A time splitting projection scheme for compressible two-phase flows. Application to the interaction of bubbles with ultrasound waves

Grégory Huber [a,b], Sébastien Tanguy [a,b,*], Jean-Christophe Béra [c,d], Bruno Gilles [c,d]

[a] *Université de Toulouse, France*
[b] *Institut de Mécanique des Fluides de Toulouse, France*
[c] *Inserm U1032, Lab. of Therapeutic Applications of Ultrasound, Lyon, France*
[d] *Université Claude Bernard Lyon 1, Lyon, France*

Keywords:
Two-phase flows
Low Mach compressible solver
Level set
Ghost fluid
Projection method

A B S T R A C T

This paper is focused on the numerical simulation of the interaction of an ultrasound wave with a bubble. Our interest is to develop a fully compressible solver in the two phases and to account for surface tension effects.

As the volume oscillation of the bubble occurs in a low Mach number regime, a specific care must be paid to the effectiveness of the numerical method which is chosen to solve the compressible Euler equations. Three different numerical solvers, an explicit HLLC (Harten–Lax–van Leer-Contact) solver [48], a preconditioning explicit HLLC solver [14] and the compressible projection method [21,53,55], are described and assessed with a one dimensional spherical benchmark. From this preliminary test, we can conclude that the compressible projection method outclasses the other two, whether the spatial accuracy or the time step stability are considered.

Multidimensional numerical simulations are next performed. As a basic implementation of the surface tension leads to strong spurious currents and numerical instabilities, a specific velocity/pressure time splitting is proposed to overcome this issue. Numerical evidences of the efficiency of this new numerical scheme are provided, since both the accuracy and the stability of the overall algorithm are enhanced if this new time splitting is used. Finally, the numerical simulation of the interaction of a moving and deformable bubble with a plane wave is presented in order to bring out the ability of the new method in a more complex situation.

1. Introduction

The interaction of a bubble with ultrasound waves can lead to volume oscillations and to deformations of the bubble, generating microstreaming in its vicinity. If the amplitude of the ultrasound waves is sufficiently large, the bubble collapses and it can break up into daughter bubbles, generating a shock wave during the phase of separation. These phenomena can be useful in many industrial and medical applications. In waste water purification, chemical, pharmaceutical, mechanical

* Corresponding author.
E-mail addresses: Gregory.Huber@imft.fr (G. Huber), Sebastien.Tanguy@imft.fr (S. Tanguy).

and food industry, as well as for ordinary instrument cleaning, sonicators and sonochemical reactors are based on the action of bubbles submitted to ultrasound. In medicine, current clinical applications of high-intensity focused ultrasound, namely extracorporeal kidney stone destruction and prostate tumor ablation, also involve ultrasound bubbles [32]. Moreover, promising investigations concern new therapy modalities specifically based on these phenomena, aiming notably to targeted drug delivery [39], extracorporeal sonothrombolysis [41] and cell sonoporation [33]. These ultrasound techniques are subject to active research, concerning in particular the interaction between a bubble and a solid wall or a living cell [28,52,58]. Numerical and theoretical investigations are required to help understanding the mechanisms of bubble action involved in these applications.

The development of numerical methods for compressible solvers in the framework of interface capturing (Level-Set method) or interface-tracking methods (Front-Tracking method) for two-phase flows is still an active topic of research. Many existing works [5,8,16,25,35,47] propose numerical strategies to compute the interaction of bubbles or drops with shock waves. In these studies, full explicit shock-capturing solvers are used for the simulations of compressible flows. It is well known that these solvers are perfectly suited to compute the formation and the propagation of shock waves, but works poorly in the low Mach number regime [14,49].

As a result, such solvers are not efficient to compute the collapse of a bubble due to the interaction with ultrasound waves. Indeed, the collapse of the bubble can be divided in different steps involving very heterogeneous values of the local Mach number. In the first one, the volume of the bubble oscillates due to the interaction with the ultrasound waves. This initial step occurs essentially in a low Mach number regime (in the liquid phase). In the second one, if the ultrasound wave amplitude is sufficiently strong, a bubble break-up occurs and leads to the formation of a shock wave. Therefore, the computation of this second step requires the use of a full explicit shock capturing solver to compute accurately the shock wave propagation whereas this solver is not efficient to compute the initial step of the physical process. In the Ref. [42], the authors have performed numerical simulations with a HLLC solver of the expansion of a bubble interacting with ultrasound waves in a one-dimensional spherical coordinate system. In that paper, they present computations with a number of grid points up to 10 000 in order to achieve a sufficient spatial convergence of the computation with only one bubble. This drastic constraint on the spatial cell size limits tremendously the ability of explicit compressible solvers, to deal with more complex problems involving multidimensional features, as it occurs in many practical situations. For example, during the interaction of a bubble with an ultrasound plane wave, the bubble can displace and undergo non-spherical deformations. A multidimensional modeling is relevant in many others situations of interest, as the bubble collapse close to a wall in order to compute the mechanical stresses acting on a wall, or if an increasing complexity approach is considered, to model the collective effects during the collapse of several bubbles interacting with ultrasound waves.

Several other studies [38,57] have been dedicated to the initial step of the bubble oscillation before the collapse. As the Mach number remains low in this step, these studies assume that the liquid remains incompressible and that the bubble pressure is spatially constant. The temporal value of the pressure in the bubble can then be imposed as a Dirichlet boundary condition to solve the pressure field in the incompressible liquid. Whereas these approaches provide very satisfactory results to compute the volume bubble oscillation since an incompressible solver works perfectly well in the Low Mach number regime, they are limited in their representation of the physical reality. Indeed, they are unable to capture the shock formation during the bubble collapse and its propagation in the liquid phase. Moreover, it is not always relevant to consider that the pressure remains uniform inside the bubble if it is strongly deformed, since the variations of curvature along the interface will involve variations on the pressure jump condition. Thus, novel numerical tools, which allow computing accurately the various steps of the bubble collapse, are still required to improve our overall understanding of this complex phenomenon. In particular, a few studies [4,13,37,56] have been dedicated to the development of numerical methods to perform the direct numerical simulations of the interaction of an interface, separating a compressible liquid and a compressible gas, with an acoustic or an ultrasound wave in the low Mach number regime.

In the framework of one-phase flow, different numerical strategies [2,14,18,21,49,50,54,55] exist to overcome the issue of computing compressible low Mach number flow. A usual strategy consists in enhancing a HLLC (Harten–Lax–van Leer-Contact) solver with a preconditioning method [14,49] to improve the spatial accuracy in the low Mach number regime. However, when an explicit temporal discretization is used, this simple method leads to a time step even more stringent than the classical HLLC solver (already too small for the low Mach number regime). An implicit temporal discretization can be used to increase the time step, but its implementation is much more complex [27]. Another kind of methods, based on the incompressible projection method, can be considered. This compressible projection method consists in a semi-implicit temporal scheme to solve the compressible Euler equations [21,53–55]. It allows splitting the computation of convective terms in an explicit temporal scheme and the acoustic terms in an implicit temporal scheme. In comparison to HLLC solver, it improves both the spatial accuracy in the low Mach number regime and the temporal stability of the numerical scheme, since the sound velocity can be removed from the convective time step constraint. It has ever been used in the framework of two-phase flows in [17] but only for the liquid phase since the authors made the assumption of a uniform gas pressure. Moreover, the numerical method, which is described in that paper, do not allow to account for surface tension effect.

Thus, we propose in this paper, a new time splitting projection scheme for compressible two-phase flows, which takes into account surface tension effects and allows computing the pressure, the density and the velocity field in the two phases. The interface motion is described with a Level Set method [36,43]. The jump conditions are imposed following the framework of the Ghost Fluid Method to compute an accurate and sharp description of the interface. Our approach mixes features of numerical methods dedicated to incompressible [19,30] and supersonic compressible [8] two-phase flows. The main nov-

elty of this paper is a new time splitting compressible projection method to compute accurately the surface tension effects in the compressible low Mach number regime. In particular, this new method prevents from numerical instabilities occurring when the jump condition on pressure, due to surface tension, is roughly incorporated in the pressure computation.

A numerical test has been designed in a one dimensional spherical configuration and can be compared to the solution of the Rayleigh–Plesset equation. Comparisons between an explicit HLLC solver, a preconditioned explicit HLLC and a compressible projection method are next presented. As the compressible projection method outclasses the other two, the extension of the compressible projection method to multidimensional flows accounting for surface tension effects is proposed. The requirement to develop a new time splitting to include accurate and stable computations of surface tension is highlighted, and the convergence of the new method is shown on a multidimensional test-case. Finally, a simulation of a moving droplet, which undergoes non-spherical deformations due to the interaction with an ultrasound plane wave, is presented.

2. Governing equations and numerical methods for a one-phase flow

2.1. Governing equations

2.1.1. Euler equations system

Inviscid compressible flows can be computed by solving the following system of partial differential equations (1) which express respectively the conservation of the mass, momentum and energy. This system, known as Euler equations, can be expressed with a conservative formulation:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \vec{u}) &= 0 \\ \frac{\partial \rho \vec{u}}{\partial t} + \nabla \cdot (\rho \vec{u} \otimes \vec{u}) + \nabla p &= 0 \\ \frac{\partial \rho E}{\partial t} + \nabla \cdot ((\rho E + p)\vec{u}) &= 0 \end{aligned} \tag{1}$$

Or with a primitive formulation:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} &= 0 \\ \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{\nabla p}{\rho} &= 0 \\ \frac{\partial p}{\partial t} + \vec{u} \cdot \nabla p + \rho c^2 \nabla \cdot \vec{u} &= 0 \end{aligned} \tag{2}$$

where ρ is the density, $\vec{u}$ the velocity, p the pressure, E the total energy ($E = e + \frac{1}{2}\vec{u} \cdot \vec{u}$, where e is the internal energy) and c the sound speed.

As inviscid flows are considered, all the dissipative effects are neglected, thus the entropy is also a conservative variable. It can be expressed with the following equation:

$$\frac{Ds}{Dt} = 0 \tag{3}$$

In one-dimensional Cartesian coordinate, System (2) reads

$$\frac{\partial W}{\partial t} + A(W)\frac{\partial W}{\partial x} = 0 \tag{4}$$

with $W = (\rho \ u \ p)^T$, and

$$A(W) = \begin{pmatrix} u & \rho & 0 \\ 0 & u & \frac{1}{\rho} \\ 0 & \rho c^2 & u \end{pmatrix} \tag{5}$$

Three eigenvalues of A matrix are $\lambda_1 = u - c$, $\lambda_2 = u$ and $\lambda_3 = u + c$, and its three left eigenvectors are

$$l_1 = \begin{pmatrix} 0 \\ -\rho c \\ 1 \end{pmatrix}, l_2 = \begin{pmatrix} -c^2 \\ 0 \\ 1 \end{pmatrix}, l_3 = \begin{pmatrix} 0 \\ \rho c \\ 1 \end{pmatrix}.$$

The compressible Euler equations System (2) is hyperbolic which means that the solutions of System (2) have wave properties: a disturbance propagates along the characteristics of the system. This mathematical property is the basis of the acoustical waves propagation in fluid mechanics.

If an incompressible flow is considered, the Euler equations simply writes:

$$\begin{aligned} \nabla \cdot \vec{u} &= 0 \\ \frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{\nabla p}{\rho} &= 0 \end{aligned} \tag{6}$$

2.1.2. System closure: equation of state

The system of Euler equations is not complete, and an equation of state must be added to close the system. An expression of the sound speed in the fluid is required. Let us remind the definition of the sound speed:

$$c^2 = \frac{\partial p}{\partial \rho}\Big)_{s=cste} \tag{7}$$

If a perfect gas is considered, $p = \rho r T$ ($r = R/M$ where $R = 287{,}058\ \mathrm{J\,kg^{-1}\,K^{-1}}$), the sound speed is:

$$c^2 = \frac{\gamma p}{\rho}. \tag{8}$$

If a liquid is considered, the Tait equation of state [8] can be used, $p = B((\frac{\rho}{\rho_0})^{\gamma} - 1) + p_0$. It leads to the following relation for the sound speed in a liquid phase

$$c^2 = \frac{\gamma B}{\rho_0}(\frac{\rho}{\rho_0})^{\gamma - 1}. \tag{9}$$

with the following value for water: $p_0 = 10^5$ Pa and $B = 3{,}31 \times 10^8$ Pa.

2.2. Numerical methods

We present in this section some usual numerical methods used to solve the compressible Euler equations.

2.2.1. Riemann type solver

This solver is commonly used to compute compressible flows. For the sake of simplicity, we detail the method in a one dimensional Cartesian coordinate system, but multidimensional extensions can be easily derived. The Euler system in conservative form (1) can be expressed as:

$$\frac{\partial U}{\partial t} + \frac{\partial F}{\partial x} = 0 \tag{10}$$

with $U = (\rho\ \ \rho u\ \ \rho E)^T$, $F = (\rho u\ \ \rho u^2 + p\ \ (\rho E + p)u)^T$.

The Godunov scheme reads

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2}^n - F_{i-1/2}^n) \tag{11}$$

Fluxes are computed using the HLLC solver [48].

$$F_{i+1/2}^{HLLC} = F(U_{i+1/2}^L, U_{i+1/2}^R) \tag{12}$$

The state at edge mesh $U_{i+1/2}^L$ and $U_{i+1/2}^R$ is calculated with a WENO scheme [15] on the characteristic variables vectors $X_{i+1/2}^L$ and $X_{i+1/2}^R$ which reads

$$X = \big((p - \rho c u) \quad (p + \rho c^2) \quad (p + \rho c u)\big)^T \tag{13}$$

$X_{i+1/2}^L$ and $X_{i+1/2}^R$ are respectively calculated with the left stencil and the right one.

The maximum time step is estimated by $\Delta t_{HLLC} = max(\frac{\Delta x}{|u+c|}, \frac{\Delta x}{|u-c|})$.

Other Riemann solvers exist such the Roe solver [40], the HLL or HLLE solvers [7], but the HLLC solver is the most commonly used due to its accuracy to compute high Mach number flows such as shock waves propagation. However, if the Mach number is low, as for the simulations which are considered in this paper, its efficiency considerably decreases. Therefore, we are now going to introduce some specific numerical methods which are able to compute accurately compressible flows in the low Mach number regime.

2.2.2. Low Mach preconditioning

The misleading behavior of the HLLC solver for low Mach number can be theoretically justified [14,27,49] by demonstrating that the compressible Euler equations does not converge to the incompressible Euler equations when the Mach number M tends to zero. This demonstration, based on an asymptotic expansion depending on Mach number, is briefly reminded here.

