Available online at www.sciencedirect.com
ScienceDirect
Journal of Computational Physics 227 (2008) 7674–7700
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A higher-order generalized ghost fluid method for the poor for the three-dimensional two-phase flow computation of underwater implosions

Charbel Farhat *, Arthur Rallu, Sriram Shankaran

Department of Mechanical Engineering and Institute for Computational and Mathematical Engineering, Stanford University, Mail Code 3035, Stanford, CA 94305, USA

Received 19 July 2007; received in revised form 24 February 2008; accepted 24 April 2008
Available online 13 May 2008

Abstract

The ghost fluid method for the poor (GFMP) is an elegant, computationally efficient, and nearly conservative method for the solution of two-phase flow problems. It was developed in one dimension for the stiffened gas equation of state (EOS) and one-step time-discretization algorithms. It naturally extends to three dimensions but its extension to higher-order, multi-step time-discretization schemes is not straightforward. Furthermore, the original GFMP and many other ghost fluid methods fail to handle the large density and pressure jumps that are encountered in underwater implosions. Therefore, the GFMP is generalized in this work to an arbitrary EOS and multi-fluid problems with multiple EOSs. It is also extended to three dimensions and developed for higher-order, multi-step time-discretization algorithms. Furthermore, this method is equipped with an exact two-phase Riemann solver for computing the fluxes across the material interface without crossing it. This aspect of the computation is a departure from the standard approach for computing fluxes in ghost fluid methods. It addresses the stiff nature of the two-phase air/water problem and enables a better handling of the large discontinuity of the density at the air/water interface. As the original GFMP, the proposed method is contact preserving, computationally efficient, and nearly conservative. Its superior performance in the presence of large density and pressure jumps is demonstrated for shock-tube problems. Its practicality and accuracy are also highlighted with the three-dimensional simulation of the implosion of an air-filled and submerged glass sphere.

Keywords: GFM for the poor; Riemann solver; Two-phase compressible flow; Underwater implosion

1. Introduction

The numerical simulation of compressible multi-medium flows arises in many applications including underwater bubble dynamics, shock wave interactions with material discontinuities, and combustion, to name only a few. The motivation for the present work is the accurate and computationally efficient prediction of the

* Corresponding author. Tel.: +1 650 723 3840; fax: +1 650 725 3525.
E-mail addresses: cfarhat@stanford.edu (C. Farhat), arallu@stanford.edu (A. Rallu), ssriram@stanford.edu (S. Shankaran).

0021-9991/$ - see front matter © 2008 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2008.04.032

bubble dynamics and pressure signatures generated by underwater implosions. The large size of the bubbles and their energy content result in strong shock and expansion waves. Typically, these propagate through air (vapor) and water and can be reflected or refracted off the air/water interface, which calls for modeling water as a compressible fluid. The ratio of water and air densities (≈ 1000) is such that the air/water interface is well approximated by a free surface where the gas can only apply a pressure on the liquid.

Underwater implosions result in bubbles whose characteristic size is considerably larger than that of bubbles obtained in liquid suspensions. Hence, such bubbles are less affected by surface tension and viscous forces and therefore their dynamics can be modeled by the Euler equations. The numerical solution of these equations for a single fluid has reached a state of considerable maturity. Godunov-type schemes [1] and extensions to higher-order semi-discretizations [2,3] are often the methods of choice for achieving crisp shock resolution in space. A variety of explicit and implicit temporal discretizations have been developed for these schemes and for both steady and unsteady problems. However, initial attempts [4,5] at the extension of these numerical algorithms to multi-fluid problems suffered from numerical instabilities and oscillations, primarily around the material interface.

The common multi-fluid solution methods published in the literature use either a Lagrangian or an Eulerian method. In a Lagrangian method, the computational mesh moves and distorts with the material interface. The interface itself is convected with the local fluid velocity and can be resolved sharply by controlling the numerical diffusion around it. However, if the problem induces large displacements of the interface, the resulting mesh distortions can adversely affect the accuracy and stability of the numerical solution process and often make the Lagrangian approach unpractical.

Eulerian methods use a fixed mesh and usually carry an auxiliary equation for tracking or capturing the material interface. In the volume of fluid (VOF) approach [6], each computational cell is assumed to possibly contain a mixture of both fluids and the volume occupied by each fluid is represented by the volume fraction. The evolution of this fraction is governed by a transport equation where the speed of propagation is determined by the local fluid velocity. The VOF method has been predominantly used for incompressible flows where the knowledge of the interface position is sufficient to recover the density field. For compressible flows, recovering the density field and the internal energies in a cell containing both fluids does not seem to be an obvious task. Another class of Eulerian methods that has found wide-spread usage is based on the level-set equation [7] for capturing the interface. This equation falls under the general class of Hamilton–Jacobi equations. It can also be viewed as a particular case of the transport equation. It governs the evolution of the zero of the level-set function which marks the interface. The level-set equation naturally allows for merger and break-up of the interface, is relatively straightforward to implement, does not incur a significant computational overhead and therefore is an attractive candidate approach for interface capturing. In addition to the volume (or mass) fraction model and the level-set approach, a γ model – where γ denotes the ratio of specific heats for a given gas – has also been suggested for capturing the evolution of the interface [8]. Here again, the evolution equation is a transport equation. In principle, any function of γ can be used as the interface marker, but the ratio $1/(\gamma - 1)$ has been shown to favor a non-oscillatory numerical solution of the pressure at the material interface [5,9,8,10].

Whether in the context of a Lagrangian or Eulerian approach, the numerical treatment of the Euler equations at the material interface still needs to be addressed. Early attempts at the numerical solution of multi-medium flows in an Eulerian setting resulted either in mass fractions outside the valid range of $[0, 1]$, or in pressure oscillations across the material interface. These oscillations are present even in first-order, monotonicity preserving schemes. To suppress them, particular forms of the discretization of the conserved variables and/or particular functions for capturing the evolution of the interface have been proposed [8].

Amid attempts to prevent pressure-oscillations in multi-fluid calculations, the ghost fluid method (GFM) was developed as a more economical alternative solution method [7]. The main feature of this method is its simplicity: it allows multi-fluid computations to be performed in the vicinity of the material interface as if they pertained to a single medium domain. Given an interface capturing technique – usually, the level-set method – the GFM exploits the concept of ghost and real fluid cells and manages them with an overlapping Schwarz-like numerical procedure. In the material interface region, it sets the values of the pressure and normal velocity in the ghost fluid cells to those in the real fluid cells. To eliminate an otherwise spurious “over-heating” phenomenon, it computes the density of the ghost fluid using an isobaric technique [11]. In its basic form, the

GFM is non conservative [12]. However, it can be equipped with an *a posteriori* correction procedure that first measures the *discrete* conservation errors generated in the neighborhood of the material interface during a given time-step, then offsets them using an error redistribution technique. This correction procedure was proposed in [13] where it was applied to stiff detonation problems. Unfortunately, the GFM fails to solve some air/water problems of interest. For such problems, it either delivers inaccurate results because of spurious oscillations, or simply fails to deliver any result [14]. An improved version of this method incorporating in a one-step time-integration scheme an approximate two-phase Riemann solver at the material interface that assumes either a two-shock or two-rarefaction wave structure was proposed in [14] for the solution of gas/water problems and illustrated with simple 1D and 2D calculations. Like the original GFM, this enhanced version relies on the isobaric technique for eliminating the spurious over-heating phenomenon. For two- and three-dimensional applications, this isobaric fix requires the solution of yet another auxiliary partial differential equation (PDE) [7] and therefore increases further the computational complexity of the method. Most recently, the approximate Riemann solver of [14] was replaced in [17] by an exact version to eliminate the need for the isobaric fix.

The ‘‘overlapping’’ aspect of the GFM induces a combined storage and computational overhead that is application dependent. The ghost fluid method for the poor (GFMP) [15] is a variant method which avoids most of this overhead by computing two numerical fluxes at the material interface: one using the thermodynamic parameters of the fluid on one side of the interface, and another one using the thermodynamic parameters of the other fluid medium. It is an elegant, computationally efficient, and nearly conservative method in the sense that it conserves all conservative variables except the energy across the material interface. The GFMP was developed in [15] for one-dimensional problems using a one-step explicit time-integration algorithm and assuming that each given fluid is a stiffened gas. It involves a subtle but crucial conversion from conservative to primitive variables (and vice-versa) before and after advancing in time the solution of the level-set equation. Its extension to multiple dimensions is straightforward. However, as it will be shown in this paper, its extension to higher-order multi-step time-integrators requires a careful sequencing of its computational steps. More importantly, the GFMP does not apply as formulated in [15] to multi-fluid problems involving either an equation of state (EOS) that is different from that of a stiffened gas, or different EOSs on the two sides of a material interface. Hence, applying the GFMP to air/water problems calls for either modeling both fluid media as stiffened gases, or generalizing this method to a larger number of EOSs and extending it to multi-fluid problems with multiple EOSs. However, even when both water and air are modeled as stiffened gases, it is the authors' experience that the GFMP, like the GFM, fails to solve most air/water problems of interest. It is the authors' opinion that these and other related observations that were also documented in [14] can be explained by the fact that inherent to the GFM and GFMP is a flux computation approach that crosses the material interface. Using data from both sides of a discontinuity in a spatial discretization scheme leads to a computational method that is usually not robust in the presence of a large jump such as that encountered for the density at an air/water interface.

When an underwater explosion occurs, the resulting energy release creates an expanding gas bubble which undergoes a multiple expansion (explosion)/collapse (implosion) process and continuously loses energy until it breaks down. The bubble oscillation process, the initiation process, the source of the instability leading to bubble collapse, and the energy loss mechanism are not completely understood. Extensive experimental [18] and computational [19] investigations have been conducted to develop a better understanding of these phenomena. Early computational studies were reported in [19] using a simplified, one-dimensional computational model in which water was modeled as a compressible fluid and the bubble was assumed to maintain a spherical shape. The gas inside the bubble was assumed to have a polytropic EOS and to undergo isentropic changes [19]. Excellent correlation with experimental data was obtained for the bubble's radius time-history. In [20], the simplified model developed in [19] was modified to account for energy losses in the gas medium. The resulting computational model produced better correlations with experimental data for the amplitude and phase of the bubble oscillations [20]. However, because they assume spherical symmetry, both models developed in [19] and [20] cannot properly account for bubble migration due to buoyancy. Furthermore, they cannot account for shape changes during the collapse phase when the bubble motion is unstable. Three-dimensional simulations are required for capturing these important details.

Given the context set above, the main objectives of this paper are three-fold: (a) to generalize the GFMP to multi-fluid problems with multiple EOSs and to extend it to higher-order time-discretizations, (b) to enable its application to air/water problems by enhancing its robustness for two-phase problems with large contact discontinuities and strong pressure jumps at the material interface, and (c) to demonstrate its potential for the three-dimensional simulation of underwater implosions. To this effect, the remainder of this paper is organized as follows.

In Section 2, the governing equations of the two-phase air/water problems of interest are presented and discussed. In Section 3, the GFMP is briefly overviewed. In Section 4, the GFMP is generalized to multi-fluid problems with multiple EOSs. Its robustness with respect to a large discontinuity of the density and a strong pressure jump at the material interface is enhanced in Section 5 via the incorporation of an exact, local, one-dimensional, two-phase Riemann solver for computing the interfacial fluxes without traversing the zone of discontinuity. The resulting multi-fluid method is referred to as the GFMP–ERS (for GFMP with Exact Riemann Solver). In Section 6, a computational framework for extending the GFMP–ERS to higher-order multi-step time-discretization algorithms is proposed. In Section 7, the higher-order GFMP–ERS is evaluated using simple benchmark problems, some of which include representative features of underwater implosions. Then, the potential of the GFMP–ERS is illustrated with the three-dimensional simulation of the implosion of an air-filled and submerged glass sphere, and the favorable comparison of the obtained numerical results to experimental as well as other numerical data. Finally, Section 8 concludes this paper.

2. Governing equations

2.1. Eulerian flow

As already mentioned, underwater implosions generate bubbles that are usually considerably larger than those encountered in liquid suspensions. Hence, such bubbles are less affected by surface tension and viscous effects. For this reason, their dynamics is modeled in this paper by the Euler equations written in the familiar conservation form

$$\frac{\partial w}{\partial t} + \nabla \cdot \mathcal{F}(w) = 0, \tag{1}$$

where t, $w(X,t), X = (x,y,z)$, and $\mathcal{F}$ denote time, the conservative fluid state vector, space, and the convective flux vector, respectively. The initial condition for the above PDE is written as:

$$w(X,0) = g(X) \tag{2}$$

and its boundary conditions are not specified here as they are problem dependent.

2.2. Equations of state

Two different EOSs are considered in this paper for modeling compressible water: (1) the stiffened gas equation, and (2) Tait's equation.

