test case using a Cartesian mesh of size 800 × 400. Values of α in *zone* 3 and *zone* 4 are shown together. Thus, its scale (see Figure (**k**)) is scaled up accordingly.

8. Conclusions and Scope for the Future Work

In the present work, we have considered the hyperbolic compressible Euler system of equations, which can serve as the basis for numerical simulations of the several industrial applications in the domain of the compressible multi-material flows. We have worked with this model in the limit of a full mechanical equilibrium, and also considered an instantaneous pressure equilibrium. Within the scope of such a mathematical model, we focused on the problem of the interface sharpening, where we chose to work with the Interface Capturing approach. We performed the analysis of interface sharpening using a recently developed Diffuse Interface Method: Multidimensional Limiting Process-Upper Bound (MLP-UB) [18]. This method has been proposed in a recent paper [18], where the authors have demonstrated its robustness for the pure advection cases. The present paper is thus considered as a natural sequel to the work presented in [18].

For performing the numerical simulations in this work we have used our in-house, high-performance, modern software: MMICADO; within its framework, we have first proposed a robust time scheme, where we first explicitly calculate the advection/convection terms. Within the framework of Finite Volume space discretization on 2D Cartesian meshes, the divergence terms in these advection/convection equations are computed by employing a second order accurate, multidimensional MUSCL [19] slope reconstruction strategy along with the MLP-UB gradient limiting method. A 1D analysis is also presented to highlight the choice made for scalar reconstruction in their primitive form (instead of their conservative form) by using the most and the least compressive form of MUSCL+MLP-UB approach.

Finally, we have provided three complex multi-material test cases, where our aim was to produce accurate results on the coarse 2D Cartesian meshes. Looking at the results we can conclude that our approach has produced:

1. Accurate and sharp capturing of the intricate interface evolution without numerical artifacts.
2. Accurate interaction of the shocks with the multi-material interfaces.
3. Bounded numerical results for any number of materials.

A direct extension of the work presented in this paper is to take into account the phenomena of phase change by considering the heat and mass transfer across the multi-material interfaces. Another work which is conceptualized is the study of *sliding interfaces*, where we want to consider the relative motion between the multi-material interfaces. Another immediate work concerns the verification and validation of the 3D version, and theoretical demonstration of the positivity preserving property (boundedness) of the method. We plan to address these issues in the future research articles.

Author Contributions: Conceptualization, C.F., M.P. and R.M.; Data curation, S.N.; Formal analysis, S.N.; Investigation, S.N.; Analysis, S.N., C.F., M.P. and F.D.V.; Methodology, C.F., S.N. and M.P.; Project administration, C.F., M.P. and R.M.; Resources, C.F. and M.P.; Software, S.N., C.F. and M.P.; Supervision, C.F. and M.P.; Validation, S.N.; Visualization, S.N.; Writing original draft, S.N.; Writing review and editing, S.N., M.P. and F.D.V. All authors have read and agreed to the published version of the manuscript.