Compressible Euler system limit when the Mach number M_a tends to zero The primitive formulation of the Euler equations (2) is expressed with dimensionless variables to carry out the asymptotic expansion. The variables are written as $\alpha = \tilde{\alpha}[\alpha]$, where $[\alpha]$ is a characteristic scale and $\tilde{\alpha}$ is the corresponding dimensionless variable. For sake of simplicity, a one-dimensional Cartesian coordinate system is still considered. Choosing $[u] = [x]/[t]$, the Euler dimensionless system can be expressed:

$$\begin{aligned} \frac{\partial \tilde{\rho}}{\partial \tilde{t}} + \tilde{u}\frac{\partial \tilde{\rho}}{\partial \tilde{x}} + \tilde{\rho}\frac{\partial \tilde{u}}{\partial \tilde{x}} &= 0 \\ \frac{\partial \tilde{u}}{\partial \tilde{t}} + \tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} + \frac{[p]}{[\rho][u]^2}\frac{1}{\tilde{\rho}}\frac{\partial \tilde{p}}{\partial \tilde{x}} &= 0 \\ \frac{\partial \tilde{p}}{\partial \tilde{t}} + \tilde{u}\frac{\partial \tilde{p}}{\partial \tilde{x}} + \frac{[\rho][c]^2}{[p]}\tilde{\rho}\tilde{c}^2\frac{\partial \tilde{u}}{\partial \tilde{x}} &= 0 \end{aligned} \tag{14}$$

The corresponding entropy dimensionless equation reads:

$$\frac{D\tilde{s}}{D\tilde{t}} = 0 \tag{15}$$

In order to show up the Mach number, we chose $[p] = [\rho][c]^2$. The dimensionless system becomes

$$\begin{aligned} \frac{\partial \tilde{\rho}}{\partial \tilde{t}} + \tilde{u}\frac{\partial \tilde{\rho}}{\partial \tilde{x}} + \tilde{\rho}\frac{\partial \tilde{u}}{\partial \tilde{x}} &= 0 \\ \frac{\partial \tilde{u}}{\partial \tilde{t}} + \tilde{u}\frac{\partial \tilde{u}}{\partial \tilde{x}} + \frac{1}{{M_a}^2}\frac{1}{\tilde{\rho}}\frac{\partial \tilde{p}}{\partial \tilde{x}} &= 0 \\ \frac{\partial \tilde{p}}{\partial \tilde{t}} + \tilde{u}\frac{\partial \tilde{p}}{\partial \tilde{x}} + \tilde{\rho}\tilde{c}^2\frac{\partial \tilde{u}}{\partial \tilde{x}} &= 0 \end{aligned} \tag{16}$$

We now calculate the limit of this previous system by an asymptotic analysis. Each variable is expressed as $\alpha = \alpha_0 + \epsilon\alpha_1 + \epsilon^2\alpha_2$ where $\epsilon \longrightarrow 0$. System (16) is expanded by considering that the Mach number tends to zero, so $Ma = \epsilon$. To alleviate the notation, $\sim$ is removed, but in the rest of the demonstration all the variables are dimensionless.

At order ϵ^{-2}, we obtain

$$\frac{\partial p_0}{\partial x} = 0 \tag{17}$$

at the order ϵ^{-1},

$$\frac{\partial p_1}{\partial x} = 0 \tag{18}$$

and at the order 0,

$$\begin{aligned} \frac{\partial \rho_0}{\partial t} + u_0\frac{\partial \rho_0}{\partial x} + \rho_0\frac{\partial u_0}{\partial x} &= 0 \\ \frac{\partial u_0}{\partial t} + u_0\frac{\partial u_0}{\partial x} + \frac{1}{\rho_0}\frac{\partial p_2}{\partial x} &= 0 \\ \frac{\partial p_0}{\partial t} + \rho_0 c_0^2\frac{\partial u_0}{\partial x} &= 0 \end{aligned} \tag{19}$$

We can observe that the limit of the compressible Euler system does not match to the incompressible Euler system which reads after the same asymptotic analysis procedure:

$$\begin{aligned} \rho &= cste \\ \frac{\partial u_0}{\partial x} &= 0 \\ \frac{\partial u_0}{\partial t} + \frac{1}{\rho_0}\frac{\partial p_2}{\partial x} &= 0 \end{aligned} \tag{20}$$

Indeed, the velocity divergence is different of zero. To insure that the velocity field will be divergence-free, a coefficient can be added in the pressure equation of the System (19) allowing to uncouple the two terms of the equation for the limit calculation. This idea has been initially proposed by Turkel [49].

$$\begin{aligned}\frac{\partial \rho_0}{\partial t}+u_0\frac{\partial \rho_0}{\partial x}+\rho_0\frac{\partial u_0}{\partial x}&=0\\ \frac{\partial u_0}{\partial t}+u_0\frac{\partial u_0}{\partial x}+\frac{1}{\rho_0}\frac{\partial p_2}{\partial x}&=0\\ \frac{\partial p_0}{\partial t}+M^2\rho_0 c_0^2\frac{\partial u_0}{\partial x}&=0\end{aligned} \tag{21}$$

This preconditioning allows the System (21) to tend to the incompressible System (20) when the Mach number tends to zero, with $M = M_a$. The System (21) is hyperbolic, and the following wave speeds can be defined: u, $u + c_+^*$ and $u - c_-^*$, where

$$\begin{aligned}c_+^* &= \frac{(1-M^2)u+\sqrt{(M^2-1)u^2+4M^2c^2}}{2}\\ c_-^* &= \frac{(M^2-1)u+\sqrt{(M^2-1)u^2+4M^2c^2}}{2}\end{aligned} \tag{22}$$

These waves speeds are only used for the computation of the fluxes, since the overall scheme remains identical to the HLLC solver [14] and the maximum time step is estimated by $\Delta t_{LM} = M\Delta t_{HLLC}$ [27].

2.2.3. Semi-implicit scheme: projection method

This method has been originally proposed by Yabe [55] and has been pursued by Xiao [53] and Kwatra et al. [21] for one phase flows. The main idea consists in a splitting of the Euler equations in two subsystems. The first one contains the convective part of the Euler equations, and the second one contains the acoustic part of the Euler equations. As the second part is solved with an implicit temporal discretization, the restrictive time step condition due to the propagation of acoustic waves is removed. Thus, the time step restriction only depends on the maximum convective velocity. As a result a significant benefit on the overall time step can be expected if this method is used in a low Mach number regime. For example in [21], the authors report a speed-up of 300 on the time step in comparison to a HLLC solver. Moreover, this compressible projection method is identical to the incompressible projection method if the Mach number tends to zero. This remarkable feature guarantees that this solver will behave accurately in the incompressible limit. The method is now briefly described by applying the following splitting to the system of Euler equations:

$$\begin{pmatrix}\rho\\ \rho\vec{u}\\ \rho E\end{pmatrix}_t + F_1 + F_2 = 0 \tag{23}$$

with

$$F_1 = \nabla.\begin{pmatrix}\rho\vec{u}\\ \rho\vec{u}\otimes\vec{u}\\ \rho E\vec{u}\end{pmatrix} \tag{24}$$

and

$$F_2 = \begin{pmatrix}0\\ \nabla p\\ \nabla\cdot(p\vec{u})\end{pmatrix} = 0 \tag{25}$$

F_1 is the advection part and F_2 the non-advection part.

Advection sub-system The first sub-system contains the mass conservation terms, and the advection part of velocity and pressure equations. It reads:

$$\begin{aligned}\frac{\partial \rho}{\partial t}+\vec{u}\cdot\nabla\rho+\rho\nabla\cdot\vec{u}&=0\\ \frac{\partial \vec{u}}{\partial t}+\vec{u}\cdot\nabla\vec{u}&=0\\ \frac{\partial p}{\partial t}+\vec{u}\cdot\nabla p&=0\end{aligned} \tag{26}$$

In one-dimensional Cartesian coordinate, the advection sub-system reads

$$\frac{\partial W}{\partial t}+A(W)\frac{\partial W}{\partial x}=0 \tag{27}$$

with

$$A(W) = \begin{pmatrix} u & \rho & 0 \\ 0 & u & 0 \\ 0 & 0 & u \end{pmatrix} \tag{28}$$

The matrix A has 3 eigenvalues which are all $\lambda = u$, and three left eigenvectors:

$$l_1 = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}, l_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, l_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}.$$

Because l_1 is zero, the A matrix is non-diagonalizable, consequently the advection sub-system is not hyperbolic. Nevertheless, the time step restriction is calculated with $\Delta t = \frac{\Delta x}{Max(u)}$. This sub-system is solved with an explicit temporal discretization.

Acoustical sub-system The remaining terms of the Euler system are grouped in the second sub-system. It corresponds to the acoustic part of the main system. If an explicit HLLC solver is used, this acoustic part is responsible for the stringent time step restriction due to the propagation of acoustic waves.

$$\begin{aligned} \frac{\partial \rho}{\partial t} &= 0 \\ \frac{\partial \vec{u}}{\partial t} + \frac{\nabla p}{\rho} &= 0 \\ \frac{\partial p}{\partial t} + \rho c^2 \nabla \cdot \vec{u} &= 0 \end{aligned} \tag{29}$$

In one-dimensional Cartesian coordinates, System (29) reads

$$\frac{\partial W}{\partial t} + A(W)\frac{\partial W}{\partial x} = 0 \tag{30}$$

with

$$A(W) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & \frac{1}{\rho} \\ 0 & \rho c^2 & 0 \end{pmatrix} \tag{31}$$

Three eigenvalues of A matrix are $\lambda_1 = -c$, $\lambda_2 = 0$ and $\lambda_3 = +c$, and its three left eigenvectors are

$$l_1 = \begin{pmatrix} 0 \\ -\rho c \\ 1 \end{pmatrix}, l_2 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, l_3 = \begin{pmatrix} 0 \\ \rho c \\ 1 \end{pmatrix}.$$

Consequently, System (29) is hyperbolic which is relevant to correctly compute the acoustical waves propagation. If an implicit temporal discretization is applied to solve this subsystem, the 'acoustical' time step restriction will be removed, the restriction will be based on the maximum of the convection velocity u, and neither on $u + c$ nor $u - c$ as it would be with a fully explicit solver.

Time discretization Following the semi-implicit strategy introduced above, the time discretization of System (2) is now described:

$$\begin{aligned} \frac{\rho^{n+1} - \rho^n}{\Delta t} + \nabla \cdot (\rho \vec{u})^n &= 0 \\ \frac{\vec{u}^{n+1} - \vec{u}^n}{\Delta t} + (\vec{u} \cdot \nabla \vec{u})^n + \frac{(\nabla p)^{n+1}}{\rho^{n+1}} &= 0 \\ \frac{p^{n+1} - p^n}{\Delta t} + (\vec{u} \cdot \nabla p)^n + (\rho c^2)^{n+1} (\nabla \cdot \vec{u})^{n+1} &= 0 \end{aligned} \tag{32}$$

Firstly, the fluid density is updated by solving the mass conservation equation which is used in a primitive formulation:

$$\rho^{n+1} = \rho^n - \Delta t((\vec{u} \cdot \nabla \rho)^n + \rho^n \nabla \cdot \vec{u}^n) \tag{33}$$

To compute the pressure, the energy conservation equation can be expressed as follows:

$$p^{n+1} + \Delta t(\rho c^2)^{n+1} (\nabla \cdot \vec{u})^{n+1} = p^n - \Delta t(\vec{u} \cdot \nabla p)^n \tag{34}$$

The following time splitting, typical of incompressible projection methods, can next be applied:

$$\vec{u}^* = \vec{u}^n - \Delta t \vec{u}^n \cdot \nabla \vec{u}^n \tag{35}$$

and

$$\vec{u}^{n+1} = \vec{u}^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}} \tag{36}$$

By using the equation (36), we can express the divergence of the velocity field as a function of u^* and p^{n+1}.

$$\nabla \cdot \vec{u}^{n+1} = \nabla \cdot \vec{u}^* - \Delta t \nabla \cdot \left(\frac{\nabla p^{n+1}}{\rho^{n+1}} \right) \tag{37}$$

This relation can be injected in the pressure equation (34) in order to obtain the following system for the pressure:

$$p^{n+1} - \Delta t^2 (\rho c^2)^{n+1} \nabla \cdot \left(\frac{\nabla p}{\rho} \right)^{n+1} = p^* - \Delta t (\rho c^2)^{n+1} \nabla \cdot \vec{u}^* \tag{38}$$

with $p^* = p^n - \Delta t (\vec{u} \cdot \nabla p)^n$ the pressure solution of the advection sub-system.

The system (38) is a Helmholtz-type equation which can be solved as a linear system provided we can compute an approximation of $(\rho c^2)^{n+1}$. Once the pressure field is known, the velocity field can be computed with the Eq. (36).

The equation (38) is a discrete formulation resulting from System (2). We have to remark that this formulation is correct if isentropic flows are considered. This condition is always respected in this paper since viscous effects and heat conduction are neglected. The way we use to formulate the conservation energy equation would be still valid for entropic flows but only for a perfect gas equation of state, provided that additional terms due to viscous friction and heat conduction were added. A more general framework which allows dealing with compressible two-phase flows involving viscous friction, heat conduction with any equation of state is proposed in [4]. In this reference, a pressure evolution equation has been derived, this equation looks similar to Eq. (38), but it accounts for an additional term on the coupling between the pressure evolution and the entropic temperature evolution.

3. Governing equations and numerical methods for two-phases flows

To our knowledge, the compressible projection method has never been applied to multidimensional simulations of two-phase flows accounting for surface tension effect. We propose in the rest of this paper a numerical method based on a new time splitting in order to achieve this task. The relevance of our new method will be highlighted in the result section of this paper.

3.1. Governing equations

The equations are identical to the one phase case (System (2)), except for the momentum conservation equation where the surface tension force must be included.

$$\begin{aligned} &\frac{\partial \rho}{\partial t} + \vec{u} \cdot \nabla \rho + \rho \nabla \cdot \vec{u} = 0 \\ &\frac{\partial \vec{u}}{\partial t} + \vec{u} \cdot \nabla \vec{u} + \frac{\nabla p}{\rho} = \frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho} \\ &\frac{\partial p}{\partial t} + \vec{u} \cdot \nabla p + \rho c^2 \nabla \cdot \vec{u} = 0 \end{aligned} \tag{39}$$

Where σ is the surface tension coefficient, κ is the local curvature of the interface, $\vec{n}$ is the normal vector at the interface, and δ_Γ is a Dirac distribution localized on the interface Γ.

Another significant difference lies in the description of the density field which is discontinuous in the grid cells which are crossed by the interface. Some equations of state have been described earlier to compute the speed sound in the gas phase (perfect gas law) and in the liquid phase (Tait equation).

3.2. Level-set method

The interface motion is captured by solving the following convection equation of a Level Set Function [36]:

$$\frac{\partial \phi}{\partial t} + \vec{u}_{int} \cdot \nabla \phi = 0 \tag{40}$$

where $\vec{u}_{int}$ is the interface velocity. If the normal component of the velocity field is continuous across the interface the interface velocity can be easily extrapolated by using the fluid velocity field. It is noteworthy to remind that such a simple

extrapolation could not be used in more complex situations where shock waves or phase change occur since these situations involve a discontinuity of the normal component of the velocity field, see the following references for more details on this topic [3,11,34,45,46]. However as the test-cases which are considered in this paper do not induce any shock waves or discontinuities of the normal velocity component, we have observed that the standard extrapolation of the velocity field used for incompressible flows, $\vec{u}_{int} = \vec{u}$, was well suited for the simulations presented in this paper.

The non-zero level curves of the level-set function are useful to compute interface geometrical properties such as the curvature:

$$\kappa = \nabla \cdot \vec{n} \tag{41}$$

where

$$\vec{n} = \frac{\nabla \phi}{\|\nabla \phi\|} \tag{42}$$

A reinitialization equation must be solved at the end of each time step to ensure that the Level Set function will remain a signed distance [43]. It is well known that this further equation can induce some slight displacements of the interface position during the reinitialization. However if sufficiently refined grids are used this spurious displacement tends toward zero.

$$\frac{\partial d}{\partial \tau} = sign(\phi)(1 - \|\nabla d\|) \tag{43}$$

In this equation, τ is a fictitious time. The steady solution of Eq. (43) provides a function d which is the signed distance to the interface.