2.2.1. The stiffened gas equation

The stiffened gas equation is a generalization of the perfect gas EOS. It can be written as

$$(\gamma - 1)\rho e = p + \gamma\pi, \tag{3}$$

where ρ, e, and p denote the density, internal energy per unit mass and pressure, respectively, and γ and π are constants that need to be specified. This EOS is versatile: it has been used for modeling gas, liquid, and solid media. The constants γ and π are set so that the speed of sound in the medium of interest, c, is correctly predicted using this EOS and the definition

$$c = \sqrt{\left.\frac{\partial p}{\partial \rho}\right|_s}, \tag{4}$$

where s denotes the entropy.

To evaluate the sound speed c, the following thermodynamic equations are first recalled

$$e = c_v T \quad T\,\mathrm{d}s = \mathrm{d}e + p\,\mathrm{d}\left(\frac{1}{\rho}\right), \tag{5}$$

where c_v denotes the specific heat at constant volume. From Eqs. (3) and (5) it follows that

$$T\,\mathrm{d}s = \mathrm{d}e + p\mathrm{d}\left(\frac{1}{\rho}\right) = \left(\frac{p+\gamma\pi}{(\gamma-1)\rho c_v}\right)\mathrm{d}s \tag{6}$$

$$= \left(\frac{1}{(\gamma-1)\rho}\right)\mathrm{d}p - \left(\frac{p+\gamma\pi}{(\gamma-1)\rho^2}\right)\mathrm{d}\rho - \left(\frac{p}{\rho^2}\right)\mathrm{d}\rho \tag{7}$$

$$= \left(\frac{1}{(\gamma-1)\rho}\right)\mathrm{d}p - \left(\frac{\gamma(p+\pi)}{(\gamma-1)\rho^2}\right)\mathrm{d}\rho. \tag{8}$$

Hence

$$\mathrm{d}s = \left(\frac{c_v}{p+\gamma\pi}\right)\mathrm{d}p - \left(\frac{\gamma c_v(p+\pi)}{(p+\gamma\pi)\rho}\right)\mathrm{d}\rho \tag{9}$$

and the speed of sound in a stiffened gas is given by

$$c = \sqrt{\left.\frac{\partial p}{\partial \rho}\right|_s} = \sqrt{\frac{\gamma(\pi+p)}{\rho}}. \tag{10}$$

For water, the following numerical values of π and γ are often found in the literature

$$\pi = 6.0 \times 10^8\ \mathrm{Pa} \quad \gamma \in \{4.4, 5.5, 7.0\}. \tag{11}$$

Remark 1. For $\pi = 0$, Eq. (3) simplifies to the perfect gas equation.

2.2.2. Tait's equation

The Tait equation of state models a liquid such as water as a compressible, barotropic liquid whose bulk modulus is an affine function of pressure. Hence, this EOS involves only the density and pressure variables. However, it is a highly non-linear equation of the form

$$p = \eta + \alpha\rho^\beta, \tag{12}$$

where η, α, and β are three constants that can be determined from the assumption that the bulk modulus K of the liquid is an affine function of pressure determined by two constants k_1 and k_2 and from the knowledge of a reference state (ρ_0, p_0). Hence

$$k_1 + k_2 p = K = \rho\frac{\mathrm{d}p}{\mathrm{d}\rho} = \beta\alpha\rho^\beta = \beta(p-\eta), \tag{13}$$

which gives

$$\eta = -\frac{k_1}{k_2} \quad \beta = k_2. \tag{14}$$

Furthermore, writing $p_0 = p(\rho_0)$ gives

$$\alpha = \frac{p_0 + \frac{k_1}{k_2}}{\rho_0^{k_2}}. \tag{15}$$

In the literature, the following numerical values are often found for water

$$k_1 = 2.07 \times 10^9\ \mathrm{kg} \times \mathrm{m}^{-3} \times \mathrm{s}^{-2} \quad k_2 = 7.15. \tag{16}$$

When water is modeled by Tait's EOS, the speed of sound in this fluid is given by

$$c = \sqrt{\frac{\mathrm{d}p}{\mathrm{d}\rho}} = \sqrt{\alpha\beta\rho^{\beta-1}} = \sqrt{\left(\frac{k_2 p_0 + k_1}{\rho_0}\right)\left(\frac{\rho}{\rho_0}\right)^{k_2-1}}. \tag{17}$$

For $\rho_0 = 1000\ \mathrm{kg} \times \mathrm{m}^{-3}$ and $p_0 = 10^6$ Pa, the predicted speed of sound is

$$c_0 = c(\rho_0) = \sqrt{\frac{k_2 p_0 + k_1}{\rho_0}} = 1441.23\ \mathrm{m} \times \mathrm{s}^{-1} \tag{18}$$

as expected.

Remark 2. When a fluid is modeled by Tait's EOS, the energy equation becomes decoupled from the continuity and momentum equations.

Remark 3. Consider a stiffened gas that is undergoing an isentropic transformation. Let T, s, and h denote temperature, entropy and enthalpy, respectively. From the second principle of thermodynamics it follows that

$$0 = T\,\mathrm{d}s = \mathrm{d}h - \frac{\mathrm{d}p}{\rho} = \mathrm{d}e + p\,\mathrm{d}\left(\frac{1}{\rho}\right). \tag{19}$$

From the stiffened gas Eq. (3) and its differentiation it follows that

$$\mathrm{d}e = \left(\frac{1}{(\gamma-1)\rho}\right)\mathrm{d}p - \left(\frac{p+\gamma\pi}{(\gamma-1)\rho^2}\right)\mathrm{d}\rho. \tag{20}$$

Substituting Eq. (20) into Eq. (19) yields after expansion

$$0 = \left(\frac{1}{(\gamma-1)\rho}\right)\mathrm{d}p - \left(\frac{p+\gamma\pi}{(\gamma-1)\rho^2}\right)\mathrm{d}\rho - \left(\frac{p}{\rho^2}\right)\mathrm{d}\rho = \left(\frac{1}{p+\pi}\right)\mathrm{d}p - \left(\frac{\gamma}{\rho}\right)\mathrm{d}\rho. \tag{21}$$

From the integration of the above result, it follows that

$$\exists k_3 \in \mathbb{R}/p = k_3\rho^{\gamma} - \pi, \tag{22}$$

which shows that Tait's EOS (12) corresponds to the particular case of an isentropic stiffened gas EOS with $\gamma = \beta = k_2$, $\pi = -\eta = \frac{k_1}{k_2}$ and $k_3 = \alpha = (p_0 + k_1/k_2)/\rho_0^{k_2}$.

2.3. Conservative level-set approach

In this work, the level-set method [21] is adopted for capturing the material interface. More specifically, the level-set equation is written in conservation form as follows:

$$\frac{\partial(\rho\phi)}{\partial t} + \nabla \cdot (\rho u \phi) = 0, \tag{23}$$

where ρ and u are the density and velocity vector of the fluid, respectively, and ϕ is a function initialized to the distance between each grid point and the material interface. Hence, $\phi = 0$ captures the interface. To ensure that the signed distance function property of ϕ is preserved during the computations, ϕ can be periodically re-initialized using the algorithm proposed in [28].

2.4. Semi-discretization

The finite volume (FV) method is chosen here to semi-discretize all PDEs introduced above. Given a Computational Fluid Dynamics (CFD) grid, this method transforms the Euler flow equations into

$$\frac{\partial w}{\partial t} + \int_{C_i} \nabla \cdot \mathcal{F}(w)\,\mathrm{d}V = 0, \tag{24}$$

where C_i is the volume of the cell or control volume surrounding the ith grid point. Throughout this paper, the control volumes are assumed to be constructed by connecting the centroids of the triangular faces of the tetrahedra and the midpoints of the edges (Fig. 1). The resulting grid is referred to as the "dual" CFD grid.

Using integration by parts, the volume integral in Eq. (24) is converted to a surface integral across the boundary of the control volume and approximated by

$$F_i(W) = \sum_{j\in\kappa(i)} \text{mes}(\partial C_{ij})\Phi_{ij}(W_i, W_j, n_{ij}), \tag{25}$$

where $\kappa(i)$ is the set of vertices connected by an edge to vertex i, ∂C_{ij} is the segment of the boundary of C_i that intersects edge $i-j$, $\text{mes}(\partial C_{ij})$ is its measure, Φ_{ij} denotes the numerical flux function across ∂C_{ij}, W_i denotes the discrete fluid state vector at vertex i and n_{ij} is the unitary outer normal to ∂C_{ij}.

To achieve second-order spatial accuracy and address in this case potential numerical oscillations, the FV scheme is equipped with the MUSCL (Monotonic Upwinding Scheme for Conservation Laws) interpolation procedure [2] and a slope limiter. In this case, the approximation (25) is replaced by

$$F_i(W) = \sum_{j\in\kappa(i)} \text{mes}(\partial C_{ij})\Phi_{ij}(W_{ij}, W_{ji}, n_{ij}), \tag{26}$$

where W_{ij} and W_{ji} are two extrapolated and limited fluid state vectors. Note that for the MUSCL interpolation procedure, the computation of the gradients at a node is based on the state values of its neighbours that lie on the same side of the material interface, but not on the state values of its neighbours that lie in another fluid.

3. The ghost fluid method for the poor

The GFMP proposed in [15] is a computationally lighter alternative to the original GFM. It trades most of the redundant storage and computational requirements of the GFM in the vicinity of the material interface with the evaluation of two numerical fluxes: one using the thermodynamic parameters of the fluid on one side of the interface, and another one using the thermodynamic parameters of that on the other side.

When the level-set method is chosen for capturing the material interface between two fluids modeled by the stiffened gas EOS, Roe's solver [22] is chosen for computing the numerical flux functions in Eq. (25) and time-variation is discretized by the forward Euler scheme using a constant time-step Δt, the key computational steps

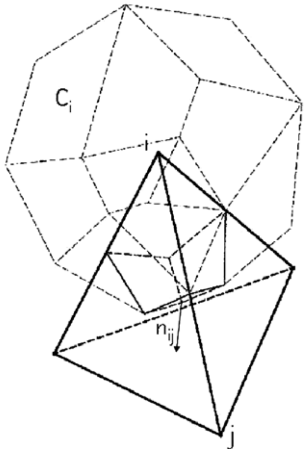

Fig. 1. Control volume (lighter lines) in an unstructured tetrahedral (heavier lines) mesh (only one of the tetrahedra needed to construct the graphically depicted control volume is shown).

of the GFMP between time $t^n = n\Delta t$ and time $t^{n+1} = (n+1)\Delta t$ can be summarized as follows using the notation adopted in this paper:

(1) Capture the material interface by checking the product of the values of the level-set function ϕ at vertices i and j of edge $i - j$. A positive value indicates that edge $i - j$ does not cross the material interface, in which case the numerical flux function is computed as usual. On the other hand, a negative value indicates that this edge crosses the material interface. In this case, compute two different fluxes: one using the coefficients γ and π of the stiffened gas where node i lies and one using those of the stiffened gas where node j lies. In algorithmic words, this can be written as follows:

$$\begin{aligned} &\text{If } \phi_i^n \times \phi_j^n > 0, \quad \Phi_{ij} = \text{Roe}\Big(W_i^n, W_j^n, (\gamma_i = \gamma_j, \pi_i = \pi_j), n_{ij}\Big), \\ &\text{If } \phi_i^n \times \phi_j^n \leqslant 0, \quad \text{then} \\ &\qquad\qquad \Phi_{ij} = \text{Roe}(W_i^n, W_j^n, \gamma_i, \pi_i, n_{ij}), \\ &\qquad\qquad \Phi_{ji} = \text{Roe}(W_j^n, W_i^n, \gamma_j, \pi_j, n_{ji}), \end{aligned} \tag{27}$$

where $n_{ji} = -n_{ij}$, γ_i and π_i denote the γ and π coefficients of the stiffened gas where node i lies, respectively, and γ_j and π_j denote the γ and π coefficients of the stiffened gas where node j lies, respectively.

(2) Time-advance the solution of the multi-fluid problem to compute a temporary value $\widetilde{W}_i^{n+1}$ of W_i^{n+1}

$$\widetilde{W}_i^{n+1} = W_i^n - \Delta t \sum_{j \in \kappa(i)} \text{mes}(\partial C_{ij}) \Phi_{ij}(W_i^n, W_j^n, n_{ij}). \tag{28}$$

(3) Using the value of the level-set function at time t^n, ϕ^n, unpack the conservative fluid state vector $\widetilde{W}^{n+1}$ – that is, convert it to a vector V^{n+1} of primitive variables to obtain $\tilde{\rho}^{n+1}$ and $\tilde{u}^{n+1}$

$$\widetilde{W}^{n+1} \xrightarrow{\phi^n} V^{n+1} \rightarrow (\tilde{\rho}^{n+1}, \tilde{u}^{n+1}). \tag{29}$$

(4) Compute ϕ^{n+1} by time-advancing the solution of the level-set Eq. (23) using the forward Euler scheme and the values of $\tilde{\rho}^{n+1}$ and $\tilde{u}^{n+1}$ stored in V^{n+1}

$$(\phi^n, \tilde{\rho}^{n+1}, \tilde{u}^{n+1}) \rightarrow \phi^{n+1}. \tag{30}$$

(5) Using the updated value of the level-set function ϕ^{n+1}, pack V^{n+1} – that is, transform it into the conservative fluid state vector W^{n+1}

$$V^{n+1} \xrightarrow{\phi^{n+1}} W^{n+1}. \tag{31}$$

The GFMP summarized above was proposed in [15] in one-dimensional form using a one-step explicit time-integration scheme. While it has been used mostly with Roe's flux [22], it is equally applicable with any flux that preserves a contact discontinuity – that is, preserves a uniform pressure and uniform density input. Its generalization to multiple dimensions is straightforward (for example, Eq. (28) is already written in multiple dimensions). Its generalization to a non stiffened gas EOS and its formulation for multi-fluid problems with different EOSs on both sides of a material interface are also relatively simple. However, its extension to higher-order, multi-step, time-integration schemes is more subtle as it requires a careful sequencing of the computational steps outlined above. These issues are discussed in Sections 4 and 5, respectively.