Funding: No external funding reported.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The present paper contains all the necessary data needed for the thorough understanding of the work. In case a requirement for any supplementary material arises, the readers are encouraged to contact the corresponding authors.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Abarzhi, S.; Bhowmick, A.; Naveh, A.; Pandian, A.; Swisher, N.; Stellingwerf, R.; Arnett, D. Supernova, nuclear synthesis, fluid instabilities, and interfacial mixing. *Proc. Natl. Acad. Sci. USA* **2018**, *116*, 201714502. [CrossRef] [PubMed]
2. Le Touze, C.; Dorey, L.H.; Rutard, N.; Murrone, A. A compressible two-phase flow framework for Large Eddy Simulations of liquid-propellant rocket engines. *Appl. Math. Model.* **2020**, *84*, 265–286. [CrossRef]
3. Nguyen, V.; Hoang, P.; Trong, N.; Park, W.G. Numerical modeling for compressible two-phase flows and application to near-field underwater explosions. *Comput. Fluids* **2021**, *215*, 104805. [CrossRef]
4. Cao, L.; Fei, W.; Grosshans, H.; Cao, N. Simulation of Underwater Explosions Initiated by High-Pressure Gas Bubbles of Various Initial Shapes. *Appl. Sci.* **2017**, *7*, 880. [CrossRef]
5. Cronenberg, A.W.; Benz, R. *Vapor Explosion Phenomena with Respect to Nuclear Reactor Safety Assessment*; Springer: Boston, MA, USA, 1980.
6. Leskovar, M.; Končar, B. Simulation of Steam Explosion with a General Purpose CFD Code. In Proceedings of the 14th International Conference on Nuclear Engineering, Miami, FL, USA, 17–20 July 2006; Volume 2006. [CrossRef]
7. Tezduyar, T.E. Interface-tracking and interface-capturing techniques for finite element computation of moving boundaries and interfaces. *Comput. Methods Appl. Mech. Eng.* **2006**, *195*, 2983–3000. [CrossRef]
8. Jafari, A.; Ashgriz, N. Numerical Techniques for Free SurfaceFlows: Interface Capturing and InterfaceTracking. In *Encyclopedia of Microfluidics and Nanofluidics*; Li, D., Ed.; Springer: Boston, MA, USA, 2008; pp. 1494–1513. [CrossRef]
9. Noh, W.F.; Woodward, P. SLIC (Simple Line Interface Calculation). In *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics June 28–July 2, 1976 Twente University, Enschede*; van de Vooren, A.I., Zandbergen, P.J., Eds.; Springer: Berlin/Heidelberg, Germany, 1976; pp. 330–340.
10. Ubbink, O.; Issa, R. A Method for Capturing Sharp Fluid Interfaces on Arbitrary Meshes. *J. Comput. Phys.* **1999**, *153*, 26–50. [CrossRef]
11. Hirt, C.; Nichols, B. Volume of fluid (VOF) method for the dynamics of free boundaries. *J. Comput. Phys.* **1981**, *39*, 201–225. [CrossRef]
12. Osher, S.; Sethian, J.A. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *J. Comput. Phys.* **1988**, *79*, 12–49. [CrossRef]
13. Enright, D.; Fedkiw, R.; Ferziger, J.; Mitchell, I. A Hybrid Particle Level Set Method for Improved Interface Capturing. *J. Comput. Phys.* **2002**, *183*, 83–116. [CrossRef]
14. She, D.; Kaufman, R.; Lim, H.; Melvin, J.; Hsu, A.; Glimm, J. Chapter 15—Front-Tracking Methods. In *Handbook of Numerical Methods for Hyperbolic Problems*; Handbook of Numerical Analysis; Abgrall, R., Shu, C.W., Eds.; Elsevier: Amsterdam, The Netherlands, 2016; Volume 17, pp. 383–402. [CrossRef]
15. Klingenberg, C.; Plohr, B. An Introduction to front Tracking. In *Multidimensional Hyperbolic Problems and Computations*; Glimm, J., Majda, A.J., Eds.; Springer: New York, NY, USA, 1991; pp. 203–216.
16. Mirjalili, S.; Jain, S.; Dodd, M. Interface-capturing methods for two-phase flows: An overview and recent development. *Cent. Turbul. Res. Annu. Res. Briefs* **2017**, 117–135. Available online: https://doddm.com/publications/2017-ctr-sm-sj-md.pdf (accessed on 14 December 2021).
17. Anderson, D.M.; McFadden, G.B.; Wheeler, A.A. Diffuse-Interface Methods in Fluid Mechanics. *Annu. Rev. Fluid Mech.* **1998**, *30*, 139–165. [CrossRef]
18. De Vuyst, F.; Fochesato, C.; Mahy, V.; Motte, R.; Peybernes, M. A geometrically accurate low-diffusive conservative interface capturing method suitable for multimaterial flows. *Comput. Fluids* **2021**, *227*, 104897. [CrossRef]
19. Van Leer, B. Towards the Ultimate Conservative Difference Scheme. A Second-order Sequel to Godunov's Method. *J. Comput. Phys.* **1979**, *32*, 101–136. [CrossRef]
20. Kim, K.H.; Kim, C. Accurate, efficient and monotonic numerical methods for multi-dimensional compressible flows: Part II: Multi-dimensional limiting process. *J. Comput. Phys.* **2005**, *208*, 570–615. [CrossRef]
21. Després, B.; Lagoutière, F. Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. *J. Sci. Comput.* **2001**, *16*, 479–524. [CrossRef]
22. Xiao, F.; Ii, S.; Chen, C. Revisit to the THINC scheme: A simple algebraic VOF algorithm. *J. Comput. Phys.* **2011**, *230*, 7086–7092. [CrossRef]
23. So, K.; Hu, X.; Adams, N. Anti-diffusion interface sharpening technique for two-phase compressible flow simulations. *J. Comput. Phys.* **2012**, *231*, 4304–4323. [CrossRef]
24. Vazquez-Gonzalez, T.; Llor, A.; Fochesato, C. Ransom test results from various two-fluid schemes: Is enforcing hyperbolicity a thermodynamically consistent option? *Int. J. Multiph. Flow* **2016**, *81*, 104–112. [CrossRef]
25. Dinh, T.N.; Nourgaliev, R.R.; Theofanous, T.G. Understanding the ill-posed two-fluid model. In Proceedings of the 10th International Topical Meeting on Nuclear Reactor Thermal Hydraulics (NURETH-10), Seoul, Korea, 5–11 October 2003; pp. 5–9.
26. Donald A. Drew, S.L.P. *Theory of Multicomponent Fluids*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 1999. [CrossRef]
27. Saurel, R.; Lemetayer, O. A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation. *J. Fluid Mech.* **2001**, *431*, 239–271. [CrossRef]