3.3. Projection method

3.3.1. Basic projection method

We describe in this section, a first attempt to incorporate the surface tension in the framework of the compressible projection method. The method, which is proposed here, is directly instigated by the classical approach used to compute the surface tension term for incompressible two-phase flows. In particular, the following splitting is carried out on the velocity field:

$$\vec{u}^* = \vec{u}^n - \Delta t \vec{u}^n \cdot \nabla \vec{u}^n \tag{44}$$

$$\vec{u}^{n+1} = \vec{u}^* - \Delta t \frac{\nabla p^{n+1}}{\rho^{n+1}} + \Delta t \left(\frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho} \right)^{n+1} \tag{45}$$

This splitting is used in the following works [19,23,44] in the framework of the Ghost Fluid Method for incompressible flows. This method allows imposing sharp jump conditions across the interface with an accurate discretization of the singular terms. Sharp approximations of singular terms as δ_Γ have been derived in [30] in order to avoid an artificial smoothing of these terms on the interface. If a compressible projection method is considered, it leads naturally to the following Helmholtz equation for the pressure:

$$p^{n+1} - \Delta t^2 (\rho c^2)^{n+1} \nabla \cdot \left(\frac{\nabla p}{\rho} \right)^{n+1} = p^* - \Delta t (\rho c^2)^{n+1} \nabla \cdot \vec{u}^* + \Delta t (\rho c^2)^{n+1} \nabla \cdot \left(\frac{\sigma \kappa \vec{n} \delta_\Gamma}{\rho} \right)^{n+1} \tag{46}$$

In a first time, we have applied this quite simple method, but it is illustrated in Fig. 9 that it provides very poor results due to strong parasitic currents which develop on the interface. Whereas the Ghost Fluid Method is known as an accurate method to compute the surface tension in the framework of projection methods for incompressible flows since it produces parasitic currents with very low amplitudes [19,44], its generalization to the compressible projection method seems to be more challenging. For more details, refer to Appendix A.

3.3.2. Splitting method for surface tension resolution

We present now a new time splitting method in order to perform accurate and stable computations accounting for surface tension effects. As stated above, the Ghost Fluid Method provides an accurate and stable representation of the surface tension effects if it is used with an incompressible projection method. According to the Helmholtz–Hodge theorem, the compressible velocity field can be decomposed as the sum of the gradient of a scalar field with a divergence-free vector field. We propose to incorporate the surface tension in the part of the velocity field which is divergence-free. Thus, the surface tension term will be computed in a similar way as for an incompressible projection method. For that purpose, we isolate the surface tension contribution on the solutions of system (39) by splitting each variable as following:

$$\begin{aligned} \rho &= \rho_0 + \rho_{ST} \\ \vec{u} &= \vec{u}_0 + \vec{u}_{ST} \\ p &= p_0 + p_{ST} \end{aligned} \tag{47}$$

with the subscript 'ST' which indicates the corresponding variable takes into account the surface tension effects. The vector field $\vec{u}_{ST}$ is the divergence-free part of the Hodge decomposition of the velocity field. Using this decomposition, the system (39) can be splitted in two sub-system (48) and (49). Transport and acoustical terms are contained in the first sub-system:

$$\begin{aligned} \frac{\partial \rho_0}{\partial t} + \vec{u}\cdot\nabla\rho_0 + \rho\nabla\cdot\vec{u}_0 &= 0 \\ \frac{\partial \vec{u}_0}{\partial t} + \vec{u}\cdot\nabla\vec{u} + \frac{\nabla p_0}{\rho} &= 0 \\ \frac{\partial p_0}{\partial t} + \vec{u}\cdot\nabla p_0 + \rho c^2\nabla\cdot\vec{u}_0 &= 0 \end{aligned} \tag{48}$$

and the surface tension terms are included in the second sub-system (49):

$$\begin{aligned} \frac{\partial \rho_{ST}}{\partial t} + \vec{u}\cdot\nabla\rho_{ST} + \rho\nabla\cdot\vec{u}_{ST} &= 0 \\ \frac{\partial \vec{u}_{ST}}{\partial t} + \frac{\nabla p_{ST}}{\rho} &= \frac{\sigma\kappa\vec{n}\delta_\Gamma}{\rho} \\ \frac{\partial p_{ST}}{\partial t} + \vec{u}\cdot\nabla p_{ST} + \rho c^2\nabla\cdot\vec{u}_{ST} &= 0 \end{aligned} \tag{49}$$

Spurious currents due to surface tension come from the second sub-system. Recall that $\vec{u}_{ST}$ is defined to respect the divergence-free condition. System (49) is incompressible: $\nabla.\vec{u}_{ST} = 0 \Longleftrightarrow \rho_{ST} = cste$. For sake of simplicity, we choose $\rho_{ST} = 0$ and $\rho = \rho_0$. The System (49) becomes simply:

$$\begin{aligned} \frac{\partial \vec{u}_{ST}}{\partial t} + \frac{\nabla p_{ST}}{\rho} &= \frac{\sigma\kappa\vec{n}\delta_\Gamma}{\rho} \\ \nabla\cdot\vec{u}_{ST} &= 0 \end{aligned} \tag{50}$$

Thus, the pressure p_{ST} field can be classically computed by solving a Poisson equation:

$$\nabla\cdot\left(\frac{\nabla p_{ST}}{\rho}\right) = \nabla\cdot\left(\frac{\sigma\kappa\vec{n}\delta_\Gamma}{\rho}\right) \tag{51}$$

Finally, the whole system, that we have to solve, can be summarized as follows:

$$\begin{aligned} \frac{\partial \rho}{\partial t} + \vec{u}\cdot\nabla\rho + \rho\nabla\cdot\vec{u} &= 0 \\ \frac{\partial \vec{u}_0}{\partial t} + \vec{u}\cdot\nabla\vec{u} + \frac{\nabla p_0}{\rho} &= \vec{0} \\ \frac{\partial p_0}{\partial t} + \vec{u}\cdot\nabla p_0 + \rho c^2\nabla\cdot\vec{u}_0 &= 0 \\ \frac{\partial \vec{u}_{ST}}{\partial t} + \frac{\nabla p_{ST}}{\rho} &= \frac{\sigma\kappa\vec{n}\delta_\Gamma}{\rho} \\ \nabla\cdot\vec{u}_{ST} &= 0 \end{aligned} \tag{52}$$

where $\vec{u} = \vec{u}_0 + \vec{u}_{ST}$ and $p = p_0 + p_{ST}$.

3.3.3. Time discretization

We describe now the temporal discretization used to solve the overall System (52). In a first time, the interface motion is computed:

$$\phi^{n+1} = \phi^n - \Delta t(\vec{u}\cdot\nabla\phi)^n \tag{53}$$

Next, the density field can be updated by solving the following equation:

$$\rho^{n+1} = \rho^n - \Delta t\left((\vec{u}\cdot\nabla\rho)^n + \rho^n\nabla\cdot\vec{u}^n\right). \tag{54}$$

In the third step, the pressure p_{ST} field is solved due to the Eq. (51) corresponding to a Poisson equation which leads to a symmetric positive definite linear system:

$$\nabla\cdot\left(\frac{\nabla p_{ST}}{\rho}\right)^{n+1} = \nabla\cdot\left(\frac{\sigma\kappa\vec{n}\delta_\Gamma}{\rho}\right)^{n+1} \tag{55}$$

The fourth step consists in computing the intermediate velocity $\vec{u}_0^*$ and the pressure p_0^* fields:

$$\vec{u}_0^* = \vec{u}_0^n - \Delta t \vec{u}^n \cdot \nabla \vec{u}^n \tag{56}$$

$$p_0^* = p_0^n - \Delta t \vec{u}^n \cdot \nabla p_0^n \tag{57}$$

In the fifth step, the following Helmholtz type Eq. (58) for the pressure p_0 is computed by solving a linear system:

$$\frac{p_0^{n+1}}{(\rho c^2)^{n+1}} - \Delta t^2 \nabla \cdot \left(\frac{\nabla p_0}{\rho} \right)^{n+1} = \frac{p_0^*}{(\rho c^2)^{n+1}} - \Delta t \nabla \cdot \vec{u}_0^* \tag{58}$$

Let us notice, that the resolution of this equation has been eased by dividing the Eq. (38) by 'ρc^2', in order to obtain a symmetric definite positive linear system. $(\rho c^2)^{n+1}$ is computed knowing the density at time $(n+1)$ thanks to Eq. (33). As the flow is assumed to be isentropic, the gas obeys to the Laplace law: $\frac{p}{\rho^\gamma} = cste$. Thanks to Eq. (8), we obtain $(\rho c^2)^{n+1} = \gamma p_0 \left(\frac{\rho^{n+1}}{\rho_0} \right)$ for air. For water, we just use Eq. (9): $(\rho c^2)^{n+1} = \gamma B \left(\frac{\rho^{n+1}}{\rho_0} \right)$.

Finally, the velocity field is computed by

$$\vec{u}^{n+1} = \vec{u}_0^{n+1} + \vec{u}_{ST}^{n+1} = \vec{u}_0^* + \vec{u}_{ST}^n - \Delta t \left(\frac{\nabla p_0^{n+1}}{\rho^{n+1}} + \frac{\nabla p_{ST}^{n+1}}{\rho^{n+1}} - \left(\frac{\sigma \kappa \vec{n} \delta}{\rho} \right)^{n+1} \right) \tag{59}$$

The redistance algorithm is then applied at the end of the temporal iteration.

3.3.4. Spatial discretization

The spatial discretizations are detailed in this section for a 2D Cartesian mesh, but it can be easily generalized to an axisymmetric mesh or a 3D Cartesian mesh. All of the convective derivatives are computed using a WENO scheme [15] which is well-known for its accuracy and stability. As it is usual for an incompressible projection method, velocity components are computed on staggered grids relative to the centered grid for the interface, density and pressure fields. Thus, the velocity divergence is computed as follows:

$$\nabla \cdot \vec{u}_{i,j} = \frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} + \frac{v_{i,j+1/2} - v_{i,j-1/2}}{\Delta y} \tag{60}$$

where u and v are the velocity components following the x-axis and the y-axis, and the components of the pressure gradient are

$$\left. \frac{\partial p}{\partial x} \right)_{i+1/2,j} = \frac{p_{i+1,j} - p_{i,j}}{\Delta x} \tag{61}$$

$$\left. \frac{\partial p}{\partial y} \right)_{i,j+1/2} = \frac{p_{i,j+1} - p_{i,j}}{\Delta y} \tag{62}$$

This standard discretization of the Laplace operator is used to compute the pressure p_0 and p_{ST}:

$$\nabla \cdot \left(\frac{\nabla p}{\rho} \right)_{i,j}^{n+1} = \frac{\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\rho_{i+1/2,j}^{n+1}} - \frac{p_{i,j}^{n+1} - p_{i-1,j}^{n+1}}{\rho_{i-1/2,j}^{n+1}}}{\Delta x^2} + \frac{\frac{p_{i,j+1}^{n+1} - p_{i,j}^{n+1}}{\rho_{i,j+1/2}^{n+1}} - \frac{p_{i,j}^{n+1} - p_{i,j-1}^{n+1}}{\rho_{i,j-1/2}^{n+1}}}{\Delta y^2} \tag{63}$$

For the segments of the grid which are crossed by the interface, the density is interpolated on the border of the computational cells with a harmonic average as it is usually done in the framework of the Ghost Fluid Method for incompressible two-phase flows [19,30,44]. The discretization of the surface tension term in the Eq. (55) is also performed following the efficient approach developed in these papers. As previously stated, this approach allows a sharp description of the singular terms by remarking the equivalency between Eq. (51) and Eq. (64):

$$\nabla \cdot \left(\frac{\nabla p_{ST}}{\rho} \right)^{n+1} = 0 \text{ with } [p_{ST}]_\Gamma = \sigma \kappa \tag{64}$$

See [23,29] for more details on this specific point.

3.3.5. Ghost fluid method

Since the pioneering works of [8,19,30], the Ghost Fluid Method became a popular approach to deal with discontinuities in heterogeneous media, see for example the following Refs. [1,3,11,12,23,34,47,45,46,51]. However, it is noteworthy to remind that global denomination can refer to many different techniques. In particular, if the basic principle of the Ghost Fluid Method is always the same, its implementation can be very different whether an explicit or an implicit resolution is performed. Indeed, if an explicit resolution is considered, an extrapolation of a known field must be computed on the other side of the interface in order to build a continuous field before the discretization. On the other hand, if an implicit discretization is carried out, the singular terms are interpolated on the interface in order to impose the correct value of the

jump conditions. Thus, as the development presented above was mainly focused on the computation of the pressure, we have essentially referred to numerical tools well suited to impose jump conditions with an implicit temporal discretization. Unlike the pressure field, an explicit algorithm is used to update the density field. Therefore, it is required to build a continuous extrapolation of the density field of each phase on the other side of the interface. It is performed by transporting the corresponding density in the ghost fields along the normals $\vec{n}$ to the interface by an iterative resolution of the following equation [8]:

$$\frac{\partial \rho}{\partial \tau} \pm \vec{n} \cdot \nabla \rho = 0 \tag{65}$$

where τ is a fictitious time. Moreover, the density values close to the interface (in a layer thickness of one cell in the real field) are updated by the extended field computed with Eq. (65). This method, known as the "isobaric fix", has been proposed in [9] to prevent from a density overshooting close to the interface. Let us remark that it is not required to use any extrapolation to compute the explicit convective derivative on pressure in Eq. (48) since this split pressure does not contain the jump condition due to the surface tension effects.

3.3.6. Some considerations on the time step stability and on the cost of a temporal iteration

A classical time-step constraint accounting for the stability conditions on convection, and surface tension is imposed if the compressible projection method is used.

$$\Delta t_{conv} = \frac{1}{2} \frac{\Delta x}{\max \|\vec{u}\|} \tag{66}$$

$$\Delta t_{Surf\_Tens.} = \frac{1}{4} \sqrt{\frac{\rho_{liq} \Delta x^3}{\sigma}} \tag{67}$$

Finally, the global time step restriction can be computed with the following relation:

$$\frac{1}{\Delta t} = \frac{1}{\Delta t_{conv}} + \frac{1}{\Delta t_{Surf\_Tens.}} \tag{68}$$

As it has been explained previously, for low Mach number flows, the compressible projection method strongly alleviates the time step restriction in comparison to an explicit HLLC solver. However, one could point out this improvement on the temporal stability leads to an important additional cost of a temporal iteration since one linear system must be solved to compute the pressure field (two linear systems if the surface tension is taken into account), unlike the classical HLLC solver, which is only based on an explicit temporal discretization. This last assertion could be relevant in some situations, for instance if an Incomplete Cholesky Conjugate Gradient (ICCG) method is used to solve the linear system. But herein, the Black Box Multi-Grid method [6] is used to compute the solution of our linear systems. This method, which is perfectly suited to deal with Poisson (or Helmholtz) type-equation with strongly inhomogeneous diffusion coefficient [31], allows saving an important computational time, to such an extent that solving the linear system is no longer the temporal limiting factor, but instead the many calls to the WENO scheme for computing the convective derivatives. Therefore, the cost of a temporal iteration can be approximately the same with a compressible projection method or a HLLC solver.

4. Results

4.1. Rayleigh–Plesset theory

The Rayleigh–Plesset equation describes the radius temporal evolution of a spherical bubble surrounded by an incompressible fluid in an infinite medium. It reads

$$R \frac{d^2 R}{dt^2} + \frac{3}{2} \left(\frac{dR}{dt} \right)^2 = \frac{p_B(t) - \frac{2\sigma}{R} - p_\infty}{\rho_L} \tag{69}$$

where R is the bubble radius, p_B the pressure in the bubble supposed to be uniform, p_∞ the pressure of the liquid far from the bubble and σ the surface tension coefficient.

It is derived from the incompressible Euler equations (20) in one-dimensional spherical geometry, as shown in [26]. No analytical solution has been found yet, but a numerical solution can be easily computed.