4. Generalization to an arbitrary EOS and multi-fluid problems with multiple EOSs

4.1. Generalization to an arbitrary EOS

The original GFMP is generalized here to an arbitrary EOS characterized by n_q parameters q_k, $k = 1, \ldots, n_q$, by replacing Step (1) and Eq. (28) of the algorithm described in Section 3 by

$$\begin{aligned}
&\text{If } \phi_i^n \times \phi_j^n > 0, \quad \Phi_{ij} = \text{Roe}\Big(W_i^n, W_j^n, (q_{k_i} = q_{k_j},\ k = 1, \ldots, n_q),\ n_{ij}\Big),\\
&\text{If } \phi_i^n \times \phi_j^n \leqslant 0, \quad \text{then}\\
&\qquad\qquad \Phi_{ij} = \text{Roe}\Big(W_i^n, W_j^n, (q_{k_i}, k = 1, \ldots, n_q), n_{ij}\Big),\\
&\qquad\qquad \Phi_{ji} = \text{Roe}\Big(W_j^n, W_i^n, (q_{k_j}, k = 1, \ldots, n_q), n_{ji}\Big),
\end{aligned} \tag{32}$$

where q_{k_i} and q_{k_j} denote the values of the parameters of the given EOS for the fluids where node i and node j lie, respectively. This generalization assumes that Roe's solver [22] can be extended to the EOS of interest. This is true, for example, for Tait's equation described in Section 2.2.2.

4.2. Generalization to multi-fluid problems with multiple EOSs

In order to address multi-medium flow problems with multiple EOSs – and more specifically, the case where the two fluids on the left and right sides of a material interface are governed by different EOSs – the GFMP method is generalized here by replacing Step (1) and Eq. (28) of the original GFMP by

$$\begin{aligned}
&\text{If } \phi_i^n \times \phi_j^n > 0, \quad \Phi_{ij} = \text{Roe}\Big(W_i^n, W_j^n, (\text{EOS}_i = \text{EOS}_j), n_{ij}\Big),\\
&\text{If } \phi_i^n \times \phi_j^n \leqslant 0, \quad \text{then}\\
&\qquad\qquad \Phi_{ij} = \text{Roe}(W_i^n, W_j^n, \text{EOS}_i, n_{ij}),\\
&\qquad\qquad \Phi_{ji} = \text{Roe}(W_j^n, W_i^n, \text{EOS}_j, n_{ji}),
\end{aligned} \tag{33}$$

where EOS_i and EOS_j denote the EOS governing the fluids where node i and node j lie, respectively.

The reader can observe that Eq. (34) include Eq. (33) as a particular case. Therefore, Eq. (34) are adopted to describe the GFMP in the general case of multi-fluid problems with arbitrary and/or multiple EOSs.

5. The ghost fluid method for the poor with an exact two-phase Riemann solver

A large number of numerical experiments performed by the authors have revealed that the GFM and the GFMP are not capable of solving some air/water flow problems, particularly at practical mesh resolutions. More specifically, the authors have found that for this class of two-phase flow applications, the GFMP tends to predict inexact pressures and densities, most of which are negative on the water side of the material interface. The use of a positive flux such as the Lax–Friedrichs flux instead of the Roe flux was not found to overcome this problem (for example, see Section 7.1.3). The authors believe that the main reason why the GFM and GFMP fail to perform in the presence of a strong discontinuity at a material interface as in the case of an air/water interface where $\rho_{\text{water}}/\rho_{\text{air}} = 1000$, is that both methods utilize data from both sides of this discontinuity when computing the interfacial fluxes (for example, see Section 3, Step (1), Eq. (28)). In other words, both of the GFM and GFMP utilize data from both sides of the discontinuity when discretizing the spatial terms of the governing equations in the vicinity of the material interface. Usually, this is not an effective approach, particularly for strong discontinuities and relatively low-order spatial discretizations.

Here, it is proposed to address the robustness issue outlined above by modifying the interfacial flux computation performed in Step (1) of the generalized GFMP described in Section 4 to use a new fluid state vector $W_i^{\mathcal{R}^n}$ that is on the same side of the material interface as W_i and a new fluid state vector $W_j^{\mathcal{R}^n}$ that is on the same side of the material interface as W_j, as follows:

$$\begin{aligned}
&\text{If } \phi_i^n \times \phi_j^n > 0, \quad \Phi_{ij} = \text{Roe}\Big(W_i^n, W_j^n, (\text{EOS}_i = \text{EOS}_j),\ n_{ij}\Big),\\
&\text{If } \phi_i^n \times \phi_j^n \leqslant 0, \quad \text{then}\\
&\qquad\qquad \Phi_{ij} = \text{Roe}(W_i^n, W_i^{\mathcal{R}^n}, \text{EOS}_i, n_{ij}),\\
&\qquad\qquad \Phi_{ji} = \text{Roe}(W_j^n, W_j^{\mathcal{R}^n}, \text{EOS}_j, n_{ji}).
\end{aligned} \tag{34}$$

In Eq. (34) and throughout the remainder of this paper, $W_i^{\mathcal{R}^n}$ and $W_j^{\mathcal{R}^n}$ denote the conservative fluid state vectors associated with the *exact* solution at the interface from the sides where node *i* and node *j* lie, respectively, of the following one-dimensional two-phase Riemann problem

$$\begin{aligned} &\frac{\partial w}{\partial t}+\frac{\partial \mathcal{F}}{\partial \xi}(w)=0, \\ &w(\xi,0)=\begin{cases} W_i^n & \text{if } \xi \leqslant 0, \\ W_j^n & \text{if } \xi > 0, \end{cases} \end{aligned} \tag{35}$$

where ξ is the abscissa along the edge $i-j$ that crosses the interface and $\xi=0$ at this initial interface (see Fig. 2).

It is noted that while the formulation of problem (35) contains data from both sides of the material interface, the solution of this problem does not involve any spatial discretization since it is carried out analytically. Hence, unlike in the original GFM and GFMP, the computation of the interfacial fluxes proposed in Eq. (34) does not cross the material interface.

The generalized GFMP described in Section 4 and equipped as proposed above with an exact, local, one-dimensional, two-phase Riemann solver for the computation of the interfacial fluxes is referred to in the remainder of this paper as the GFMP–ERS (for GFMP with Exact Riemann Solver). To keep this paper as self-contained as possible, the important aspects of the exact solution of the Riemann problem (35) are described in Appendix A of this paper for both cases of the stiffened gas EOS and Tait's EOS.

Next, a rationale for the proposed interfacial flux computation (34) is presented. Then, some important implementational details are described before the proposed GFMP–ERS is described in details. Finally, the aforementioned implementational details are justified by proving an important mathematical property of the new GFMP–ERS.

5.1. Rationale

Let EOS$_i$ (EOS$_j$) denote the equation of state on the left (right) side of a material interface where node *i* (*j*) lies. The rationale for the proposed flux computation described in Eq. (34) is provided by the structure of the solution of the two-phase Riemann problem (35). This solution is composed of four constant states W_i, $W_i^{\mathcal{R}}$, $W_j^{\mathcal{R}}$ and W_j separated by non-linear waves and a contact discontinuity.

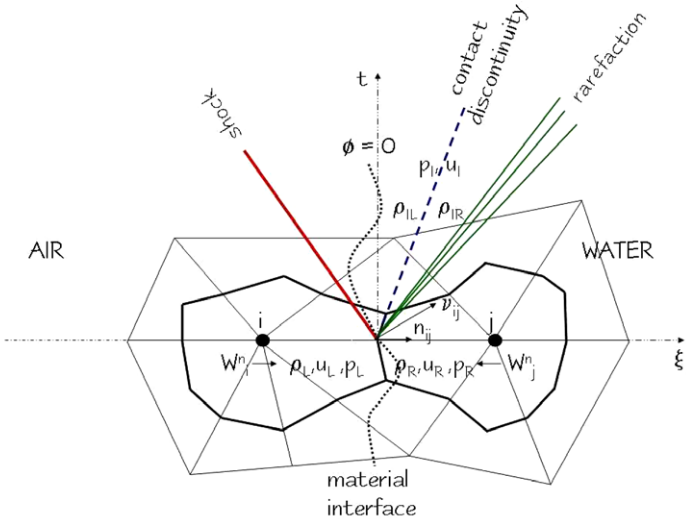

Fig. 2. Illustration of the local, one-dimensional, two-phase Riemann problem on a two-dimensional grid. (Subscripts *I*, *L*, and *R* denote the material interface and the media at its left and right sides, respectively. n_{ij} denotes the normal to the control volume at the face between nodes *i* and *j*. v_{ij} denotes the normal to the material interface at the same point).

Consider first the following one-phase Riemann problem associated with a fluid whose EOS is EOS_i

$$\begin{aligned}&\frac{\partial w}{\partial t}+\frac{\partial \mathcal{F}}{\partial \xi}(w)=0\\ &w(\xi,0)=\begin{cases}W_i^n & \text{if } \xi \leqslant 0\\ W_i^{\mathcal{R}} & \text{if } \xi>0\end{cases}\end{aligned}\tag{36}$$

and then the similar one-phase Riemann problem associated with a fluid whose EOS is EOS_j

$$\begin{aligned}&\frac{\partial w}{\partial t}+\frac{\partial \mathcal{F}}{\partial \xi}(w)=0,\\ &w(\xi,0)=\begin{cases}W_j^{\mathcal{R}} & \text{if } \xi \leqslant 0,\\ W_j^n & \text{if } \xi>0.\end{cases}\end{aligned}\tag{37}$$

The solution of problem (36) is composed of two constant states, W_i^n and $W_i^{\mathcal{R}}$, and a single non-linear wave connecting them. The restriction to $\xi<\xi_{\text{contact}}$ of this solution, where ξ_{contact} denotes the coordinate of the contact discontinuity (which has zero strength in this case), is identical to the restriction to $\xi<\xi_{\text{contact}}$ of the solution of the original two-phase Riemann problem (35). Similarly, the restriction to $\xi>\xi_{\text{contact}}$ of the solution of problem (37) is identical to the restriction to $\xi>\xi_{\text{contact}}$ of the solution of the original two-phase Riemann problem (35). These two results hold for any EOS_i, EOS_j, W_i, and W_j. Therefore, utilizing $W_i^{\mathcal{R}}$ ($W_j^{\mathcal{R}}$) in the Roe flux function associated with node i (j), which itself is an approximate Riemann solver, gives the sought-after accuracy and robustness effects.

5.2. Some implementational details

For the purpose of solving the Riemann problem (35), the intersection of the instantaneous position of the material interface and the dual CFD grid is assumed to coincide with the intersection of the boundaries of the control volumes and the edges of the original CFD grid (see Fig. 2). This facilitates the computation of the interfacial fluxes but raises the issue of which normal to use in this computation: that to the material interface or that to the corresponding face of the control volume, since both are available but are different except for one-dimensional problems. In order to remain consistent with the principles of the finite volume method, the normal to the face of the control volume, n_{ij}, is always chosen here for computing a flux Φ_{ij}. However, it will be shown in Section 5.4 that in order to ensure that the GFMP–ERS is contact preserving, the input and output entities of the Riemann solver (see Fig. 2) must be computed using the normal to the material interface. This normal between two connected grid points i and j on both sides of the material interface can be evaluated using the gradient of the level-set function as follows:

$$v_{ij}=\nabla\phi_{ij}\approx\frac{1}{2}(\nabla\phi_i+\nabla\phi_j),\tag{38}$$

where $\nabla\phi_k$, $k=i,j$ is the nodal gradient and can by computed by a least square technique such as that described in [24].

5.3. The GFMP–ERS multi-fluid method

Let the subscripts I, L, and R denote the material interface and the media at its left and right sides, respectively, n_{ij} denote the normal to the control volume at the face between nodes i and j, and v_{ij} denote the normal to the material interface at the same point (see Fig. 2).

First, the case of the simple forward Euler time-integrator is considered. In this case, the proposed GFMP-ERS multi-fluid method can be summarized as follows.