28. Paillère, H.; Corre, C.; Garca Cascales, J. On the extension of the AUSM+ scheme to compressible two-fluid models. *Comput. Fluids* **2003**, *32*, 891–916. [CrossRef]
29. Courant, R.; Friedrichs, K.; Lewy, H. Über die partiellen Differenzengleichungen der mathematischen Physik. *Math. Ann.* **1928**, *100*, 32–74. [CrossRef]
30. Harlow, F.H.; Amsden, A.A. A numerical fluid dynamics calculation method for all flow speeds. *J. Comput. Phys.* **1971**, *8*, 197–213. [CrossRef]
31. Harlow, F.H.; Amsden, A.A. Numerical calculation of almost incompressible flow. *J. Comput. Phys.* **1968**, *3*, 80–93. [CrossRef]
32. Bruce Stewart, H.; Wendroff, B. Two-phase flow: Models and methods. *J. Comput. Phys.* **1984**, *56*, 363–409. [CrossRef]
33. Versteeg, H.K. *An Introduction to Computational Fluid Dynamics: The Finite Volume Method*; Longman Scientific & Technical: New York, NY, USA; Wiley: Harlow, Essex, UK, 1995.
34. Blazek, J. Chapter 11—Principles of Grid Generation. In *Computational Fluid Dynamics: Principles and Applications*, 3rd ed.; Blazek, J., Ed.; Butterworth-Heinemann: Oxford, UK, 2015; pp. 357–393. [CrossRef]
35. Gay, G. Modélisation de l'Interaction Combustible-réFrigérant: Fragmentation et Explosion de Vapeur. PhD Thesis, Aix-Marseille University, Marseille, France, 2020.
36. Petsc Newton. Available online: https://petsc4py.readthedocs.io/en/stable/manual/snes/#equation-fx0 (accessed on 14 December 2021).
37. Sander, O. *DUNE—The Distributed and Unified Numerics Environment*; Springer Nature: Berlin/Heidelberg, Germany, 2020.
38. TinyXml. Available online: http://www.grinninglizard.com/tinyxml/ (accessed on 14 December 2021).
39. PETSc Documentation. Available online: https://petsc.org/ (accessed on 14 December 2021).
40. Petsc SNES. Available online: https://petsc.org/release/docs/manualpages/SNES/SNESNEWTONLS.html (accessed on 14 December 2021).
41. PETSc Linear Solver. Available online: https://petsc.org/release/docs/manualpages/KSP/KSPPREONLY.html (accessed on 14 December 2021).
42. Roe, R. Some contributions to the modelling of discontinuous flows. *Lect. Appl. Math.* **1985**, *22*, 163–193.
43. Saurel, R.; Abgrall, R. A Simple Method for Compressible Multifluid Flows. *Siam J. Sci. Comput.* **1999**, *21*, 1115–1145. [CrossRef]
44. Karni, S. Multicomponent Flow Calculations by a Consistent Primitive Algorithm. *J. Comput. Phys.* **1994**, *112*, 31–43. [CrossRef]
45. Roe, P.L. Characteristic-Based Schemes for the Euler Equations. *Annu. Rev. Fluid Mech.* **1986**, *18*, 337–365. [CrossRef]
46. Burden, A.; Burden, R.; Faires, J. *Numerical Analysis*, 10th ed.; Cengage Learning: Boston, MA, USA , 2016. [CrossRef]
47. Haas, J.-F.; Sturtevant, B. Interaction of weak shock waves with cylindrical and spherical gas inhomogeneities. *J. Fluid Mech.* **1987**, *181*, 41–76. [CrossRef]
48. Sod, G.A. A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. *J. Comput. Phys.* **1978**, *27*, 1–31. [CrossRef]
49. Richtmyer, R.D. Taylor instability in shock acceleration of compressible fluids. *Commun. Pure Appl. Math.* **1960**, *13*, 297–319. [CrossRef]
50. Meshkov, E.E. Instability of the interface of two gases accelerated by a shock wave. *Fluid Dyn.* **1969**, *4*, 101–104. [CrossRef]