Yang and Prosperetti [57] proposed an extension of the Rayleigh–Plesset equation accounting for the containment effects by supposing that the incompressible liquid domain is finite. This modified equation is expressed as follows:

$$\left(\frac{S - R}{S} \right) R \frac{d^2 R}{dt^2} + [2 - \frac{(S^2 + R^2)(S + R)}{2S^3}] \left(\frac{dR}{dt} \right)^2 = \frac{p_B(t) - \frac{2\sigma}{R} - p_S(t)}{\rho_L} \tag{70}$$

where S is the radius of the incompressible liquid domain, and p_S the pressure of the liquid at the distance S of the bubble center.

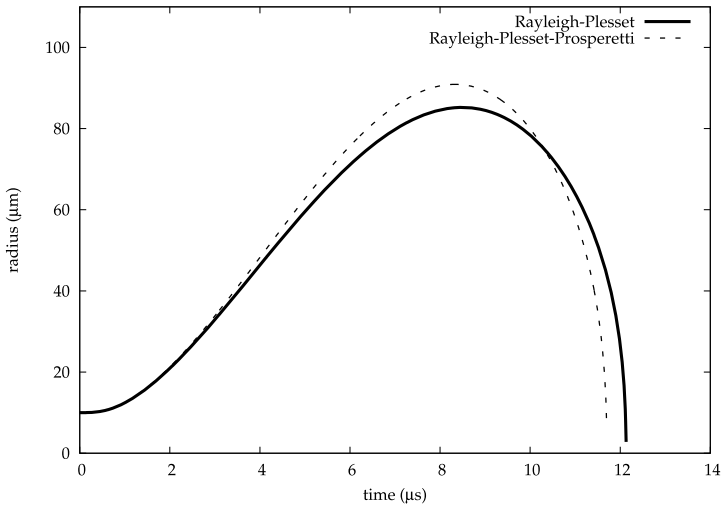


Fig. 1. Comparison between Rayleigh–Plesset and Rayleigh–Plesset–Prosperetti equation. The containment cannot be neglected.

The influence of the containment effect on the temporal evolution of the bubble radius is investigated on the following configuration presented in [42]. An air bubble is surrounded by water. On the domain boundary, a spherical ultrasound wavefront is generated. The surface tension is neglected. R_0 is the initial radius. The wavefront is simulated by setting the pressure: $p_S(t) = p_0 - Asin(wt)$. For this test case, $R_0 = 10^{-5}$ m, $S = 32 \times 10^{-5}$ m, $p_0 = 10^5$ Pa, $A = 4 \times 10^5$ Pa and $w = 5 \times 10^5$ rad s^{-1}.

The bubble radius evolution has been plotted in the Fig. 1 to compare the solution of the Rayleigh–Plesset equation with the solution of the Rayleigh–Plesset–Prosperetti equation. We observe that the amplitude of the radius oscillation is increasing and the time of the collapse is reduced if the containment effect is accounted for. As we target to design a relevant benchmark for the validation of our numerical approaches, we can conclude from this preliminary test that benchmark should be based on a reference solution which includes the containment effect in our reference solution, since a simulation must be performed in a finite size computational domain.

The influence of the liquid compressibility can be considered by solving the more complex Keller–Miksis equation [20]. This equation provides similar results as the Rayleigh–Plesset equation on the configuration which is considered above. To our knowledge, it does not exist any equations accounting for both the containment and the compressibility of the liquid phase. Therefore, as it has been demonstrated above that the containment should not be neglected, the solution of the Rayleigh–Plesset–Prosperetti equation will be considered in the rest of this paper as our reference solution.

4.2. Comparison between the different compressible numerical methods on a 1D spherical bubble oscillating test case

In this section, we perform a comparative study between the different numerical methods which have been previously presented by using the test-case described in Section 4.1. This preliminary test is computed in a one dimensional spherical coordinate system in order to assess the suitability of the different solvers to deal with low Mach number flows. Therefore, as it is not required to account for the surface tension, it will be neglected.

In a first time, the HLLC solver has been tested. The results are presented in Fig. 2 for different mesh sizes. As it has been previously reported in [42], if this solver is used, a very important number of grid cells is required to obtain an accurate solution. This test illustrates the low efficiency of such a solver in the low Mach number regime, since the maximum Mach number measured in the water is about 10^{-2}. As expected, we can conclude this solver is not suitable to perform numerical simulations in the low Mach number regime of the interaction between a liquid–gas interface with acoustical or ultrasound waves.

Assessments of the preconditioning Low Mach HLLC solver are now presented on the same test-case for several values of the Mach coefficient. The bubble radius is discretized with 16 computational cells at the initial time. $M = 1$ corresponds to the HLLC solver without the Low Mach preconditioning. It is clear in Fig. 3 that the lower the Mach coefficient is, the more accurate are the results. Due to the Low Mach preconditioning, a sufficient accuracy can be achieved with much less computational cells. However, if this method is used, the time step which was already small with the classical HLLC solver, is even smaller. An implicit computation method can be performed [27], but it is much more complex since it leads to a coupled non-linear system.

The results obtained with the compressible projection method have been plotted in Fig. 4. We first note that the spatial convergence is strongly enhanced in comparison to the results obtained with the HLLC solver. In particular, relevant results can be obtained with very few points in the initial bubble radius. Secondly, due to the semi-implicit temporal discretization, the time step is 16 times larger than Δt_{HLLC}. Thus, if a low Mach number flow is considered, in comparison to the HLLC

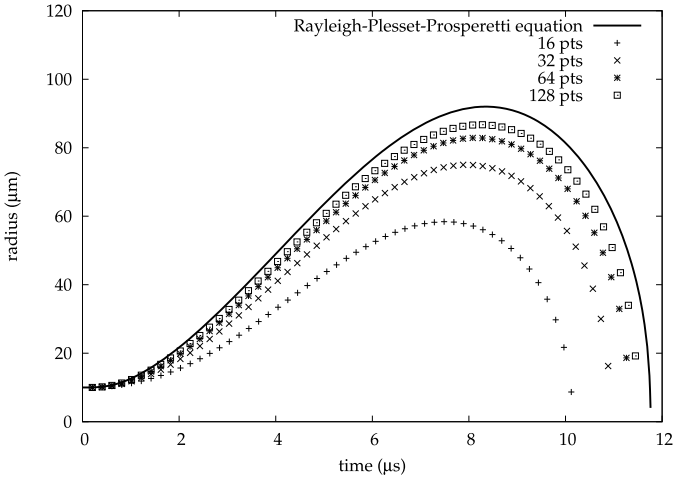


Fig. 2. Convergence study of HLLC solver on the Rayleigh–Plesset–Prosperetti test case. The number of cells which is indicated is the one contained in the initial radius of the bubble. This number must be multiplied by 32 to obtain the total number of cells.

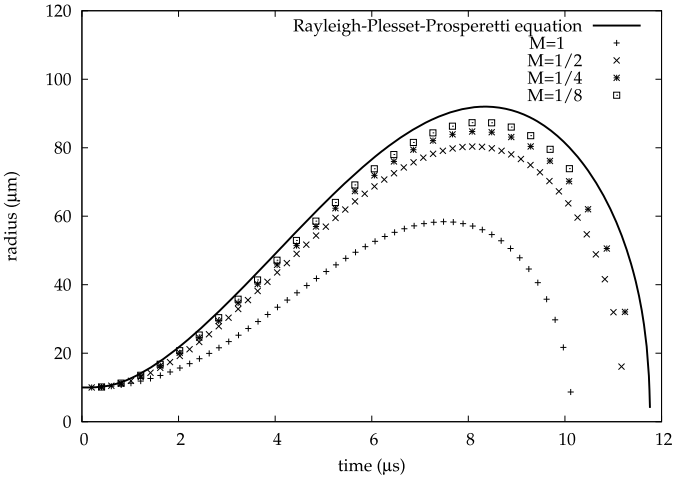


Fig. 3. Influence of the M parameter on the spatial accuracy of the Low Mach Preconditioning method with 16 computational cells on the initial bubble radius. The time step is M times smaller than the acoustical time step from the HLLC solver.

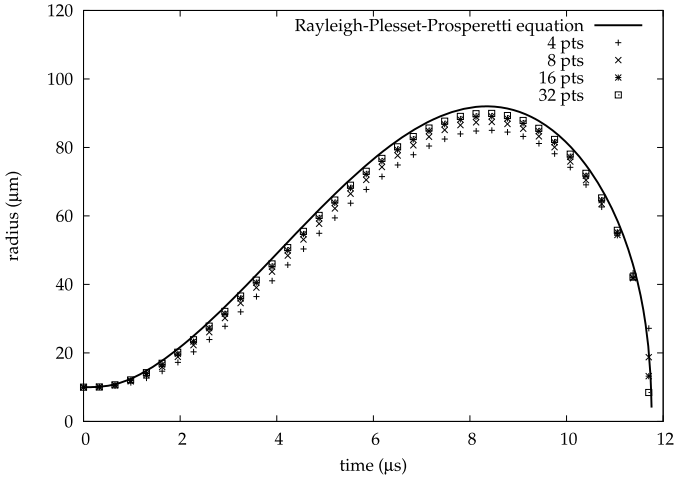


Fig. 4. Convergence study of the compressible projection method. The number of cells which is indicated is the one contained in the initial radius of the bubble. The total number of cells is 32 times higher.

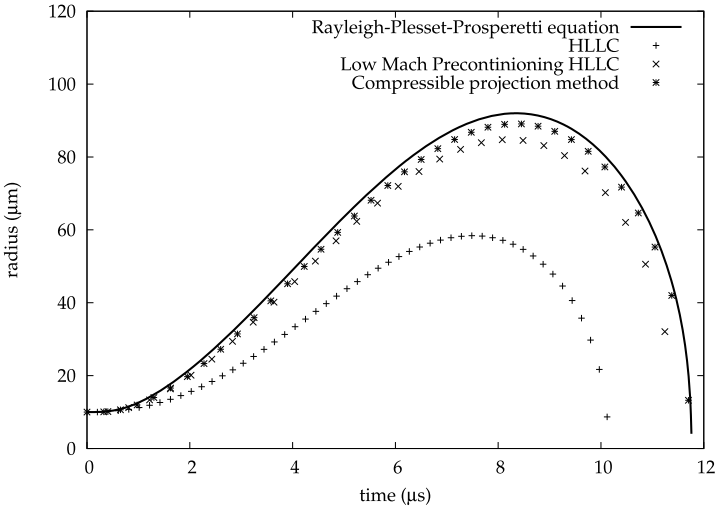


Fig. 5. Comparison between the different numerical methods with 16 computational cells to discretize the initial radius of the bubble. The Mach coefficient of preconditioning is set to $M = 1/4$.

Table 1
Convergence study of the different numerical methods. N is the number of cells in the initial bubble radius.

N	HLLC		Preconditioning HLLC ($M = \frac{1}{4}$)		Projection method	
	Relative error	Order	Relative error	Order	Relative error	Order
4	91.84%		35.69%		11.95%	
8	74.76%	0.30	15.45%	1.21	6.34%	0.65
16	36.86%	1.02	6.99%	1.14	3.23%	0.92
32	18.03%	1.03	3.31%	1.08	1.58%	0.97
64	8.98%	1.01	1.65%	1.01	0.75%	1.03
128	4.71%	0.93	0.86%	0.94	0.33%	1.18
256	3.46%	0.44			0.12%	1.47

solver, this method allows saving computational time both by using a fewer number of grid cells and by performing a fewer number of temporal iterations.

The results obtained with the three different solvers are plotted in Fig. 5 for computations involving 16 computational cells to discretize the initial bubble radius. As stated before, the HLLC solver provides very poor results in comparison to other solvers. Moreover, its time step restriction is stringent, since its value is about $\Delta t_{HLLC} = 2 \times 10^{-10}$ s. The Low Mach preconditioning HLLC method increases significantly the spatial accuracy of the HLLC solver. However, the time step restriction is even more stringent than with the simple HLLC solver, since it is equal to: $\Delta t_{HLLC_{Precond}} = 5 \times 10^{-11}$ s. Finally, it highlights that the spatial accuracy of the compressible projection method is improved in comparison to the other two. As this method also allows using a higher time step (16 times than HLLC one): $\Delta t_{Proj} = 32 \times 10^{-10}$ s, we can conclude from these preliminary simulations that the compressible projection method outclasses the other two for computing the volume oscillations of a bubble in a low Mach number regime.

In the Table 1, the evolution of the error on the bubble radius is reported for different computational grids and at a given time. We can conclude that the three solvers are first order accurate in space when the coarsest grids are used, whereas the values of the errors are very different. Moreover we can observe that the accuracy of the compressible projection method increases when the spatial resolution increases, it means that the computation tends toward an asymptotic regime which could lead to an almost second order accuracy. On the other side, the accuracy of the HLLC Solvers seems to decrease when the spatial resolution is increasing. It confirms that these solvers are not well suited to compute this kind of applications involving low Mach number regime.

Therefore, this compressible projection method will be used in the following to perform multidimensional computations accounting for surface tension.

4.3. Multidimensional computations of the interaction of a bubble with an ultrasound wavefront

4.3.1. Shape oscillation

We present in this section some computations of the shape oscillation of a deformed bubble. This classical test-case for the numerical simulation of incompressible two-phase flows [10,22] will allow us bringing out the suitability of the time

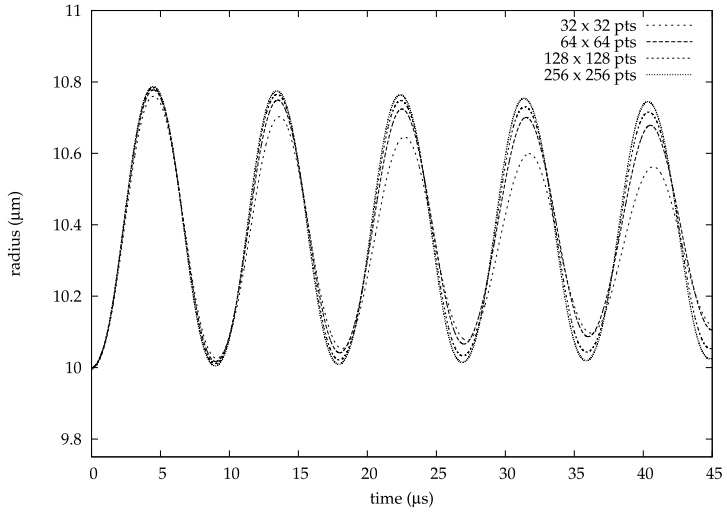


Fig. 6. Temporal evolution of the bubble displacement in the x direction.

Table 2
Comparison of the shape oscillation frequency between theory and computations.

Number of cells	Oscillation frequency (kHz)	Relative error (%)
32×32	110.62	1.45
64×64	111.09	1.03
128×128	111.46	0.70
256×256	111.53	0.63

splitting projection scheme for compressible two-phase flows in the strictly incompressible limit. The oscillation frequency is provided by the following relation [24]:

$$f_l = 2\pi \sqrt{\frac{\sigma}{R^3} \frac{(l-1)l(l+1)(l+2)}{l\rho_L + (l+1)\rho_G}} \tag{71}$$

where R is the equivalent radius of the droplet, and l the oscillation mode number. In this test case, we only consider the mode $l = 2$, and $\rho_L = 10^3$ kg m^{-3}, $\rho_G = 1.226$ kg m^{-3} and $\sigma = 7 \times 10^{-2}$ Pa m.

Initially, the interface is defined by an ellipsoid equation:

$$\frac{x^2}{a^2} + \frac{y^2}{b^2} + \frac{z^2}{c^2} = 1 \tag{72}$$

with $a = 10^{-5}$ m, $b = 10^{-5}$ m and $c = 1.25 \times 10^{-5}$ m. This corresponds to $R = 1.077 \times 10^{-5}$ m.