(1) Capture the material interface by checking the product of the values of the level-set function ϕ at vertices i and j of edge $i-j$. A positive value indicates that edge $i-j$ does not cross the material interface, in which case the numerical flux function is computed as usual. On the other hand, a negative value indi-

cates that this edge crosses the material interface, in which case two different fluxes are computed after a local, one-dimensional, two-phase Riemann problem along this edge is solved exactly. In algorithmic words, this can be written as follows:

If $\phi_i^n \times \phi_j^n > 0$,

$$\Phi_{ij} = \mathrm{Roe}\left(W_i^n, W_j^n, (\mathrm{EOS}_i = \mathrm{EOS}_j), n_{ij}\right),$$

If $\phi_i^n \times \phi_j^n \leqslant 0$,

- extract the density ρ_k^n, the velocity vector u_k^n, and the pressure p_k^n, from $W_k^n, k = i, j$
- decompose each of the velocity vectors $u_k^n, k = i, j$, into a normal component $u_{\nu_{ij_k}}^n$ and a tangential component $u_k^n - u_{\nu_{ij_k}}^n \nu_{ij}$ where $\nu_{ij} = \frac{\nabla\phi_{ij}}{|\nabla\phi_{ij}|}$ is computed using Eq. (38)
- solve exactly the two-phase Riemann problem (35) along the edge $i - j$ using ρ_k^n, $u_{\nu_{ij_k}}^n$, and p_k^n, $k = i, j$ as inputs and compute $\rho_{I_L}^n, \rho_{I_R}^n, u_I^n$ and p_I^n where $\rho_{I_L}, \rho_{I_R}, u_I$, and p_I have the same meaning as in Fig.2
- reconstruct the velocity vectors at both nodes i and j as $u_k^{\mathcal{R}^n} = u_k^n - u_{\nu_{ij_k}}^n \nu_{ij} + u_I^n \nu_{ij}$, $k = i, j$
- construct $W_i^{\mathcal{R}^n}$ and $W_j^{\mathcal{R}^n}$
- compute the two fluxes

$$\Phi_{ij} = \mathrm{Roe}\left(W_i^n, W_i^{\mathcal{R}^n}, \mathrm{EOS}_i, n_{ij}\right)$$

$$\Phi_{ji} = \mathrm{Roe}\left(W_j^n, W_j^{\mathcal{R}^n}, \mathrm{EOS}_j, n_{ji}\right) \tag{39}$$

(2) Time-advance the solution of the multi-fluid problem to compute a temporary value $\widetilde{W}_i^{n+1}$ of W_i^{n+1}

$$\widetilde{W}_i^{n+1} = W_i^n - \Delta t \sum_{j\in\kappa(i)} \mathrm{mes}(\partial C_{ij}) \Phi_{ij}(W_i^n, W_j^n, n_{ij}) \tag{40}$$

(3) Using the value of the level-set function at time t^n, ϕ^n, unpack the conservative fluid state vector $\widetilde{W}^{n+1}$ – that is, convert it to a vector V^{n+1} of primitive variables to obtain $\tilde{\rho}^{n+1}$ and $\tilde{u}^{n+1}$

$$\widetilde{W}^{n+1} \xrightarrow{\phi^n} V^{n+1} \rightarrow (\tilde{\rho}^{n+1}, \tilde{u}^{n+1}). \tag{41}$$

(4) Compute ϕ^{n+1} by time-advancing the solution of the level-set Eq. (23) using the forward Euler scheme and the values of $\tilde{\rho}^{n+1}$ and $\tilde{u}^{n+1}$ stored in V^{n+1}

$$(\phi^n, \tilde{\rho}^{n+1}, \tilde{u}^{n+1}) \rightarrow \phi^{n+1}. \tag{42}$$

(5) For each node k

if $\phi_k^{n+1}\phi_k^n \geqslant 0$, pack V_k^{n+1} using $\phi_k^{n+1} : V_k^{n+1} \xrightarrow{\phi_k^{n+1}} W_k^{n+1}$,

if $\phi_k^{n+1}\phi_k^n < 0$, set $W_k^{n+1} = W_k^{\mathcal{R}^n}$.

The reader can verify that the second substep within Step (5) above is not part of the original GFMP (for example, see Step (5) in Section 3). In Section 5.4 below, it is shown that the main purpose of this feature is to preserve the structure of the solution of the contact problem.

The extension of the GFMP–ERS summarized above to a higher-order, explicit or implicit, one-step time-integrator is straightforward. Essentially, the desired one-step time-integrator is introduced in Step (2) outlined above and all other steps of this multi-fluid method are kept unchanged.

5.4. Contact preserving property

Consider a material front with the contact conditions $u_L^n = u_R^n = u$, $p_L^n = p_R^n = p$, but $\rho_L^n \neq \rho_R^n$, where the subscripts L and R designate the left and right sides of the front. The nodes of the computational mesh can be divided in two groups: one where each node has as all its neighbors in the same fluid medium as itself, and one where each node has at least one neighboring node that lies in a different fluid medium than itself.

Consider a node i in the first group of nodes. Since all its neighbours are in the same fluid medium as itself, it follows that

$$\forall j \in \kappa(i) \quad W_j^n = W_i^n.$$

Hence

$$F_i(W^n) = \sum_{j\in\kappa(i)} \mathrm{mes}(\partial C_{ij})\Phi_{ij}(W_i^n, W_i^n, n_{ij}).$$

Assuming that the flux function is consistent, it follows that

$$F_i(W) = \sum_{j\in\kappa(i)} \mathrm{mes}(\partial C_{ij})\mathcal{F}(W_i)\cdot n_{ij} = \mathcal{F}(W_i)\cdot \sum_{j\in\kappa(i)} \mathrm{mes}(\partial C_{ij})n_{ij} = \mathcal{F}(W_i)\cdot \int_{\partial C_i} n\,\mathrm{d}s = 0,$$

since the surface of a control volume is closed, and therefore

$$W_i^{n+1} = W_i^n,$$

which implies that the state of contact is preserved.

Next, consider a node i in the second group of nodes described above, and consider a neighboring node j that belongs to a different fluid medium than that of node i. During the solution of the two-phase Riemann problem (35), the densities, normal velocities, and pressures at these nodes are input to the exact Riemann solver. If the normal velocities are computed using the normal to the material interface, the exact Riemann solver delivers the input itself as the solution. In this case, $\rho_{I_L}^n = \rho_L^n$, $\rho_{I_R}^n = \rho_R^n$, $u_I^n = u$, and $p_I^n = p$. Consequently, $W_i^{\mathcal{R}^n} = W_i^n$, $W_j^{\mathcal{R}^n} = W_j^n$

$$\begin{aligned}\Phi_{ij} &= \mathrm{Roe}(W_i^n, W_i^n, \mathrm{EOS}_i, n_{ij})\\ \Phi_{ji} &= \mathrm{Roe}(W_j^n, W_j^n, \mathrm{EOS}_j, n_{ji})\end{aligned}$$

and therefore

$$F_i(W) = \sum_{j\in\kappa(i)} \mathrm{mes}(\partial C_{ij})\Phi_{ij}(W_i, W_i, n_{ij}) = 0$$

and

$$W_i^{n+1} = W_i^n$$

for the same reasons as in the previous case. This concludes the proof that as long as the input to the exact Riemann solver is computed using the normal to the material interface – and not the normal to the control volumes – the GFMP–ERS is a contact preserving multi-fluid method.

Finally, given that the material interface moves in time, a node i on one side of the material interface at time t^n can become on the other side of this interface at t^{n+1}. To preserve the structure of the solution of the contact problem at t^{n+1}, W_i^{n+1} needs to be properly updated. This is done in Step (5) of the GFMP–ERS where W_i^{n+1} is overwritten by $W_j^{\mathcal{R}^n} = W_j^n$ in order to preserve the state of contact.

6. Extension to higher-order multi-step time-integrators

Extending the GFMP-ERS summarized in Section 5.3 to a higher-order, explicit or implicit, k-step time-integrator requires paying special attention to Step (3) of this method (see Section 5.3) – that is, the unpacking of the conservative fluid state vector $\widetilde{W}^{n+1}$. Straightforward extensions turned out to be numerically unstable. On the other hand, the following is a proposed extension which achieved excellent results for a large number of different multi-fluid problems benchmarked by the authors:

(1) Capture the material interface and compute the numerical fluxes

If $\phi^n_i \times \phi^n_j > 0$,

$$\Phi_{ij} = \mathrm{Roe}\left(W^n_i, W^n_j, (\mathrm{EOS}_i = \mathrm{EOS}_j), n_{ij}\right)$$

If $\phi^n_i \times \phi^n_j \leqslant 0$,

- extract the density ρ^n_k, the velocity vector u^n_k, and the pressure p^n_k from $W^n_k, k = i, j$
- decompose each of the velocity vectors $u^n_k, k = i, j$, into a normal component $u^n_{\nu_{ij_k}}$ and a tangential component $u^n_k - u^n_{\nu_{ij_k}}\nu_{ij}$ where $\nu_{ij} = \frac{\nabla\phi_{ij}}{|\nabla\phi_{ij}|}$
- solve exactly the two-phase Riemann problem (35) along the edge $i - j$ using $\rho^n_k, u^n_{\nu_{ij_k}}$, and p^n_k, $k = i, j$ as inputs and compute $\rho^n_{I_L}, \rho^n_{I_R}, u^n_I$ and p^n_I where $\rho_{I_L}, \rho_{I_R}, u_I$, and p_I have the same meaning as in Fig.2
- reconstruct the velocity vectors at both nodes i and j as $u^{\mathcal{R}^n}_k = u^n_k - u^n_{\nu_{ij_k}}\nu_{ij} + u^n_I\nu_{ij}, k = i, j$
- construct $W^{\mathcal{R}^n}_i$ and $W^{\mathcal{R}^n}_j$
- compute the two fluxes

$$\begin{aligned}\Phi_{ij} &= \mathrm{Roe}\left(W^n_i, W^{\mathcal{R}^n}_i, \mathrm{EOS}_i, n_{ij}\right)\\ \Phi_{ji} &= \mathrm{Roe}\left(W^n_j, W^{\mathcal{R}^n}_j, \mathrm{EOS}_j, n_{ji}\right)\end{aligned} \tag{43}$$

(2) Compute a temporary value $\widetilde{W}^{n+1}_i$ of W^{n+1}_i by time-advancing the solution of the multi-fluid problem using the chosen higher-order, k-step time-integrator

$$(W^{n-k+1}, \ldots, W^{n-1}, W^n) \rightarrow \widetilde{W}^{n+1} \quad \text{(explicit case)}, \tag{44}$$

$$(W^{n-k+1}, \ldots, W^{n-1}, W^n, W^{n+1}) \rightarrow \widetilde{W}^{n+1} \qquad \text{(implicit case)}. \tag{45}$$

(3) Using the values of the level-set function $\phi^{n-k+2}, \ldots, \phi^{n-1}$, and ϕ^n, unpack the conservative fluid state vectors $W^{n-k+2}, \ldots, W^{n-1}, W^n$, and $\widetilde{W}^{n+1}$ as follows:

$$\begin{array}{lcl} \widetilde{W}^{n+1} & \overset{\phi^n}{\rightarrow} & V^{n+1} \overset{\phi^n}{\rightarrow} (\tilde{\rho}^{n+1}, \tilde{u}^{n+1}) \\ W^n & \overset{\phi^n}{\rightarrow} & V^n \\ W^{n-1} & \overset{\phi^{n-1}}{\rightarrow} & V^{n-1} \\ & \vdots & \\ W^{n-k+2} & \overset{\phi^{n-k+2}}{\rightarrow} & V^{n-k+2}. \end{array} \tag{46}$$

(4) Compute ϕ^{n+1} by time-advancing the solution of the level-set Eq. (23) using the chosen higher-order, k-step time-integrator and the values of $\tilde{\rho}^{n+1}$ and $\tilde{u}^{n+1}$ stored in V^{n+1}

$$(\phi^{n-k+1}, \ldots, \phi^{n-1}, \phi^n, \tilde{\rho}^{n+1}, \tilde{u}^{n+1}) \rightarrow \phi^{n+1}. \tag{47}$$

(5) Using the updated value of the level-set function ϕ^{n+1}, pack appropriately all of $V^{n-k+2}, \ldots, V^{n-1}, V^n$, and V^{n+1}

$$\begin{array}{lcl} V^{n+1} & \overset{\phi^{n+1}}{\rightarrow} & W^{n+1} \\ & (\text{if } \phi^{n+1}_k\phi^n_k < 0, \quad \text{set} \quad W^{n+1}_k = W^{\mathcal{R}^n}_k) & \\ V^n & \overset{\phi^{n+1}}{\rightarrow} & W^n \\ V^{n-1} & \overset{\phi^{n+1}}{\rightarrow} & W^{n-1} \\ & \vdots & \\ V^{n-k+2} & \overset{\phi^{n+1}}{\rightarrow} & W^{n-k+2} \end{array} \tag{48}$$

Note that at each time-station t^{n+1}, Step (5) of the above GFMP–ERS not only constructs the solution at t^{n+1}, W^{n+1}, but also re-evaluates the solutions at the previous k time-stations.