The numerical computations are performed on an axisymmetric Cartesian mesh. Moreover, an additional symmetry is imposed following the radial plane in order to save computational time. The whole computational domain is a square with a length side equal to 2×10^{-5} m.

We observe the position of the interface crossing the radial plane over time in Fig. 6, and we compare the oscillations frequency for different meshes with the theoretical frequency $f_2 = 112{,}24$ kHz. A good agreement between the computations and the theory is reported in the Table 2, since the relative error on the frequency is always less than 2%. We also remark in the Fig. 6 that the oscillation amplitude is decreasing with time due to the numerical dissipation of the numerical scheme. However by using more refined grids, this numerical dissipation tends to zero and the oscillation amplitude is well maintained along the simulation. In the Fig. 7, the interface location, the pressure field and the velocity field have been plotted in order to highlight the suitable behavior of the numerical solver on this specific situation involving an incompressible flow. In the droplet, we notice that the pressure field directly depends on the interface curvature.

4.3.2. Interaction of a bubble with a spherical wavefront

We now investigate the ability of the compressible projection method to compute multidimensional bubble oscillations. The surface tension is now considered, and it tends to keep the bubble spherical. The initial bubble radius is $R_0 = 10^{-5}$ m. Spherical boundary conditions are imposed on the boundary of the computational field in order to generate a spherical wavefront. At R_1, the wavefront is created by setting the pressure on the boundary: $p_S(t) = p_0 - Asin(wt + S/c)$, where $S = 4 \times 10^{-5}$ m, $p_0 = 10^5$ Pa, $A = 8 \times 10^4$ Pa, $w = 5 \times 10^5$ rad s^{-1}, $\sigma = 7 \times 10^{-2}$ Pa m and c is the sound speed in water computed with Eq. (9).

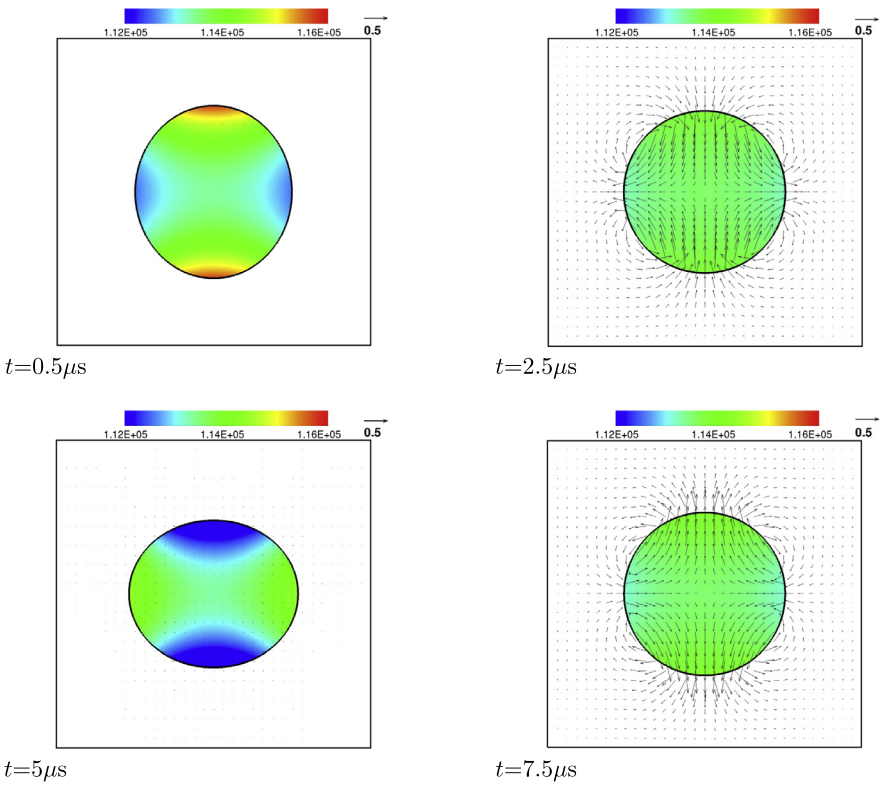


Fig. 7. Interface location, pressure (Pa) and velocity ($\mathrm{m\,s^{-1}}$) fields at different times. The grid contains 256×256 computational cells.

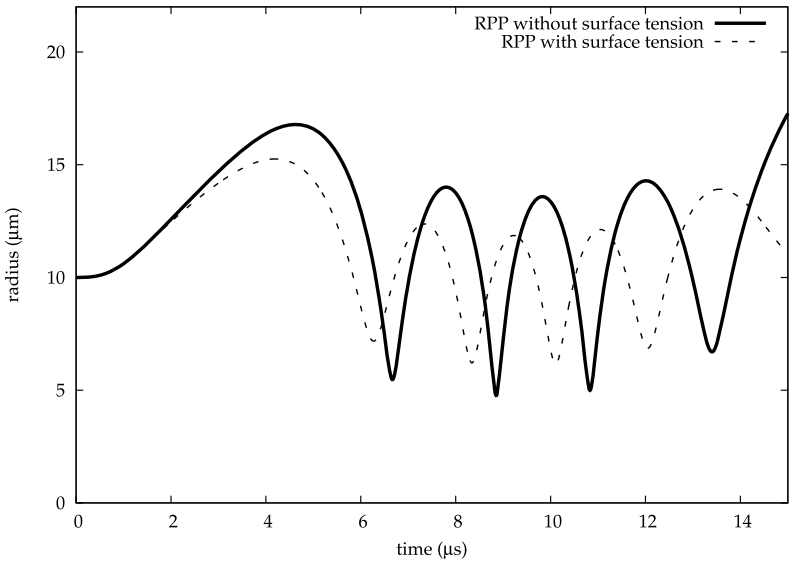


Fig. 8. Comparison between the solutions of the Rayleigh–Plesset–Prosperetti equation with and without surface tension effects.

Eqs. (69) and (70) show that the surface tension decreases the amplitude of the bubble volume. This observation is verified in Fig. 8. These effects are significant for micro-bubbles. Indeed, the pressure jump across the interface is inversely proportional to the bubble radius. For this test, with $\sigma = 7 \times 10^{-2}$ Pa m the pressure jump is $\frac{2\sigma}{R_0} = 14\,000$ Pa which cannot be neglected compared to the magnitude of the pressure p_0.

The computations are performed on an axisymmetric Cartesian mesh. As before, a symmetry is imposed on the radial plane. The sizes of the computational field are $lr = 4 \times 10^{-5}$ m and $lz = 4 \times 10^{-5}$ m. In a first time, we have attempted to perform this test case by using the basic projection method described in the Section 3.3.1. We have observed this method

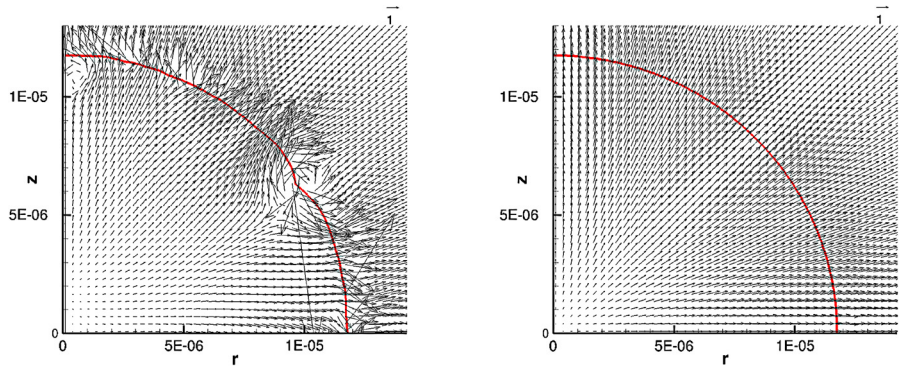


Fig. 9. Comparison between the basic projection method (left) and the time-splitting compressible projection method for surface tension (right). The grids contain 256×256 computational cells. Thanks to splitting projection method, spurious currents disappear.

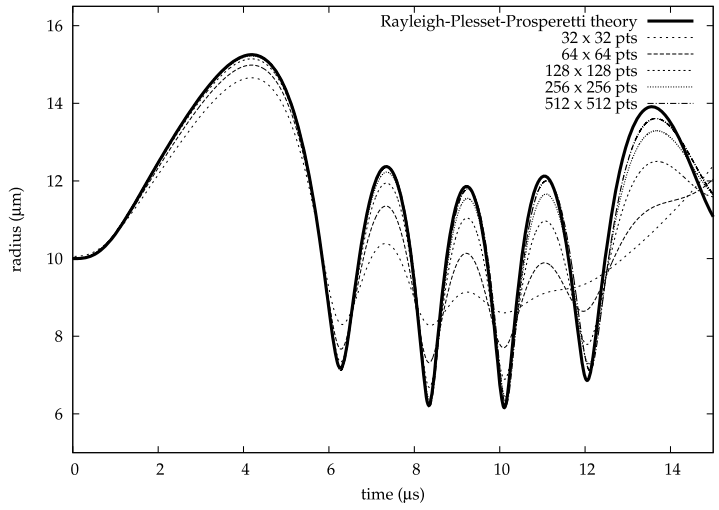


Fig. 10. Temporal evolution of the bubble radius. Comparison between the computation and the reference solution for different grids.

was subject to numerical instabilities if the surface tension effect is taken into account. See for example the snapshots of velocity field in the Fig. 9, at time $t = 1.55 \times 10^{-6}$ s with a mesh containing 256×256 cells, where very strong spurious currents have appeared on the interface. Moreover, we have observed that with the basic projection method, the finer is the mesh, the stronger are the spurious currents. This leads to a quite unstable method which requires some improvements. This observation has motivated us to design the new time splitting method which is presented in Section 3.3.2 to overcome this difficulty. For more details on these misleading behaviors, refer to Appendix A.

In Fig. 9, a snapshot of the velocity field has also been plotted with the time-splitting compressible projection method for surface tension. If this method is used, a clean velocity field is obtained since the unstable spurious currents have vanished. Therefore, this time-splitting will be used in all the simulations which are presented in the rest of this paper. In particular, the spatial convergence on the bubble radius temporal evolution is studied in Fig. 10. It clearly demonstrates that the numerical algorithm converges to the reference solution. Whereas, the simulation, which is performed with the coarsest grid (only 16 grid cells in the initial radius), provides accurate result on the amplitude of the first oscillation, we observe that 128 computational cells in the initial bubble radius are required to compute accurately the amplitude of the fifth oscillation. It corresponds to 512×512 cells in the whole domain. In addition, the temporal convergence on the bubble radius temporal evolution is also studied in Fig. 11. It clearly shows the temporal convergence of the numerical algorithm, since the discrepancy between the theory and the simulation decreases when the time step decreases. With a 128×128 grid, dividing the time step (Eq. (68)) by 8 allows to obtain about the 256×256 grid accuracy without dividing the time step.

In the Table 3, the variation of the error on the maximum radius during the first oscillation is reported for several computational grids. We can observe a similar behavior as for the 1D test-case, since the spatial accuracy is first order for the coarsest grids and next it increases to a second order accuracy, which is compatible with the second order centered schemes which are used in several steps of the compressible projection method.

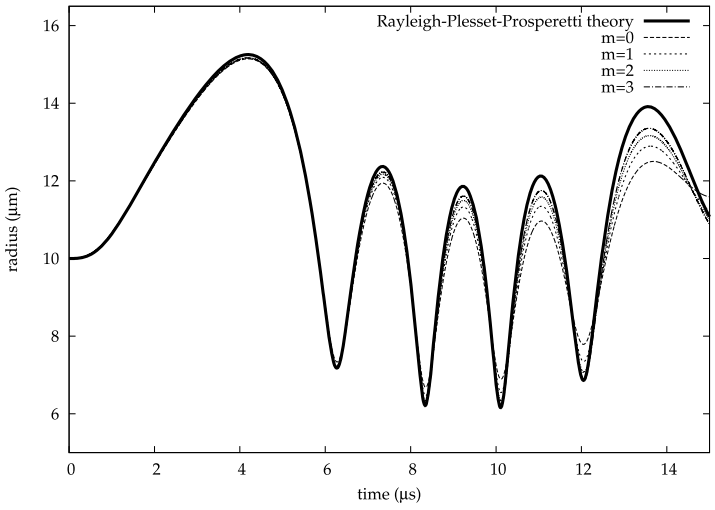


Fig. 11. Temporal evolution of the bubble radius. Comparison between the computation and the reference solution for different time steps. Computations are done with the same grid (128 × 128 cells). For each computations, $\Delta t_m = \Delta t \times (1/2)^m$, where Δt is the time-step calculated with Eq. (68).

Table 3
Error on the maximum radius during the first oscillation for several computational grids.

	Relative error	Order
32 × 32	11.31%	
64 × 64	5.10%	1.15
128 × 128	2.04%	1.32
256 × 256	0.61%	1.74
512 × 512	0.10%	2.68

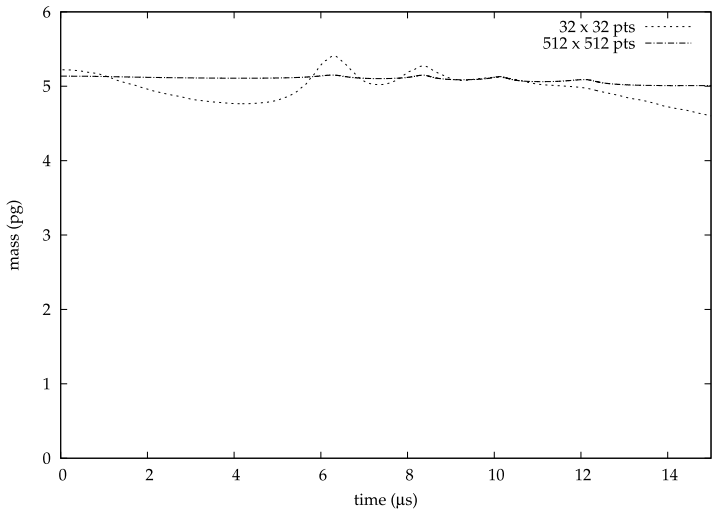


Fig. 12. Mass temporal evolution of the static bubble for different grids.

As it is well known that the Level-Set method is not a mass-conserving method, we have performed further investigations in order to quantify this lack of mass conservation in our simulations. The results are summarized in Fig. 12 which clearly points out that the mass conservation is not strictly imposed, but the method converges toward a mass-conserving solution.

According to the second principle of thermodynamics, the entropy is also a conservative variable in the situation which is considered here. It has been checked that property was satisfied by plotting the temporal evolution of the bubble entropy. In Fig. 13, we can observe that conservation property is perfectly preserved when the grid cells are sufficiently refined.

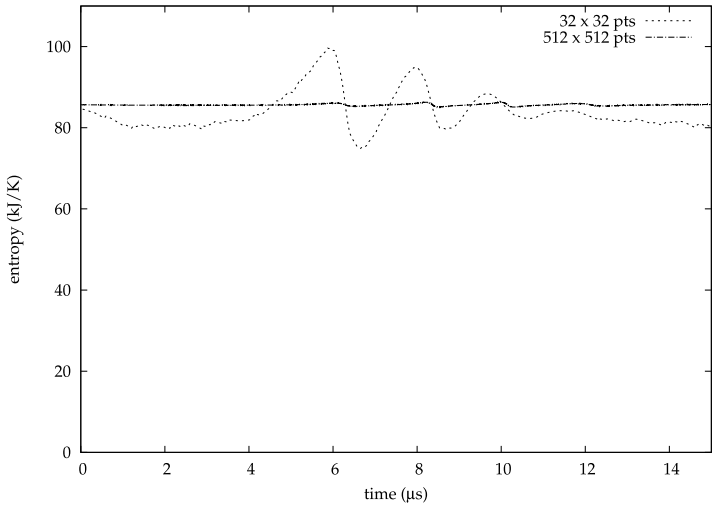


Fig. 13. Entropy temporal evolution of the static bubble for different grids.

Finally, the shape of the interface, the velocity field and the pressure fields are represented in Fig. 14 for different times. We observe the expansion and the compression of the bubble evolving with time. The velocity field is free from parasitic currents on the vicinity of the interface, and the pressure jump due to the surface tension is correctly captured.

To summarize, we have presented results for the interaction of a bubble with a spherical wave-front which allow a strong validation of the new time splitting compressible projection method for surface tension in a multidimensional configuration. Once this milestone has been addressed, simulations involving more complex configurations can be performed.

4.3.3. Interaction of a bubble with a plane wave-front

Low shape oscillation of a bubble We now study the interaction of a bubble and a plane wave-front, which is a more realistic situation than with a spherical wave-front. In the previous test-case, the symmetries induce that the bubble remains spherical and static. If a plane wave-front is considered, as the spherical symmetry is broken, the bubble can displace and its shape can evolve. Let us notice however, that an axial symmetry is still preserved, which allows performing an axisymmetric simulation instead of a more costly 3D computation. The initial bubble radius is still $R_0 = 10^{-5}$ m. At the top of the domain, the wavefront is generated by setting the pressure at $p(t) = p_0 - Asin(wt)$, where $p_0 = 10^5$ Pa, $A = 6 \times 10^4$ Pa and $w = 5 \times 10^5$ rad s^{-1}.

The amplitude of the wave-front is lower than in the previous test case to prevent from the bubble break-up. The sizes of the computational field are $lr = 8 \times 10^{-5}$ m and $lz = 16 \times 10^{-5}$ m. This simulation has been carried out with 4 three different grids, corresponding to a number of elements 64×128, 128×256, 256×512, 512×1024. Similarly as in the previous test case, the evolution of the equivalent radius is plotted in Fig. 15. The Rayleigh–Plesset–Prosperetti solution is no longer valid for this test-case test, but it is still marked as an indication. In particular, as it can be observed in Fig. 15, this simulation is still in accordance with the theory during the initial step of the computation. That can be explained by considering both that the wavelength $\lambda = 18.85 \times 10^{-3}$ m is much more important than the bubble radius and because the transmission coefficient of the ultrasound wave-front across the interface is almost zero. Indeed, it can be visualized on the snapshots of the pressure field in Fig. 16 that the wave wraps around the bubble, the pressure contours adopting an almost spherical shape. In Fig. 17, the average vertical velocity of the bubble has been plotted in order to bring out that the coupling between the bubble oscillation and its displacement is effective. In particular, it clearly appears that the velocity oscillations are strongly correlated with volume oscillations. The convergence study on the average vertical velocity is satisfactory since the two more refined grids provide very close results.

As in the previous section, we have also checked on these computations that the mass conservation and the entropy conservation were preserved if the grid cells are sufficiently refined (see respectively Fig. 18 and Fig. 19). Finally, the interface position, the velocity field and the pressure field have been plotted in the Fig. 16 for different times. These figures allow visualizing the temporal evolution of the bubble shape and its displacement. It also brings out that the pressure jump condition is correctly computed and that the velocity field is free from parasitic currents.

High shape oscillation of a bubble In the previous section, the numerical simulations have involved a coupling between the volume oscillations and the translation of the bubble. However, as it can be visualized in the Fig. 16, the bubble undergoes weak shape deformations. We present in this section simulations where the amplitude of the wave front has been increased in order to induce strong deformations of the bubble shape. The parameters of the simulations are the same as in the former section, except the amplitude of the wave front which is equal to $A = 7 \times 10^4$ Pa. In the Fig. 20 and Fig. 21, the pressure field, the velocity field and the interface location have been plotted for different times. These snapshots allow

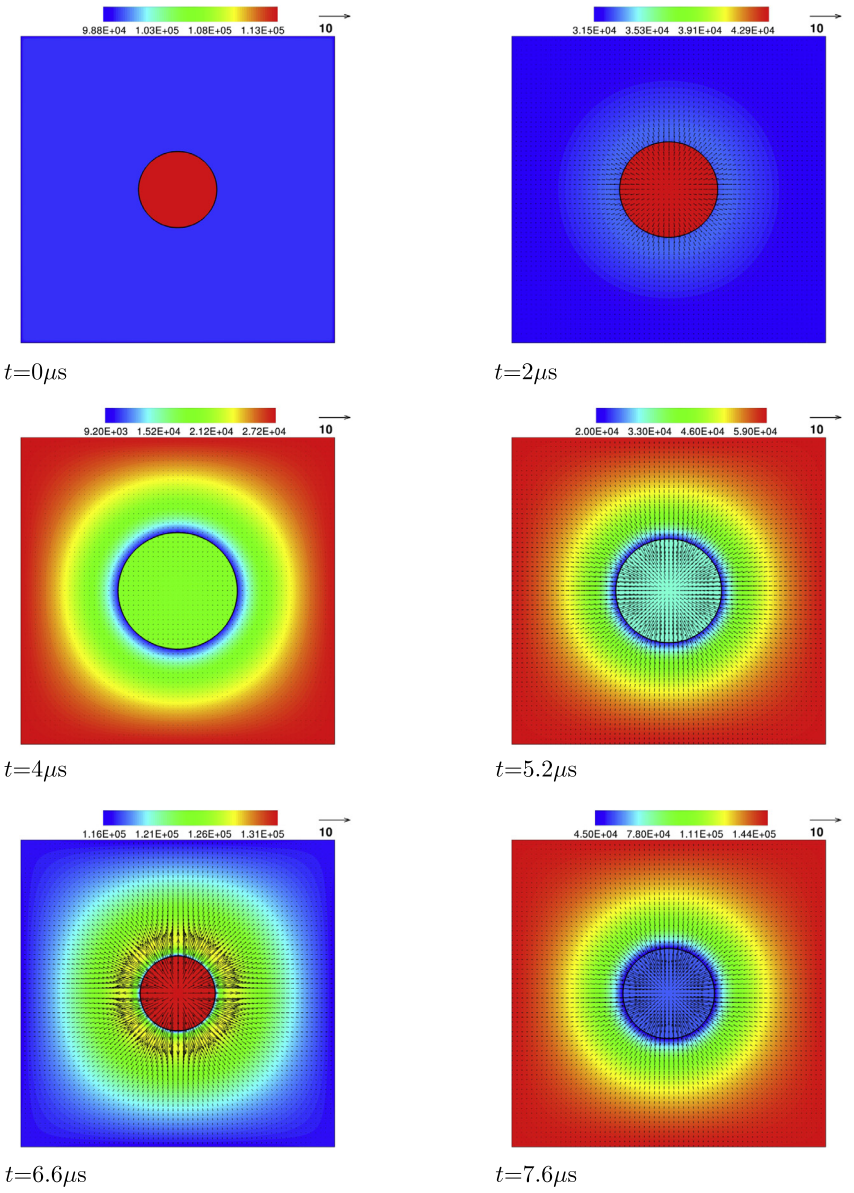


Fig. 14. Interface location, pressure (Pa) and velocity ($\mathrm{m\,s^{-1}}$) fields at different times.

strengthening the ability of the time splitting projection method for compressible flows to perform an efficient coupling between strong interface deformation and volume oscillation. In the Fig. 22 and in the Fig. 23, the temporal evolution of the equivalent radius and the average vertical velocity of the bubble are plotted for different grids in order to check the spatial convergence of the overall algorithm. Although the simulation performed on the thinnest grid seems to be well converged in the first time of the simulation, more important deviations appear in the end of the simulations. It suggests that a more refined grid could be used to improve the numerical results. However, we must also notice that our numerical model is based on the Euler equations which do not include the viscous dissipation effects. As a rotational flow and a deformed bubble are considered in this situation, the computations could not converge since the model does not account for any spatial cut-off due to the low scale viscous dissipation. Indeed, in such a situation the Reynolds number is infinite due to the zero viscosity in the two phases. Therefore, hydrodynamic instabilities in the wake of the bubble or shape instabilities can develop in a way which could depend on the mesh size. This is why the spatial convergence can sometimes be difficult to achieve for infinite Reynolds number simulations involving rotational flows. Moreover, as stronger deformations of the bubble are involved in this case than in the previous section, we can assume that a more refined grid could be required to obtain a better spatial convergence in this situation.

Finally, the Fig. 24 and the Fig. 25 demonstrate that the mass conservation and the entropy conservation are still suitably respected if the grid is refined.

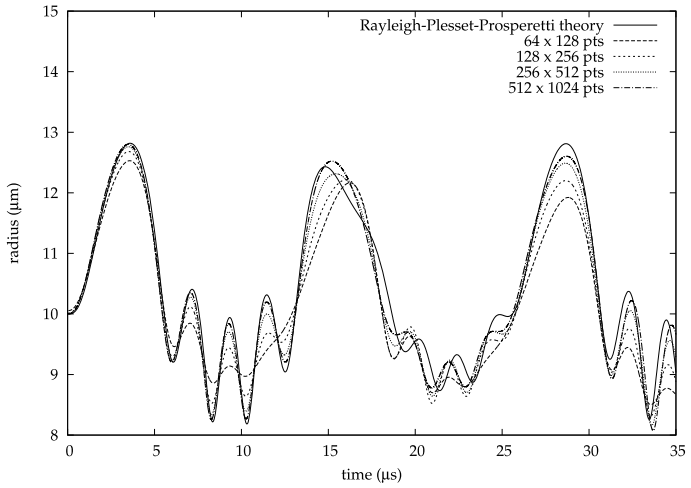


Fig. 15. Temporal evolution of the bubble radius for different grids.

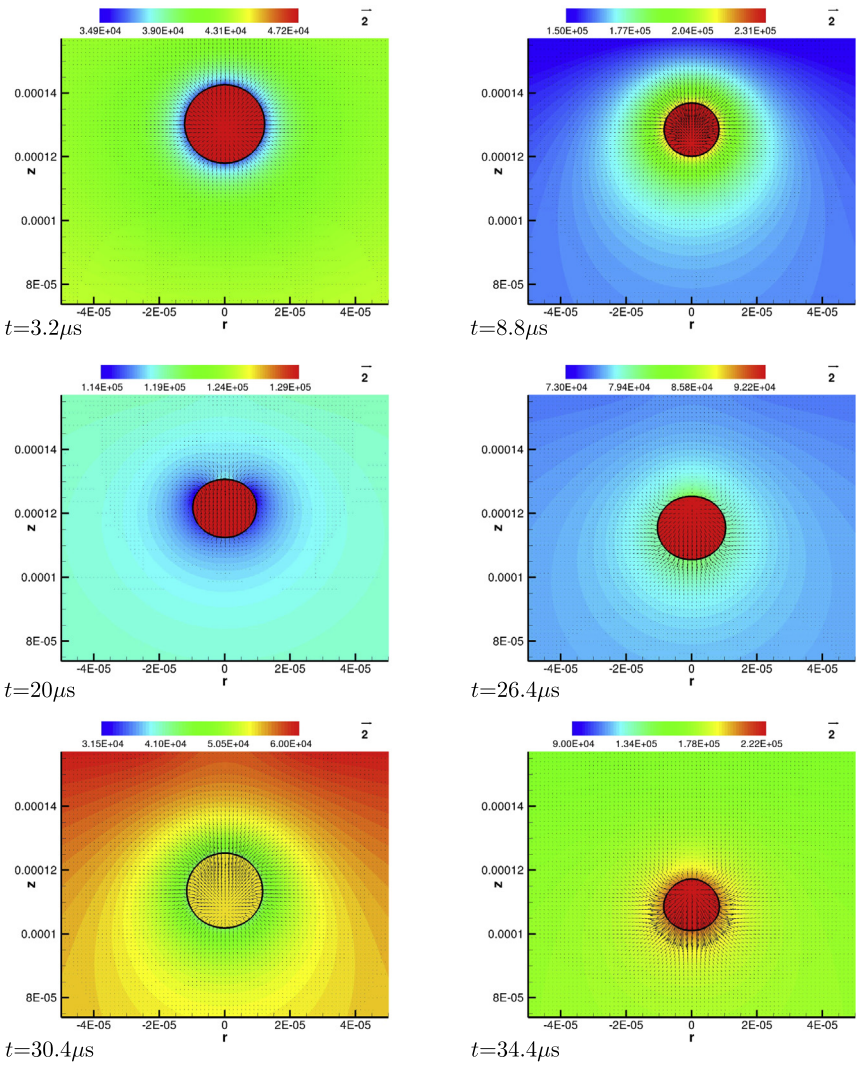


Fig. 16. Interface location, pressure (Pa) and velocity ($\mathrm{m\,s^{-1}}$) fields at different times.

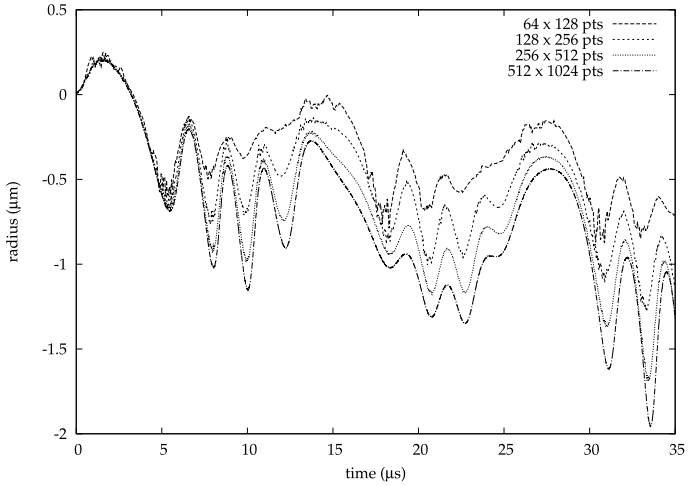


Fig. 17. Temporal evolution of the bubble average velocity for different grids.

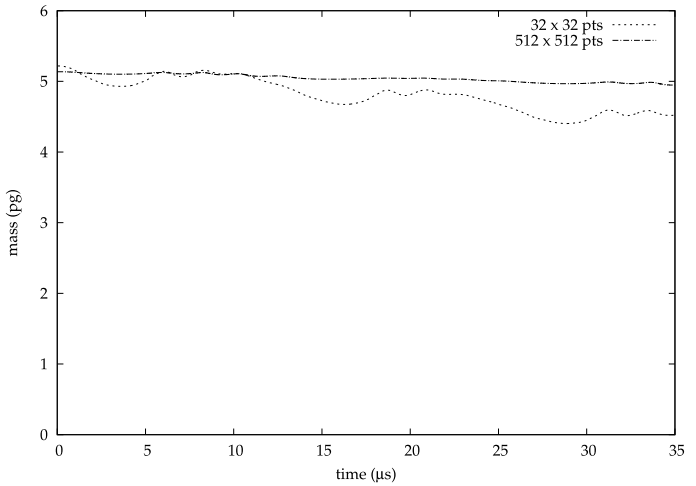


Fig. 18. Mass temporal evolution of the moving bubble for different grids.