7. Applications and performance assessments

First, a series of one-dimensional two-phase flow problems in a shock tube is considered to illustrate the behavior and performance of the various methods discussed in this paper. More specifically, the shock tube is assumed to have a unit length in the x direction. It contains two different fluids that are initially at rest and separated by a thin membrane. At $t = 0$, the two-phase flow is generated by the bursting of the membrane. This flow is one-dimensional, but all reported calculations are performed on a three-dimensional unstructured mesh with either 201 or 801 grid points along the x direction.

Next, the potential of the GFMP–ERS for the solution of realistic multi-fluid problems is demonstrated with the three-dimensional simulation of the implosion of an air-filled and submerged glass sphere, and the comparison of the obtained numerical results to experimental as well as other numerical simulation data.

Various single- and multi-step time-integrators are considered but in all cases, the governing Euler and level-set equations are semi-discretized by the second-order FV scheme outlined in Section 2.4.

7.1. One-dimensional two-phase flow benchmark problems

7.1.1. Perfect gas – perfect gas computations

The purpose of this first example problem, which was also considered in [15], is to illustrate the extension of the basic GFMP to a multi-step time-integrator. The two fluids are in this case perfect gases and the membrane is positioned at $x = 0.5$. The initial states of the gases at the left and right sides of the membrane and the constants of their EOSs are

$$\rho_L = 1.0 \quad u_L = 0 \quad p_L = 1.0 \quad \gamma_L = 1.4 \quad \text{and} \quad \rho_R = 0.125 \quad u_R = 0 \quad p_R = 0.1 \quad \gamma_R = 1.2 \tag{49}$$

and therefore

$$\frac{\rho_L}{\rho_R} = 8 \tag{50}$$

The spatial discretization is performed with 201 grid points along the length of the tube. Three computations are performed: one using the GFMP with a fourth-order Runge–Kutta (RK4) time-integrator operating at $\text{CFL} = 0.8$, and two using the GFMP with a three-point backward difference implicit (3PBDF) time-integrator operating at $\text{CFL} = 5.0$ and $\text{CFL} = 8.0$, respectively. Fig. 3 which reports the numerical results at $t = 0.2$ and compares them to the analytical solution shows that the GFMP equipped with the RK4 reproduces correctly the variations of the density, velocity, and pressure along the tube. The GFMP equipped with the 3PBDF implicit scheme is also reported to correctly reproduce the variations of these quantities, except for the small bumps it introduces in the pressure and velocity at the shock. These bumps are not due to the GFMP but to the second-order time-accurate 3PBDF operating at $\text{CFL} = 8.0$. At the lower CFL value of 5.0, the bumps become even smaller. In any case, the computed solutions at the material interface are shown to be in very good agreement with the analytical solution.

7.1.2. Perfect gas – perfect gas problem with a reflectionless shock

The shock tube problem considered here was also discussed in [14,17]. In this case, the membrane is positioned at $x = 0.2$ and separates two perfect gases whose initial states and EOS constants are

$$\rho_L = 3.2 \quad u_L = 9.43499279 \quad p_L = 100.0 \quad \gamma_L = 5/3 \quad \text{and} \quad \rho_R = 1.0 \quad u_R = 0 \quad p_R = 1.0 \quad \gamma_R = 1.2. \tag{51}$$

Hence

$$\frac{\rho_L}{\rho_R} = 3.2. \tag{52}$$

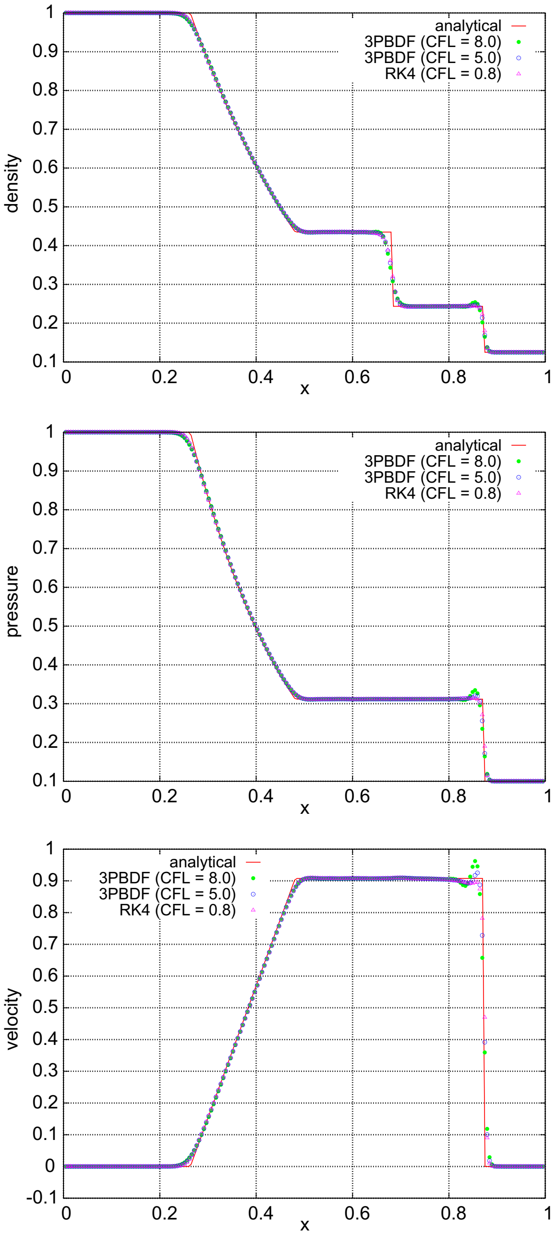

Fig. 3. Perfect gas – perfect gas: variations of the density, pressure, and velocity at $t = 0.2$ along the length of the shock-tube (GFMP, $\Delta x = 1/201$).

This problem is easier than the previous one from the density ratio viewpoint. However, it is more challenging than the previous problem from the following viewpoint. The exact solution of this problem consists of a shock wave and a contact discontinuity only that propagate to the right side of the material interface. On the other hand, most if not all numerical methods applied to the solution of this problem can be expected to generate a non-physical reflection at the material interface and therefore produce a solution containing also a wave that propagates to the left of the material interface. This is because of so-called start-up errors – that is, errors due to the unavoidable inexact representation of the initial conditions. In this sense, this problem allows to evaluate the sensitivity of a computational method to imperfect initial data.

Two numerical computations are performed on the mesh with 201 grid points in the x direction: one using the GFMP and one using the GFMP–ERS. In both cases, the RK4 time-integrator is chosen and the CFL number is set to 0.8. The results obtained at $t = 0.06$ are reported in Fig. 4. Some small amplitude oscillations can be observed in the computed solutions. They are due to the reflection at the material interface. As mentioned in [14], this spurious reflection is difficult to remove. The amplitude of the main oscillation exhibited in the GFMP–ERS solution is shown to be twice as small as that exhibited by the GFMP solution. In any case, the numerical solution delivered here by the GFMP–ERS appears to be more accurate than that reported in [14] and comparable to that reported in [17].

7.1.3. Perfect gas – stiffened gas system with a density ratio of 20 and higher

Here, a series of shock tube problems is considered to illustrate the limits of the GFM and GFMP for multi-fluid problems with a strong interfacial contact discontinuity, and highlight the superior performance of the GFMP–ERS for such problems.

The first shock tube problem discussed herein was also considered in [15]. In this problem, the membrane is positioned at $x = 0.3$. The fluid at the left side of this membrane is a perfect gas. The fluid at the right side of the membrane is water and is modeled as a stiffened gas. The initial states of both fluids and the constants of their EOSs are

$$\begin{aligned} &\rho_L = 50.0 \quad u_L = 0 \quad p_L = 10^5 \quad \gamma_L = 1.4 \\ &\qquad\qquad \text{and} \\ &\rho_R = 1000.0 \quad u_R = 0 \quad p_R = 10^9 \quad \gamma_R = 4.4 \quad \pi_R = 6.0 \times 10^8 \end{aligned} \tag{53}$$

and therefore

$$\frac{\rho_L}{\rho_R} = 20. \tag{54}$$

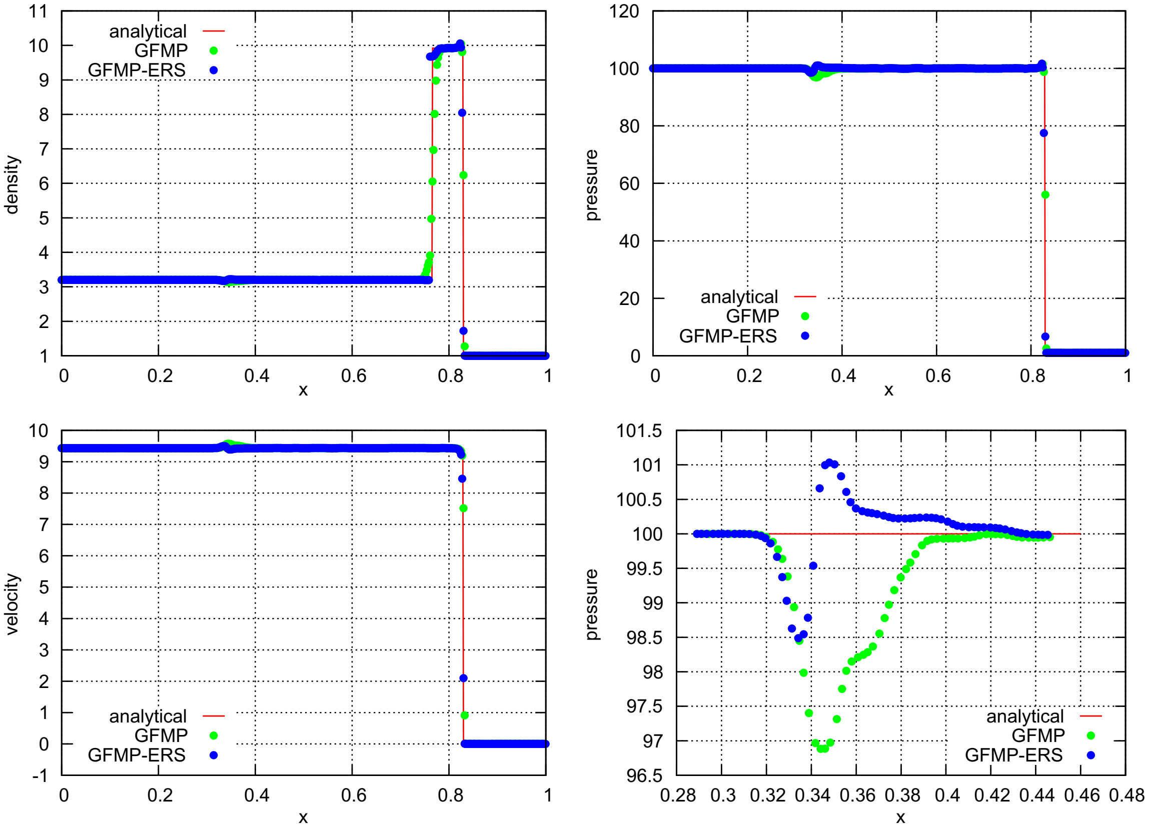

Fig. 4. Reflectionless gaseous shock problem: variations of the density, pressure, and velocity at $t = 0.06$ along the shock tube (GFMP–ERS, $\Delta x = 1/201$) – zoom on the main oscillation is shown for the pressure field at the bottom right part of the figure.

Two meshes are generated: one with 201 grid points along the x direction, and one with 801 grid points along this direction.

On each mesh, two computations are performed: the first one using the GFMP and the second one using the GFMP–ERS. In both cases, the time-discretization is performed using the RK4 time-integrator and the CFL number is set to 0.8. The results at $t = 2.4 \times 10^{-4}$ are reported in Fig. 5 ($\Delta x = 1/201$) and Fig. 6 ($\Delta x = 1/801$). On the mesh with 801 grid points in the x direction, both of the GFMP and GFMP–ERS perform well. However, the GFMP–ERS predicts a sharper density jump close to the material interface. On the coarser mesh with 201 grid points in the x direction, only the GFMP–ERS captures the density plateau between the shock and the contact surface. This underscores the superior performance of the GFMP–ERS for such problems.

Next, variants of the above problem with an increasingly higher density ratio are considered by decreasing the initial value of the density of the perfect gas. All other parameters of the above shock tube problem are kept unchanged. The GFM, GFMP, and GFMP–ERS are applied to the solution of these problems on both generated meshes in the time-interval $[0, 1.2 \times 10^{-4}]$. For this purpose, all three methods are equipped with the RK4 time-integrator. However, the GFM and GFMP are equipped in this case with the Lax–Friedrichs flux scheme characterized by the positivity property [16], whereas the standard Roe flux is used in the GFMP–ERS computations. The outcomes of the performed simulations are characterized in Table 1 below where "succeeds" means that the simulation terminates successfully and produces the correct results, and "fails" means that the computations fail during the simulation – typically, early on and because of encountered negative pressure values.