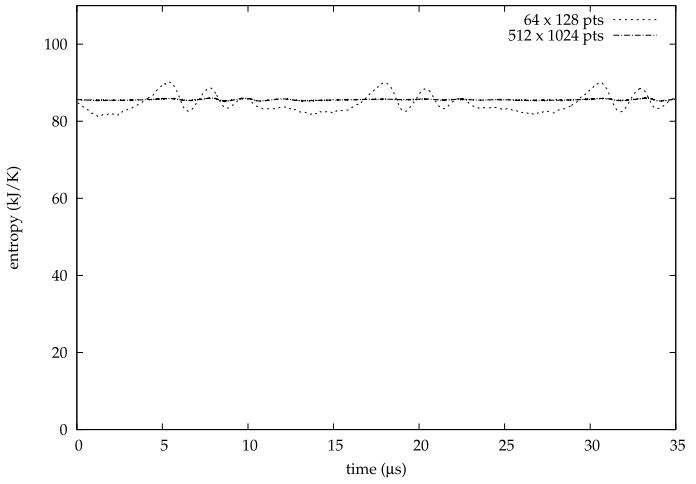
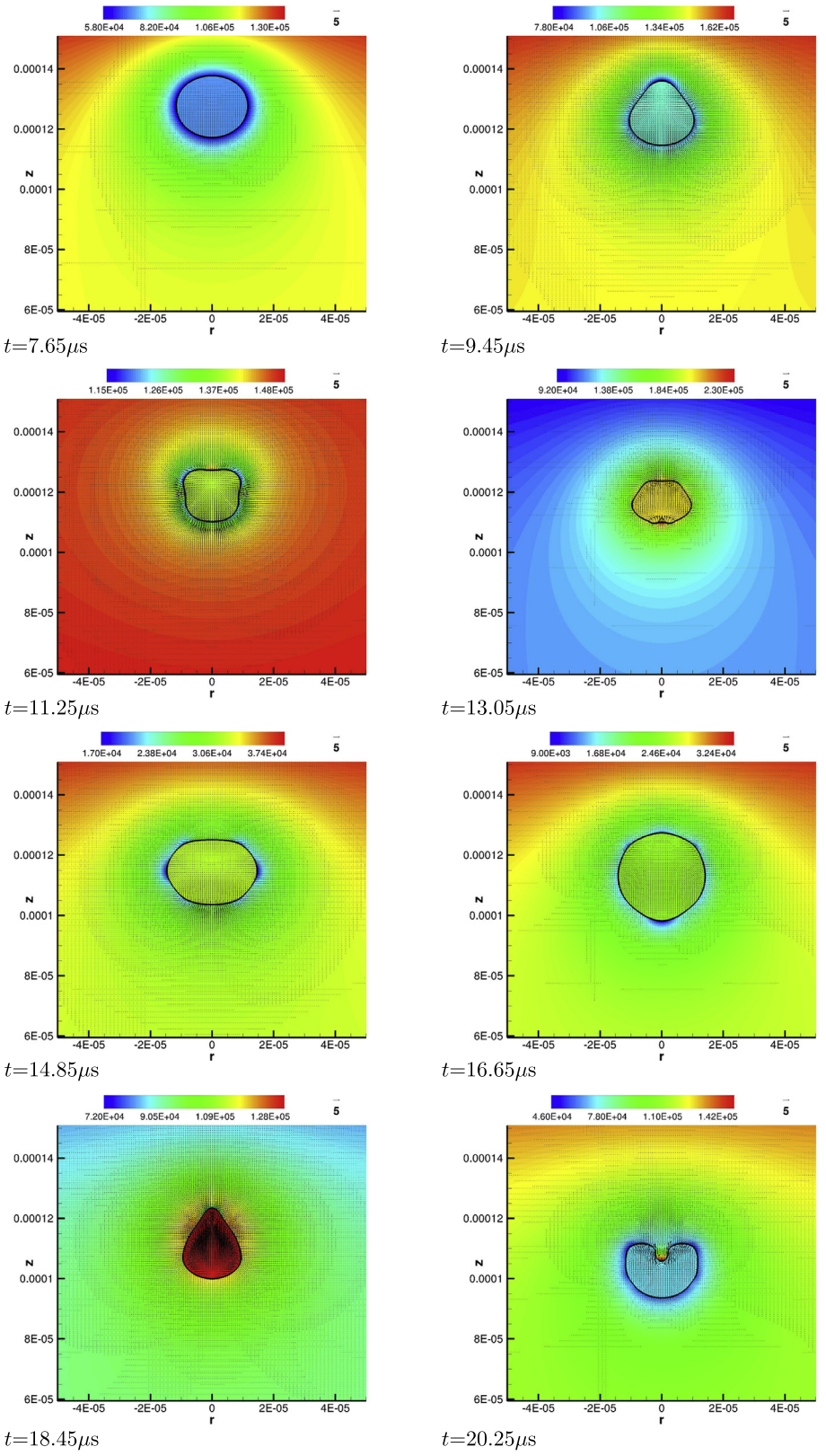


Fig. 19. Entropy temporal evolution of the moving bubble for different grids.



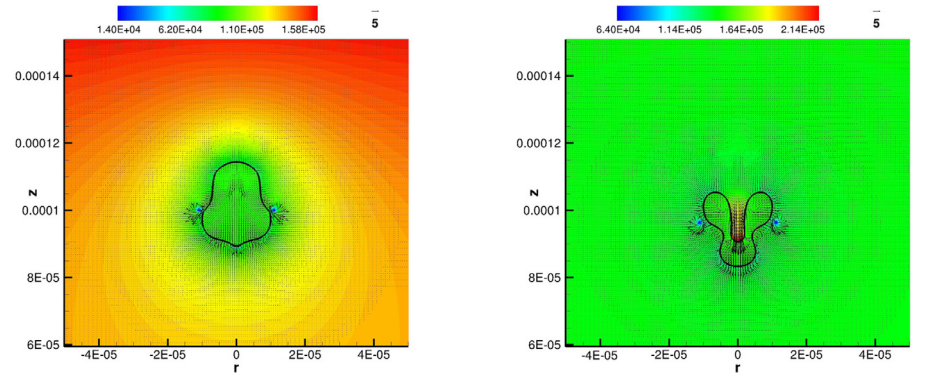
t=7.65μs t=9.45μs

t=11.25μs t=13.05μs

t=14.85μs t=16.65μs

t=18.45μs t=20.25μs

Fig. 20. Interface location, pressure (Pa) and velocity ($\mathrm{m\,s^{-1}}$) fields at different times.



t=22.05μs t=23.85μs

Fig. 20. *(continued)*

5. Conclusion

In this paper, numerical simulations of the interaction between an ultrasound wavefront and an air bubble surrounded by liquid water have been performed.

In a first step, several numerical methods have been assessed by using a one dimensional benchmark based on a reference solution computed from the Rayleigh–Plesset–Prosperetti equation. As a low Mach number flow is considered, the compressible projection method clearly appears as the most suitable one.

In a second step, this method has been developed both in a multidimensional framework and with surface tension effects. Our first attempt was a simple implementation of the surface tension terms which was instigated from the usual approach for an incompressible two-phase flow. We report that this simple implementation is affected by strong parasitic currents and numerical instabilities. Therefore, a novel velocity/pressure splitting method is proposed to overcome this issue. The multidimensional numerical simulations clearly highlight that by applying this velocity/pressure splitting method, the spurious currents are strongly reduced. Moreover, the spatial convergence of this new algorithm toward a physically consistent solution is successfully investigated. Thus, the velocity/pressure splitting for the compressible projection method proves to be an efficient numerical tool to compute the interaction of acoustic or ultrasound waves with a liquid–gas interface for low Mach number flows.

This work opens many new perspectives, but other developments are required to study accurately the mechanism of shock formation during the bubble collapse. Previous works, in [21,53,55], have ever demonstrated that the compressible projection method can be an efficient method to compute the formation and the propagation of shock waves for a one phase flow, provided that the spatial discretization is applied to a conservative formulation of the fundamental equations. It is noteworthy that we use in this work a primitive formulation of these equations. Therefore, as it stands, this numerical method is not able to compute accurately supersonic flows. If a conservative formulation is used in a two-phase flow approach, computing the convective derivatives is a more complex task. Indeed, the density jump condition must be accounted accurately if the interface crosses the stencil of the numerical scheme which is used. Therefore, future investigations will be dedicated to the extension of this new numerical solver to all Mach number two-phase flows. A more complete model should also include dissipative effects as the viscous friction and the heat conduction.

Acknowledgement

This work was supported by the French National Research Agency in the framework of the LabEx CeLyA (ANR-10-LABX-60) and the program SonInCaRe (Sonoporation In vivo by Cavitation Regulation, ANR-10-TECS-0003).

Appendix A. Comparison on the spurious currents between the basic projection method and the time splitting projection method

We provide here some evidences on the misleading behavior of the basic implementation of the surface tension in the framework of the compressible projection method. Some qualitative elements have already been mentioned in the paper, as for instance in the Fig. 9(a).

In this appendix, we provide more quantitative elements on this specific point. In order to demonstrate more precisely the failure of the method, we present here more detailed results with this method. The configuration of the test case is described in Section 4.3.2. We use a coarse mesh of 32×32 computation cells. For each computation, the time step is $\Delta t_m = \Delta t \times (1/2)^m$ where Δt is the time step calculated with Eq. (68). The temporal convergence of the basic projection method is investigated by plotting the temporal evolution of the bubble mass in the Fig. 26. It shows that this method does not converge since the mass loss increases when the time step decreases.

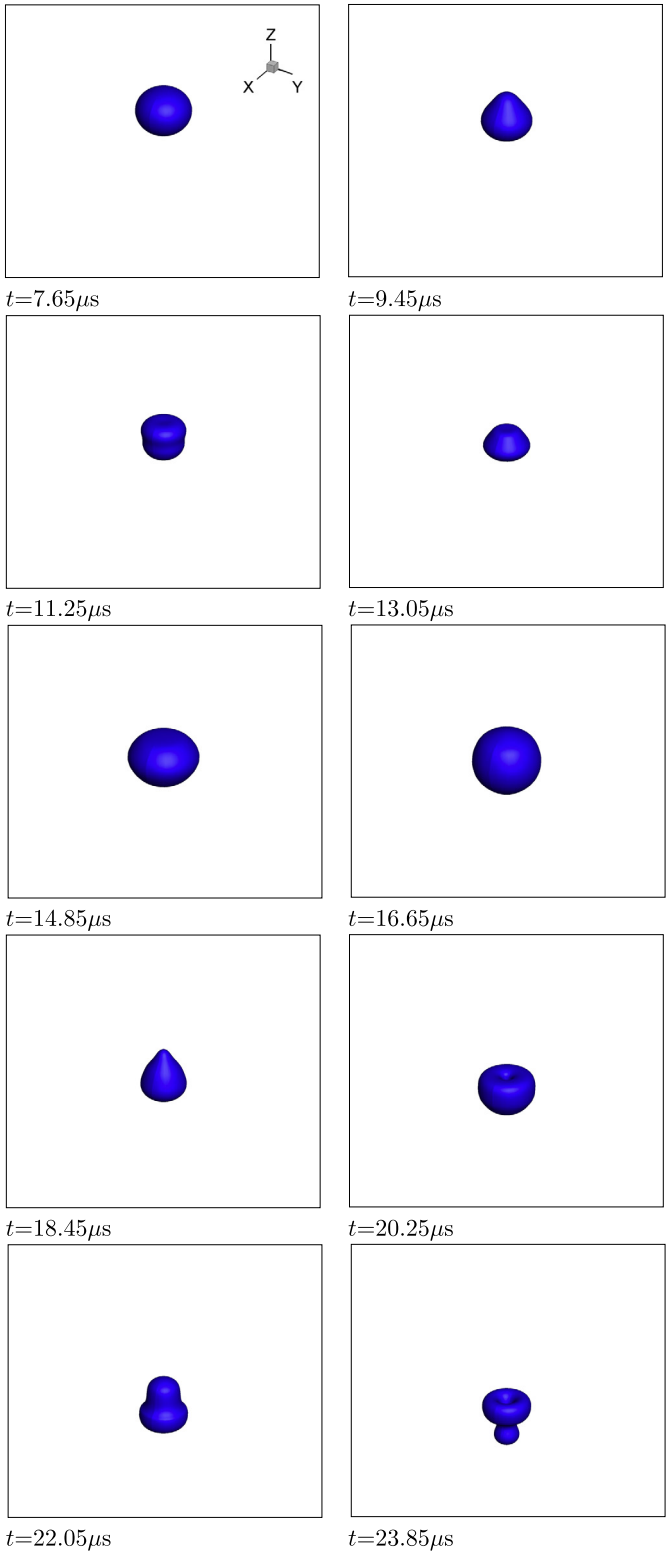


Fig. 21. Interface location (3D) at different times.

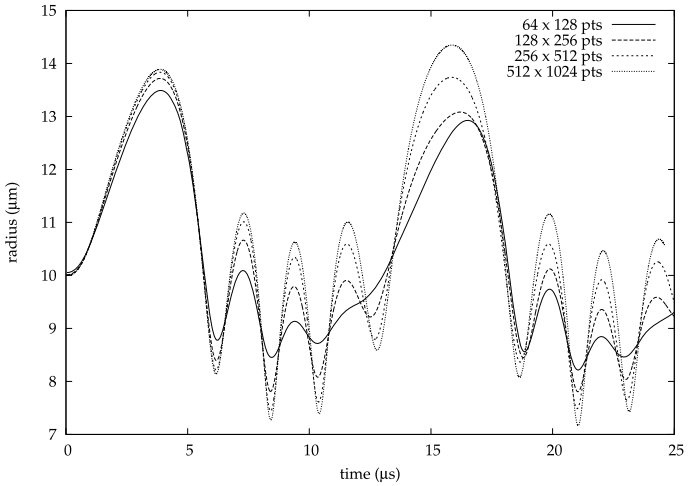


Fig. 22. Temporal evolution of the bubble radius for different grids.

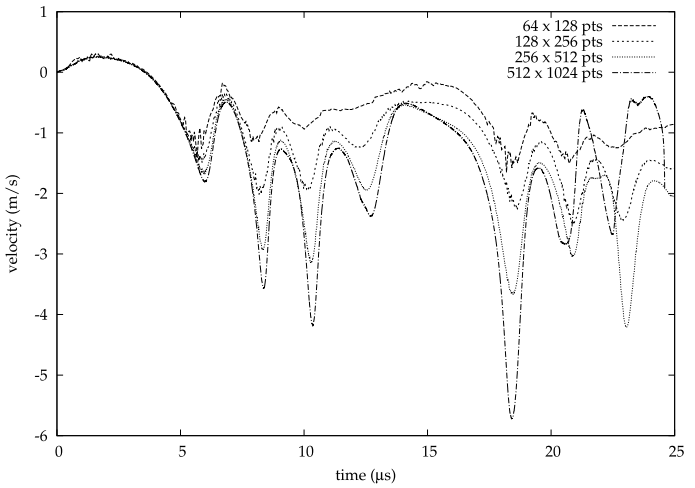


Fig. 23. Temporal evolution of the bubble average velocity for different grids.

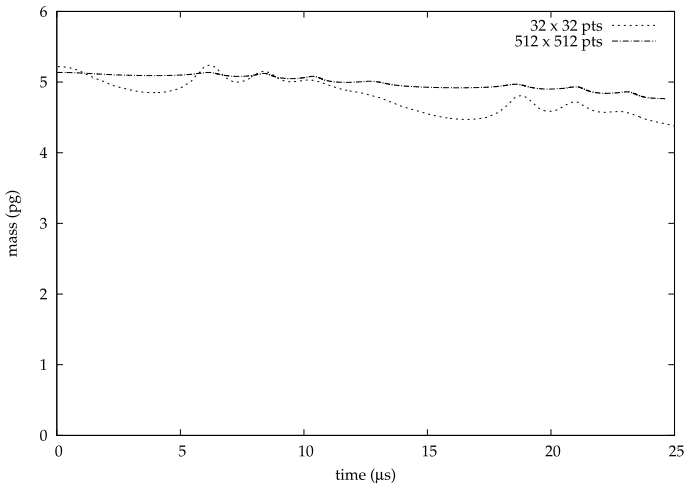


Fig. 24. Mass temporal evolution of the moving bubble for different grids.