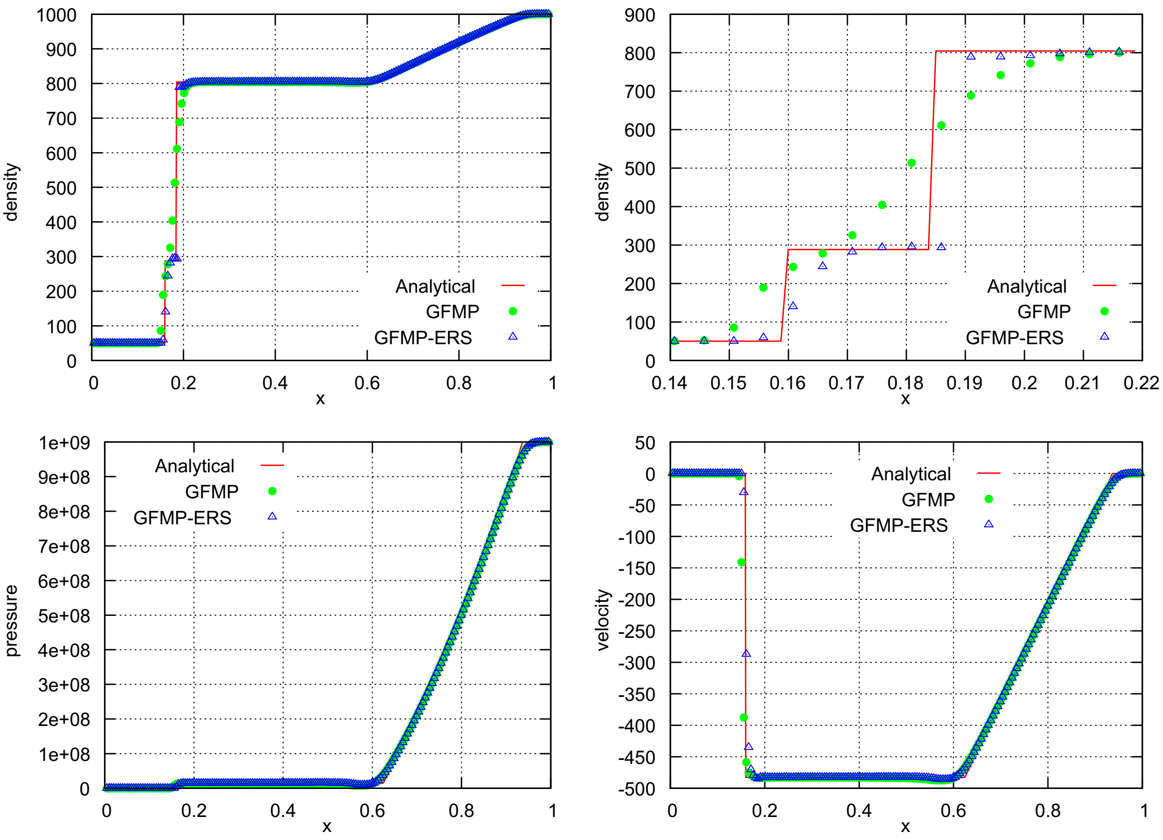

Fig. 5. Perfect gas – stiffened gas: variations of the density, pressure, and velocity at $t = 2.4 \times 10^{-4}$ along the length of the shock-tube ($\Delta x = 1/201$) – zoom on the "plateau" region is shown for the density field at the top right part of the figure.

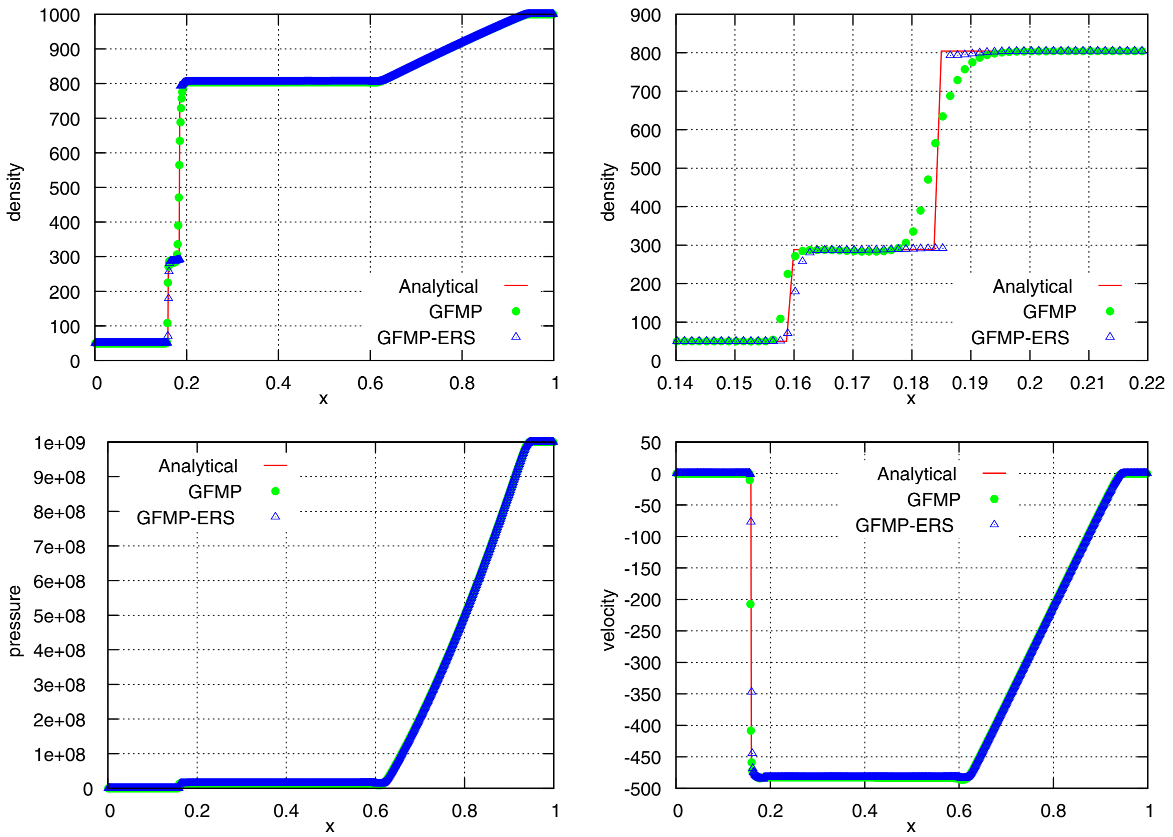

Fig. 6. Perfect gas – stiffened gas: variations of the density, pressure, and velocity at $t = 2.4 \times 10^{-4}$ along the length of the shock-tube ($\Delta x = 1/801$) – zoom on the "plateau" region is shown for the density field at the top right part of the figure.

Table 1
Perfect gas – stiffened gas: limits of the GFM and GFMP and advantage of the GFMP–ERS for problems with a strong interfacial contact discontinuity

Density ratio	GFM Lax–Friedrichs flux CFL = 0.1	GFMP Lax–Friedrichs flux CFL = 0.1	GFMP-ERS Roe flux CFL = 0.8
1000/50 = 20	Succeeds	Succeeds	Succeeds
1000/40 = 25	Fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	Succeeds	Succeeds
1000/10 = 100	Fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	Succeeds	Succeeds
1000/5 = 200	Fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	Fails (negative pressure, $\Delta x = 1/201$ and $\Delta x = 1/801$)	Succeeds

The reader can observe that despite using a flux with the positivity property, the GFM fails to solve all instances of the considered problem with a density ratio higher or equal to 25, even when the CFL number is set as low as 0.1. The GFMP equipped with the same flux scheme also fails as soon as the density ratio exceeds the value of 200, even when the CFL number is reduced to 0.1. On the other hand, the GFMP–ERS equipped with the standard Roe flux successfully solves all considered instances of the problem on both coarse and fine grids. This highlights the limits of the GFM and GFMP for problems with a strong interfacial contact discontinuity, even when equipped with a flux scheme with the positivity property, and the robustness of the GFMP–ERS for such problems. It also supports the explanation that for multi-fluid problems with a large discontinuity of the density at the material interface, using a flux scheme with the positivity property does not seem to be as crucial as using an appropriate discretization scheme that does not cross the material interface.

7.1.4. Gas – water system with a density ratio of 1000

Here, a stiffer model problem with two differents EOSs for modeling gas and water is considered. The membrane is positioned at $x = 0.3$. The initial conditions for both fluid media are more relevant than previously to underwater implosions where the ratio of densities at the material interface is approximately 1000 as they are set to

$$\rho_L = 1.0 \quad u_L = 0 \quad p_L = 10^5 \text{ and } \rho_R = 1000.0 \quad u_R = 0 \quad p_R = 10^7 \tag{55}$$

and therefore

$$\frac{\rho_L}{\rho_R} = 1000. \tag{56}$$

Two computations are performed on the mesh with 201 points in the x direction. In the first one, the water is modeled by Tait's EOS with $k_1 = 2.07 \times 10^9$, $k_2 = 7.15$, $p_0 = p_R$ and $\rho_0 = \rho_R$. In the second one, the water is modeled by the stiffened gas EOS with $\gamma = 7.15$ and $\pi = \frac{k_1}{k_2} = \frac{2.07\times10^9}{7.15}$ In both cases, the air is modeled by the perfect gas EOS with $\gamma = 1.4$.

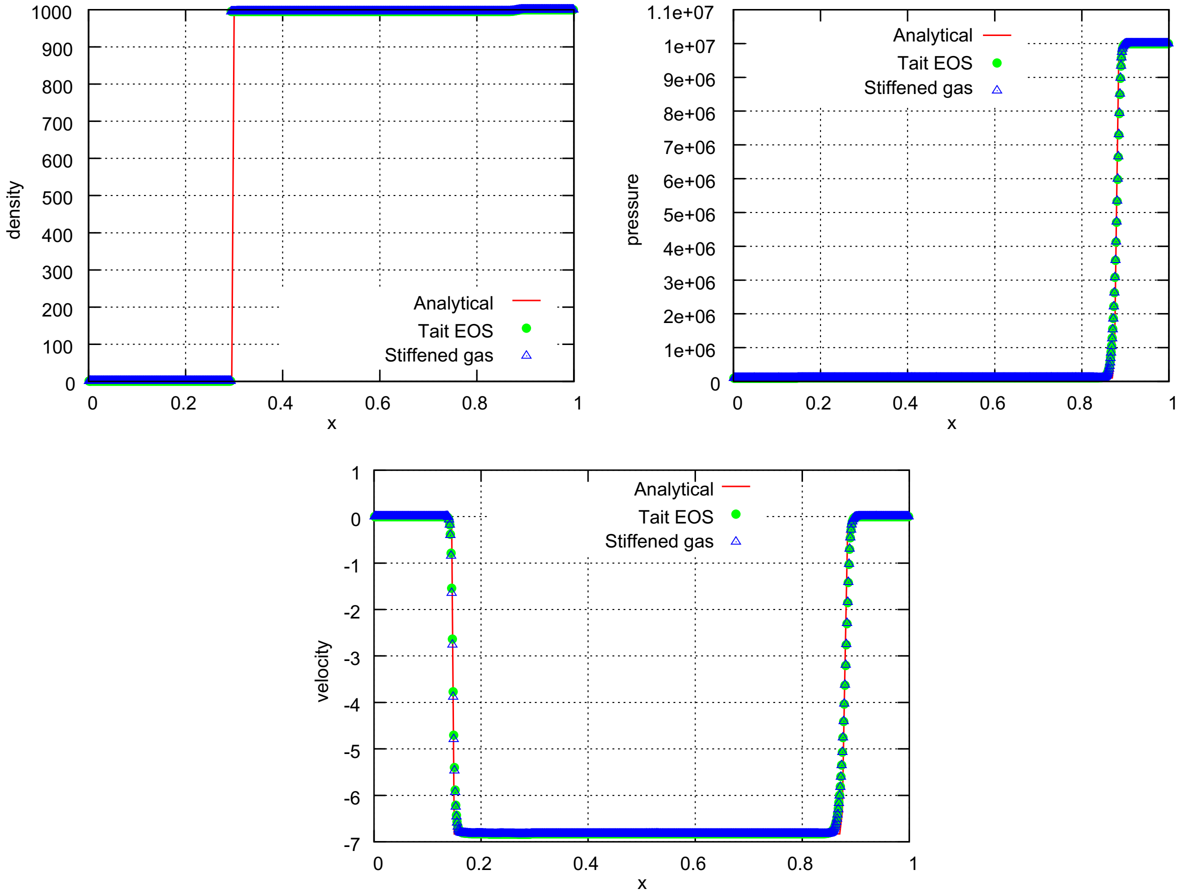

Fig. 7. Air (perfect gas) – water (barotropic fluid/stiffened gas): variations of the density, pressure, and velocity at $t = 4.0 \times 10^{-4}$ along the length of the shock-tube (GFMP–ERS, $\Delta x = 1/201$).

For this problem, the GFM and the GFMP fail early on in the simulation because of the presence of a strong contact discontinuity. On the other hand, the GFMP–ERS equipped with the RK4 time-integrator operating at CFL = 0.8 delivers in both cases excellent results, as shown in Fig. 7 for $t = 4 \times 10^{-4}$.

Note that the analytical solution of the above problem is the same whether the water is modeled as a barotropic fluid or as a stiffened gas. The structure of this solution consists of a shock wave travelling in the air, a contact discontinuity, and a rarefaction wave propagating in the water. This is consistent with both the physics of the problem and the chosen models. Indeed, as shown in Remark 3 (see Section 2.2.2), a stiffened gas behaves during an isentropic transformation like a barotropic fluid modeled by Tait's EOS.

7.2. Underwater implosion

The GFMP–ERS was implemented in the AERO-F flow code [25,26]. Here, it is applied to the three-dimensional simulation of the implosion of an air-filled and submerged glass sphere. The parameters of this simulation correspond to the experiments and test data recently reported in [27].

In the experimental setup described in [27], an air-filled glass sphere was submerged in a pressure vessel filled with water. The implosion of the glass sphere was initiated either by a critical hydrostatic pressure, or by the actuation of a piston at the bottom of the sphere. The test stand consisted of an aluminium base plate and a 7.62 cm diameter pipe standing vertically. A glass sphere with an outer radius of 3.81 cm was placed on top of the pipe. Four implosion experiments were performed with an initial hydrostatic pressure of 6.996 MPa, and an initial pressure inside the glass sphere of 101.3 kPa. Three dynamic pressure sensors were installed at 10.16 cm from the center of the sphere, at the same height, and in three directions 120° apart. For these four experiments, the recorded pressure time-histories (see Fig. 9) reveal pressure drops of 1.6 MPa and pressure peaks ranging between 25.8 and 27.2 MPa (variations of the order of 5%). A secondary peak can also be observed in Fig. 9; however, its amplitude and position in time have a greater variability than the pressure drop and primary peak.

Using the two-dimensional axisymmetric computational domain shown in Fig. 8, various numerical simulations were also performed by the author of [27] using the DYSMAS code [29]. In these simulations, both

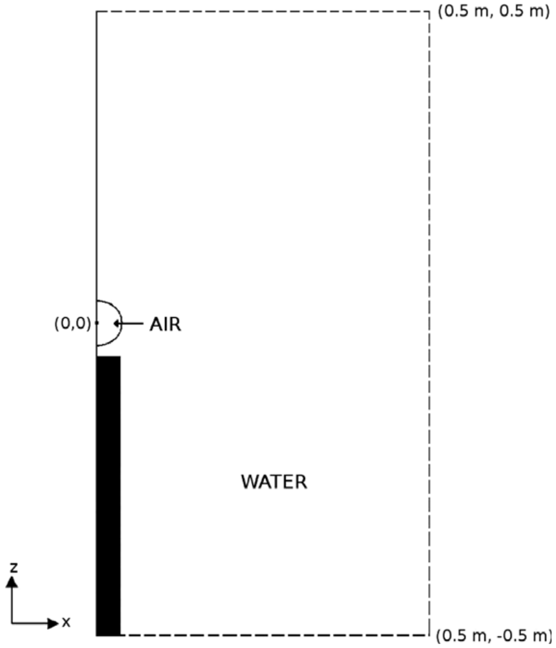

Fig. 8. Schematic of the implosion experiment reported in [27] and corresponding computational domain (dashed lines represent the non-reflecting boundaries of the two-dimensional computational domain adopted in [27]).

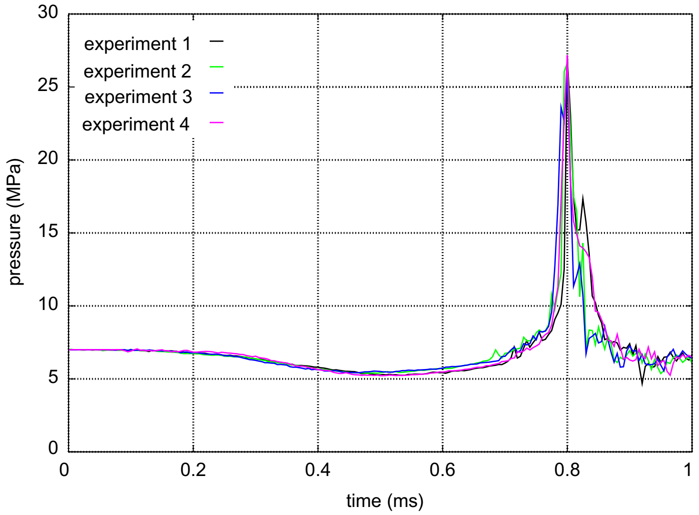

Fig. 9. Pressure time-histories recorded at one of the sensors during the four experiments described in [27].

fluid media were assumed to be inviscid. Water was modeled by Tillotson's EOS and air by the perfect gas EOS. The initial conditions were set to

$$\rho_w = 1000.0\ \mathrm{kg/m^3},\quad u_w = 0\ \mathrm{m/s},\quad p_w = 6.996\ \mathrm{MPa},$$
$$\rho_a = 1.3\ \mathrm{kg/m^3},\quad u_a = 0\ \mathrm{m/s},\quad p_a = 101.3\ \mathrm{kPa},$$

where the subscripts w and a designate water and air, respectively. An element deletion technique for prescribing the removal of the glass material was also applied at various speeds. However, only the case where the glass was assumed to have disappeared at $t = 0$ (infinite element deletion speed) is reported here (for the sake of comparison with this paper's results where no such technique was used). In general, DYSMAS predicted a pressure drop of almost 3.0 MPa and a primary pressure peak of 38.4 MPa at the sensor locations. It also predicted a secondary pressure peak and a pressure dip after both pressure peaks of approximately 4.5 MPa (see Fig. 10).

As mentioned at the beginning of this section, the AERO-F code equipped with the proposed GFMP–ERS is also applied here to the simulation of the implosion experiment described above. Because the main purpose

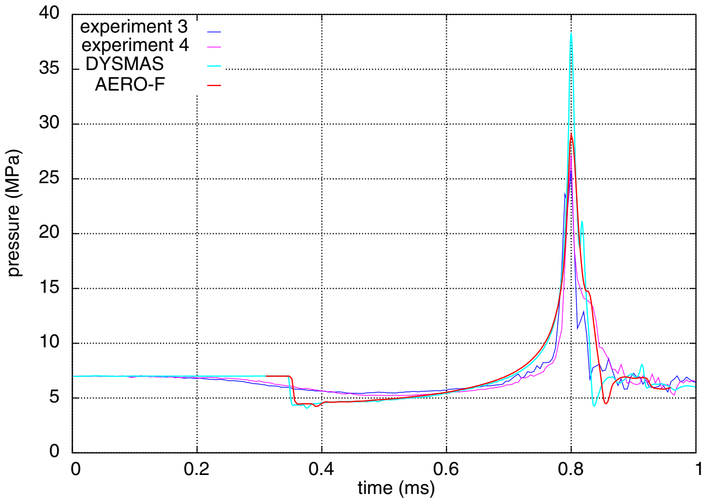

Fig. 10. Comparison of the pressure time-histories predicted by AERO-F and DYSMAS as well as their corresponding test data.

of this simulation is the verification of a three-dimensional code, a three-dimensional computational domain covering a 20-degree slice of the cylindrical pressure vessel is chosen for this purpose. All other dimensions of this computational domain and corresponding non-reflecting boundaries are chosen to be the same as those used in [27]. This domain is discretized by a grid with 794,254 nodes, 4,484,412 tetrahedra, and a mesh density similar to that used for the numerical simulations reported in [27]. Symmetry boundary conditions are applied on the lateral boundaries of this computational domain. The water is modeled by the stiffened gas EOS with $\gamma = 7.15$ and $P_\infty = 2.89 \times 10^8$ Pa. The air is modeled by the perfect gas EOS with $\gamma = 1.4$. Both media are also assumed to be inviscid. The following initial conditions, which are consistent with those of the experiments reported in [27], are adopted. At $t = 0$, the air is assumed to occupy the same volume as the sphere of glass before it breaks, and to be still at a uniform pressure of 101.3 kPa and a uniform density of 1.3 kg/m^3. The initial hydrostatic pressure of the water surrounding this air bubble is assumed to be equal to 6.996 MPa at the depth of the center of the air bubble; its initial density is set to 1000.0 kg/m^3 all over the computational domain. The AERO-F simulation is performed using a second-order space-accurate GFMP–ERS and the second-order Runge–Kutta time-integrator. The CFL number is fixed to 0.5 and the computation is performed until reaching the physical time of 0.6495 ms.

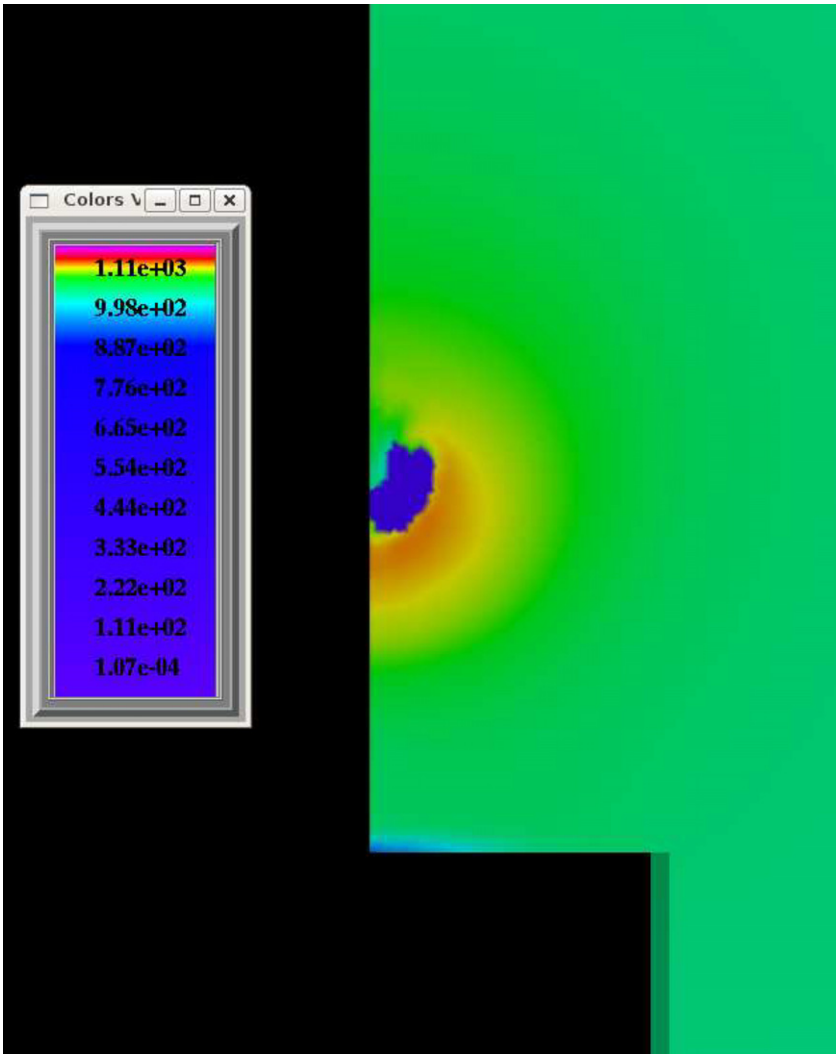

Fig. 11. Density contourplots (in kg/m^3) predicted by AERO-F equipped with the GFMP–ERS in the vicinity of the air bubble during its collapse.

Fig. 10 reports the pressure time-history predicted by AERO-F (equipped with the GFMP–ERS) and compares it to: (a) that predicted by the DYSMAS code, and (b) those recorded during experiment 3 and experiment 4. The focus on the results of these two experiments is only because they "envelop" the results of the other two experiments, and reporting only these keeps Fig. 10 readable. (Note that as done in [27], time was shifted in this figure so that the pressure peak is reached at $t = 0.8$ ms). The GFMP–ERS is shown to reproduce the recorded pressure signal fairly accurately. A first pressure drop of almost 3 MPa is predicted at $t = 0.354$ ms, slightly later than that predicted by DYSMAS ($t = 0.350$ ms). The primary pressure peak predicted by AERO-F is 29.0 MPa: it is closer to the measured pressure peak value (25.8–27.2 MPa) than that predicted by DYSMAS (38.4 MPa). On the other hand, the secondary peak predicted by AERO-F is flattened. The lowest pressure level predicted by AERO-F (4.5 MPa) is comparable to that predicted by DYSMAS. After this lowest pressure level is reached, both codes correctly predict similar rises to the initial pressure level.

Finally, it is noted that the implosion experiment described herein cannot be simulated by a one-dimensional spherical model. Indeed, Fig. 11 which displays the contourplots of the density field computed by AERO-F at $t = 0.87$ ms reveals that after some point during its collapse, the bubble is no longer spherical (until after its rebound).