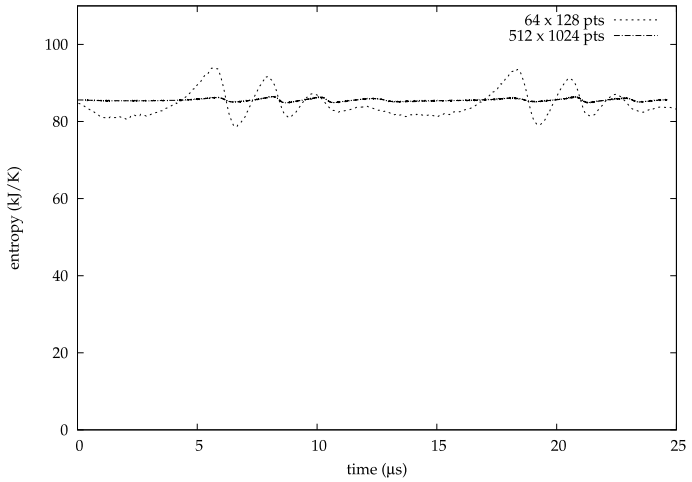


Fig. 25. Entropy temporal evolution of the moving bubble for different grids.

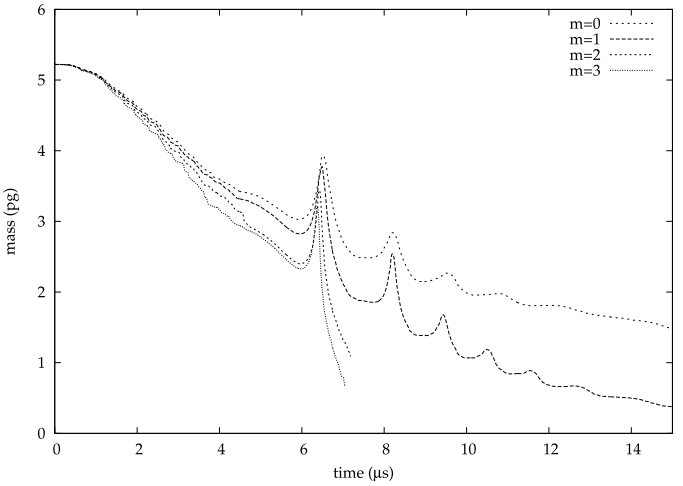


Fig. 26. Temporal evolution of the bubble mass. For each computation, the time step is $\Delta t_m = \Delta t \times (1/2)^m$.

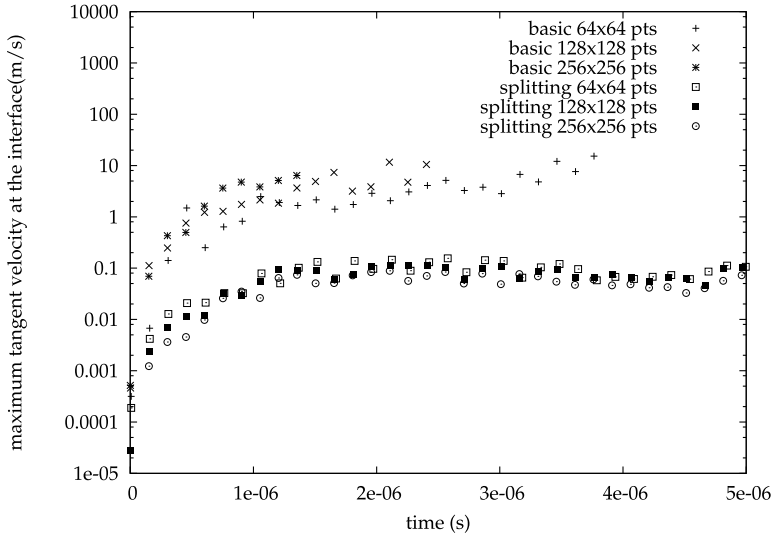


Fig. 27. Evolution of the maximum tangent velocity at the interface for the first oscillation.

This mass decrease is linked to spurious currents. We have plotted in the Fig. 27 the temporal evolution of the maximum tangential velocity in the vicinity of the interface for a volume oscillation test-case. Indeed, as the theoretical solution assumes a radial flow the tangential velocity should be equal to zero in this situation. The Fig. 27 clearly shows that the spurious currents are much lower with the time splitting projection method than with the basic projection method for different computational grids. Moreover, we can observe in this figure that the amplitude of the spurious currents decreases if the computational grid is refined for the time splitting projection method, whereas it increases with the spatial resolution if the basic projection method is used. It confirms the misleading behavior of the latter and the relevance of the time splitting projection for this kind of simulations. Finally, if the basic projection method is used we have observed that the parasitic currents are so important that they can lead to the divergence of the simulation, especially if refined grids are used.

References

[1] T.D. Aslam, A partial differential equation approach to multidimensional extrapolation, J. Comput. Phys. 193 (1) (2004) 349–355.
[2] H. Bijl, P. Wesseling, A unified method for computing incompressible and compressible flows in boundary-fitted coordinates, J. Comput. Phys. 141 (2) (1998) 153–173.
[3] R. Caiden, R.P. Fedkiw, C. Anderson, A numerical method for two-phase flow consisting of separate compressible and incompressible regions, J. Comput. Phys. 166 (1) (2001) 1–27.
[4] J.P. Caltagirone, S. Vincent, C. Caruyer, A multiphase compressible model for the simulation of multiphase flows, Comput. Fluids 50 (1) (2011) 24–34.
[5] C.H. Chang, X. Deng, T.G. Theofanous, Direct numerical simulation of interfacial instabilities: a consistent, conservative, all-speed, sharp-interface method, J. Comput. Phys. 242 (2013) 946–990.
[6] J. Dendy, Black box multigrid, J. Comput. Phys. 48 (3) (1982) 366–386.
[7] B. Einfeldt, On Godunov-type methods for gas dynamics, SIAM J. Numer. Anal. 25 (2) (1988) 294–318.
[8] 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) 457–492.
[9] R. Fedkiw, A. Marquina, B. Merriman, An isobaric fix for the overheating problem in multimaterial compressible flows, J. Comput. Phys. 148 (2) (1999) 545–578.
[10] D.E. Fyfe, E.S. Oran, M.J. Fritts, Surface tension and viscosity with lagrangian hydrodynamics on a triangular mesh, J. Comput. Phys. 76 (2) (1988) 349–384.
[11] F. Gibou, L. Chen, D. Nguyen, S. Banerjee, A level set based sharp interface method for the multiphase incompressible Navier–Stokes equations with phase change, J. Comput. Phys. 222 (2) (2007) 536–555.
[12] F. Gibou, C. Min, R. Fedkiw, High resolution sharp computational methods for elliptic and parabolic problems in complex geometries, J. Sci. Comput. 54 (2–3) (2013) 369–413.
[13] N. Grenier, J.P. Vila, P. Villedieu, An accurate low-Mach scheme for a compressible two-fluid model applied to free-surface flows, J. Comput. Phys. 252 (2013) 1–19.
[14] H. Guillard, C. Viozat, On the behaviour of upwind schemes in the low Mach number limit, Comput. Fluids 28 (1999) 63–86.
[15] G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes, J. Comput. Phys. 126 (1) (1996) 202–228.
[16] E. Johnsen, T. Colonius, Numerical simulations of non-spherical bubble collapse, J. Fluid Mech. 629 (2009) 231–262.
[17] S.Y. Kadioglu, M. Sussman, Adaptive solution techniques for simulating underwater explosions and implosions, J. Comput. Phys. 227 (2008) 2083–2104.
[18] S.Y. Kadioglu, M. Sussman, S. Osher, J.P. Wright, M. Kang, A second order primitive preconditioner for solving all speed multi-phase flows, J. Comput. Phys. 209 (2005) 477–503.
[19] M. Kang, R. Fedkiw, X.D. Liu, A boundary condition capturing method for multiphase incompressible flow, J. Sci. Comput. 15 (3) (2000) 323–360.
[20] J.B. Keller, M. Miksis, Bubble oscillations of large amplitude, J. Acoust. Soc. Am. 68 (1980) 628–633.
[21] N. Kwatra, J. Su, J.T. Grétarson, R. Fedkiw, A method for avoiding the acoustic time step restriction in compressible flow, J. Comput. Phys. 228 (2009) 4146–4161.
[22] B. Lalanne, S. Tanguy, F. Risso, Effect of rising motion on the damped shape oscillations of drops and bubbles, Phys. Fluids (1994–present) 25 (11) (2013) 112107.
[23] B. Lalanne, S. Tanguy, L. Rueda Villegas, F. Risso, On the computation of viscous terms for incompressible two-phase flows with level-set/ghost fluid method, J. Comput. Phys. (2015), http://dx.doi.org/10.1016/j.jcp.2015.08.036, submitted.
[24] H. Lamb, Hydrodynamics, Cambridge University Press, 1932.
[25] G. Layes, O. Le Métayer, Quantitative numerical and experimental studies of the shock accelerated heteregeneous bubbles motion, Phys. Fluids 19 (2007) 042105.
[26] T.G. Leighton, Derivation of the Rayleigh–Plesset equation in terms of volume, ISVR Technical report N 308, 2007.
[27] S. Lemartelot, B. Nkonga, R. Saurel, Liquid and liquid–gas at all speeds, J. Comput. Phys. 255 (2013) 53–82.
[28] I. Lentacker, I. De Cock, R. Deckers, S.C. De Smedt, C.T.W. Moonen, Understanding ultrasound induced sonoporation: definitions and underlying mechanisms, Adv. Drug Deliv. Rev. 72 (2014) 49–64.
[29] R.J. Leveque, Z. Li, Immersed interface methods for Stokes flow with elastic boundaries or surface tension, SIAM J. Sci. Comput. 18 (3) (1997) 709–735.
[30] X.D. Liu, R. Fedkiw, M. Kang, A boundary condition capturing method for Poisson's equation on irregular domains, J. Comput. Phys. 160 (2000) 151–178.
[31] S.P. MacLachlan, J.M. Tang, C. Vuik, Fast and robust solvers for pressure-correction in bubbly flow problems, J. Comput. Phys. 227 (23) (2008) 9742–9761.
[32] Y. Matsumoto, J.S. Allen, S. Yoshizawa, T. Ikeda, Y. Kaneko, Medical ultrasound with microbubbles, Exp. Therm. Fluid Sci. 29 (3) (2005) 255–265.
[33] D.L. Miller, S.V. Pislaru, J.F. Greenleaf, Sonoporation: mechanical DNA delivery by ultrasonic cavitation, Somat. Cell Mol. Genet. 27 (1–6) (2002) 115–134.
[34] D.Q. Nguyen, R.P. Fedkiw, M. Kang, A boundary condition capturing method for incompressible flame discontinuities, J. Comput. Phys. 172 (1) (2001) 71–98.
[35] R.R. Nourgaliev, T.N. Dinh, T.G. Theofanous, Adaptive characteristics-based matching for compressible multifluid dynamics, J. Comput. Phys. 213 (2) (2006) 500–529.
[36] S. Osher, J.A. Sethian, Fronts propagating with curvature-dependent speed: algorithms based on Hamilton–Jacobi formulations, J. Comput. Phys. 79 (1988) 12–49.
[37] G. Pianet, S. Vincent, J. Leboi, J.P. Caltagirone, M. Anderhuber, Simulating compressible gas bubbles with a smooth volume tracking 1-fluid method, Int. J. Multiph. Flow 36 (4) (2010) 273–283.
[38] S. Popinet, S. Zaleski, Bubble collapse near a solid boundary: a numerical study of the influence of viscosity, J. Fluids Mech. 464 (2002) 137–163.
[39] R.J. Price, D.M. Skyba, S. Kaul, T.C. Skalak, Delivery of colloidal particles and red blood cells to tissue through microvessel ruptures created by targeted microbubble destruction with ultrasound, Circulation 98 (13) (1998) 1264–1267.

[40] P.L. Roe, Approximate Riemann solvers, parameter vectors, and difference schemes, J. Comput. Phys. 43 (2) (1981) 357–372.
[41] U. Rosenschein, V. Furman, E. Kerner, I. Fabian, J. Bernheim, Y. Eshel, Ultrasound imaging-guided noninvasive ultrasound thrombolysis preclinical results, Circulation 102 (2) (2000) 238–245.
[42] S.J. Shaw, P.D.M. Spelt, Shock emission from collapsing bubbles, J. Fluid Mech. 646 (2010) 363–373.
[43] M. Sussman, P. Smereka, S. Osher, A level set approach for computing solutions for incompressible two-phase flows, J. Comput. Phys. 114 (1994) 146–159.
[44] M. Sussman, K.M. Smith, M.Y. Hussaini, M. Ohta, R. Zhi-Wei, A sharp interface method for incompressible two-phase flows, J. Comput. Phys. 221 (2) (2007) 469–505.
[45] S. Tanguy, T. Ménard, A. Berlemont, A level set method for vaporizing two-phase flows, J. Comput. Phys. 221 (2) (2007) 837–853.
[46] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, C. Colin, Benchmarks and numerical methods for the simulation of boiling flows, J. Comput. Phys. 264 (2014) 1–22.
[47] H. Terashima, G. Tryggvason, A front-tracking/ghost-fluid method for fluid interfaces in compressible flows, J. Comput. Phys. 228 (11) (2009) 4012–4037.
[48] E. Toro, M. Spruce, W. Speares, Restoration of the contact surface in the HLL Riemann solver, Shock Waves 4 (1994) 25–34.
[49] E. Turkel, Preconditioned methods for solving the incompressible and low speed compressible equations, J. Comput. Phys. 8 (2) (1987) 277–298.
[50] D.R. Van der Heul, C. Vuik, P. Wesseling, A conservative pressure-correction method for flow at all speeds, Comput. Fluids 32 (8) (2003) 1113–1132.
[51] B.P. Van Poppel, O. Desjardins, J.W. Daily, A ghost fluid, level set methodology for simulating multiphase electrohydrodynamic flows with application to liquid fuel injection, J. Comput. Phys. 229 (20) (2010) 7977–7996.
[52] A. Van Wamel, K. Kooiman, M. Harteveld, M. Emmer, F.J. Ten Cate, M. Versluis, N. De Jong, Vibrating microbubbles poking individual cells: drug transfer into cells via sonoporation, J. Control. Release 112 (2) (2006) 149–155.
[53] F. Xiao, Unified formulation for compressible and incompressible flows by using multi-integrated moments I: one-dimensional inviscid compressible flow, J. Comput. Phys. (2004) 629–654.
[54] F. Xiao, R. Akoh, S. Ii, Unified formulation for compressible and incompressible flows by using multi-integrated moments II: multi-dimensional version for compressible and incompressible flows, J. Comput. Phys. 213 (1) (2006) 31–56.
[55] T. Yabe, P.Y. Wang, Unified numerical procedure for compressible and incompressible fluid, J. Phys. Soc. Jpn. 60 (1991) 2105–2108.
[56] T. Yabe, F. Xiao, T. Utsumi, The constrained interpolation profile method for multiphase analysis, J. Comput. Phys. 169 (2) (2001) 556–593.
[57] B. Yang, A. Prosperetti, A second-order boundary-fitted projection method for free-surface flow computations, J. Comput. Phys. 213 (2006) 574–590.
[58] H. Yu, S. Chen, A model to calculate microstreaming-shear stress generated by oscillating microbubbles on the cell membrane in sonoporation, Bio-Med. Mater. Eng. 24 (1) (2014) 861–868.