8. Conclusions

The ghost fluid method for the poor (GFMP) was developed for the solution of two-phase flow problems using the stiffened gas equation of state and one-step time-discretization algorithms. It is a nearly conservative and computationally efficient method. However, it cannot handle problems with strong contact discontinuities, particularly at practical mesh resolutions. As such, it is not applicable to the solution of two-phase air/water applications such as underwater implosions. In this paper, the GFMP was generalized to arbitrary equations of state and multi-fluid problems with multiple equations of state. It was also extended to higher-order multi-step time-integrators. Most importantly, the GFMP was also equipped with an exact, local, one-dimensional, two-phase Riemann solver for computing the fluxes at the material interface without crossing it, in order to make this method robust with respect to a large discontinuity of the density and a strong pressure jump at the material interface. Consequently, the resulting multi-fluid method was labeled the GFMP–ERS (for GFMP with Exact Riemann Solver). Like the original GFMP, the GFMP–ERS is computationally efficient, contact preserving, and nearly conservative. Its application in this paper to the solution of various shock tube problems with stiffened gas and barotropic equations of state has revealed a superior performance in the presence of large density and pressure jumps. Also, its successful application to the simulation of the implosion of an air-filled and submerged glass sphere for which experimental data is available has highlighted its potential for the analysis of underwater implosion problems.

Appendix A. One-dimensional two-phase Riemann problems

At each time-step, the one-dimensional two-phase Riemann problem (35) is constructed along each edge $i - j$ that crosses the material interface which is designated here by the subscript I. This problem can be reduced to an explicit expression of the normal velocity at the material interface, u_I, as a function of the pressure at this material interface, p_I, and a non-linear equation in p_I. For example, when both media on the left and right sides of the material interface are modeled as stiffened gases, the local Riemann problem can be written as

$$\begin{aligned} u_I &= \frac{1}{2}(u_L + u_R) + \frac{1}{2}(\mathcal{R}_R(p_I; p_R, \rho_R) - \mathcal{R}_L(p_I; p_L, \rho_L)) \\ \mathcal{R}(p_I; u_L, p_L, \rho_L, u_R, p_R, \rho_R) &= \mathcal{R}_L(p_I; p_L, \rho_L) + \mathcal{R}_R(p_I; p_R, \rho_R) + u_R - u_L = 0 \end{aligned} \tag{A.1}$$

where the subscripts L and R designate the left and right sides of the material interface, respectively, $\mathcal{R}_L$ and $\mathcal{R}_R$ are two vector functions that depend on the structure of the wave solution at the left and right sides of the contact discontinuity (see Fig. 2), and a ";" is used to separate the unknown variables from known quantities.

When both media on the left and right sides of the material interface are modeled by Tait's EOS, the Riemann problem can be written as

$$\begin{aligned} u_I &= \frac{1}{2}(u_L + u_R) + \frac{1}{2}\left(\mathcal{R}_R^*(p_I;\rho_R) - \mathcal{R}_L^*(p_I;\rho_L)\right), \\ \mathcal{R}^*(p_I;u_L,\rho_L,u_R,\rho_R) &= \mathcal{R}_L^*(p_I;\rho_L) + \mathcal{R}_R^*(p_I;\rho_R) + u_R - u_L = 0, \end{aligned} \tag{A.2}$$

where $\mathcal{R}_L^*$ and $\mathcal{R}_R^*$ are two vector functions that depend on the structure of the wave solution at the left and right sides of the contact discontinuity (see Fig. 2). Analytical expressions for $\mathcal{R}_L$, $\mathcal{R}_R$, $\mathcal{R}_L^*$, and $\mathcal{R}_R^*$ can be obtained from the Rankine–Hugoniot jump conditions for shocks and the isentropic relations for rarefactions [23]. For the sake of completeness, these are given in Sections A.1,A.2,A.3,A.4 of this appendix. Once Eq. (A.1) is solved for p_I – for example, using Newton's method – the computation of other interface quantities such as u_I, ρ_{I_L} and ρ_{I_R} becomes straightforward.

A.1. Shock wave relations for a stiffened gas

The shock wave relations for a stiffened gas can be written in terms of the unknown value of the pressure at the material interface, p_I, as follows

$$\mathcal{R}_K(p_I;p_K,\rho_K) = \left(\sqrt{\frac{a_K}{\bar{p}_I + b_K}}\right)(p_I - p_K), \tag{A.3}$$

where the subscript K designates either the medium at the left (L) or that at the right (R) of the material interface,

$$a_K = \frac{2}{(\gamma_K + 1)\rho_K} \quad b_K = \left(\frac{\gamma_K - 1}{\gamma_K + 1}\right)\bar{p}_K \quad \bar{p}_K = p_K + \pi_K \tag{A.4}$$

and γ_K and π_K have been defined in Section 2.2.1 and correspond to the EOS on the K side of the interface as indicated by the subscript.

The pressure derivative of $\mathcal{R}_K$ at the material interface is given by

$$\mathcal{R}'_K(p_I;p_K,\rho_K) = \frac{\mathrm{d}\mathcal{R}_K}{\mathrm{d}p_I} = -\frac{a_K}{2(\bar{p}_I + b_K)^2}. \tag{A.5}$$

A.2. Expansion wave relations for a stiffened gas

The expansion wave relations for a stiffened gas are given by

$$\mathcal{R}_K(p_I;p_K,\rho_K) = \left(\frac{2c_K}{\gamma_K - 1}\right)\left(\left(\frac{\bar{p}_I}{\bar{p}_K}\right)^{\frac{\gamma_K-1}{2\gamma_K}} - 1\right) \tag{A.6}$$

and

$$\mathcal{R}'_K(p_I;p_K,\rho_K) = \left(\frac{c_K}{\gamma_K \bar{p}_K^{\frac{\gamma_K-1}{2\gamma_K}}}\right)\left(\frac{\bar{p}_I}{\bar{p}_K}\right)^{-\left(\frac{\gamma_K+1}{2\gamma_K}\right)}, \tag{A.7}$$

where c_K denotes as before the speed of sound (see Eq. (10)) on the side K of the interface.

A.3. Shock wave relations for Tait's EOS

For Tait's EOS, the shock wave relations can be written as

$$\mathcal{R}_K^*(p_I;\rho_K) = \sqrt{\frac{(p_I - p_K)(\rho_I - \rho_K)}{\rho_K \rho_I}} \tag{A.8}$$

and

$$\mathcal{R}_K^{*\prime}(p_I;\rho_K) = \frac{d\mathcal{R}_K^*}{dp_I} = \left(\frac{1}{2\mathcal{R}_K^*}\right)\left(\frac{\rho_I(\rho_I-\rho_K)+(p_I-p_K)\rho_K\rho_I'}{\rho_K\rho_I^2}\right), \tag{A.9}$$

where

$$\rho_I = \left(\frac{p_I-\eta_K}{\alpha_K}\right)^{\beta_K^{-1}} \qquad \rho_I' = \frac{\rho_I^{1-\beta_K}}{\alpha_K\beta_K} \qquad p_K = \eta_K + \alpha_K\rho_K^{\beta_K} \tag{A.10}$$

α_K and β_K and η_K have been defined in Section 2.2.2 and correspond to the EOS on the side K of the interface.

A.4. Expansion wave relations for Tait's EOS

On the other hand, the expansion wave relations governing a medium modeled by Tait's EOS are given by

$$\mathcal{R}_K^*(p_I;\rho_K) = \left(\frac{2c_K}{\beta_K-1}\right)\left(\left(\frac{\bar{p}_I}{\bar{p}_K}\right)^{\frac{\beta_K-1}{2\beta_K}}-1\right) \tag{A.11}$$

and

$$\mathcal{R}_K^{*\prime}(p_I;\rho_K) = \frac{c_K}{\beta_K\bar{p}_K}\left(\frac{\bar{p}_I}{\bar{p}_K}\right)^{-\frac{\beta_K+1}{2\beta_K}}. \tag{A.12}$$

A.5. Local solution by Newton's method

The application of Newton's method to the solution of the local non-linear Eq. (A.1) for the interface pressure, p_I, by Newton's method generates the following sequence of iterate values of p_I

$$p_I^{(m+1)} = p_I^{(m)} - \frac{\mathcal{R}(p_I^{(m)})}{\mathcal{R}'(p_I^{(m)})}, \tag{A.13}$$

where m designates the Newton iteration. In this work, convergence of the sequence (A.13) is declared when

$$\frac{2|p_I^{(m+1)}-p_I^{(m)}|}{p_I^{(m+1)}+p_I^{(m)}} < \epsilon, \tag{A.14}$$

where ϵ is a specified tolerance.

References

[1] S.K. Godunov, A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics, Mat. Sb. 47 (1959) 357.

[2] B. VanLeer, Towards the ultimate conservative difference scheme, V. A second order sequel to Godunov's method, J. Comput. Phys. 32 (1979) 1011.

[3] B. VanLeer, Flux-vector splitting for the euler equations, Technical report, Lecture Notes in Physics, Springer, Berlin, 1982.

[4] R. Abgrall, Generalization of the Roe scheme for the computation of mixture of perfect gases, La Recherche Aerospatiale 6 (1988) 31.

[5] S. Karni, Multicomponent flow calculations by a consistent primitive algorithm, J. Comput. Phys. 112 (1994) 31.

[6] C.W. Hirt, B.D. Nichols, Volume of fluid method (vof) for dynamics of free boundaries, J. Comput. Phys. 31 (1981) 201.

[7] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A non-oscillatory Eulerian approach to interfaces in multimaterial flows (the ghost fluid method), J. Comput. Phys. 152 (1999) 457.

[8] R. Abgrall, How to prevent pressure oscillation in multicomponent flow calculation: a quasi-conservative approach, J. Comput. Phys. 125 (1996) 150.

[9] S. Karni, Hybrid multifluid algorithms, SIAM J. Sci. Comput. 17 (1996) 1019.

[10] R. Saurel, R. Abgrall, A simple method for compressible multifluid flows, SIAM J. Sci. Comput. 21 (3) (1999) 1115.

[11] A. Marquina, R. Fedkiw, B. Merriman, An isobaric fix for the overheating problem in multimaterial compressible flows, J. Comput. Phys. 148 (1999) 545.

[12] Y. Liu, J. Glimm, X.L. Li, N. Zhao, Conservative front tracking and level set algorithms, PNAS 98 (25) (2001) 14198.

[13] F. Gibou, D. Nguyen, R. Fedkiw, A fully conservative ghost fluid method & stiff detonation waves, Technical report, in: 12th International Detonation Symposium, San Diego, CA, 2002.

[14] B.C. Khoo, T.G. Liu, C.W. Wang, The ghost fluid method for compressible gas–water simulations, J. Comput. Phys. 204 (2005) 193.

[15] R. Abgrall, S. Karni, Compressible multifluid flows, J. Comput. Phys. 169 (2) (2001) 594.

[16] R. Abgrall, Personal communication.

[17] C.W. Wang, T.G. Liu, B.C. Khoo, A real ghost fluid method for the simulation of mutimedium compressible flow, SIAM J. Sci. Comput. 28 (2006) 278.

[18] R.H. Cole, UnderWater Explosions, Dover, New York, 1965.

[19] J.B. Keller, I.I. Kolodner, Damping of underwater explosion bubbles, J. Appl. Phys. 27 (10) (1956) 1152.

[20] T.L. Geers, R.S. Lagumbay, O.V. Vasilyev, Numerical modeling and simulation of an underwater explosion bubble. Technical report, American Physical Society, Chicago, November 2005.

[21] P. Smerka, M. Sussman, S. Osher, A level-set approach to computing solutions to two phase incompressible flows, J. Comput. Phys. 114 (1994) 146.

[22] P.L. Roe, Approximate Riemann solver, parameters vectors and difference schemes, J. Comput. Phys. 43 (1981) 357.

[23] D.M. Causon, M.J. Ivings, E.F. Toro, On Riemann solvers for compressible liquids, Int. J. Numer. Meth. Fluids 28 (1998) 395.

[24] A. Haselbacher, J. Blazek, Accurate and efficient discretization of Navier–Stokes equations on mixed grids, AIAA J. 38 (2000) 2094.

[25] P. Geuzaine, G. Brown, C. Harris, C. Farhat, Aeroelastic dynamic analysis of a full F-16 configuration for various flight conditions, AIAA J. 41 (2003) 363.

[26] C. Farhat, P. Geuzaine, G. Brown, Application of a three-field nonlinear fluid-structure formulation to the prediction of the aeroelastic parameters of an F-16 fighter, Comput. Fluids 32 (2003) 3.

[27] S.E. Turner, Underwater implosion of glass spheres, J. Acoust. Soc. Am. 121 (2007) 844.

[28] F. Mut, G.C. Buscaglia, E.A. Dari, New mass-conserving algorithm for level set redistancing on unstructured meshes, J. Appl. Mech. 73 (2006) 1011.

[29] A.B. Wardlaw Jr., R. McKeown, H. Chen, Implementation and application of the P-α equation of state in the DYSMAS code. NSWCDD/TR-95/107, 1996.