Volume 46, issue 1, July 2011
ISSN 0045-7930
ELSEVIER
An International Journal
computers & fluids
Editor-in-Chief:
MICHEL DEVILLE
Associate Editor:
JEAN-JACQUES CHATTOT
Special Issue
10th ICFD Conference Series on Numerical Methods for Fluid Dynamics
(ICFD 2010)
Guest Editor
Peter K. Sweby

Computers & Fluids 46 (2011) 59–69

Contents lists available at ScienceDirect

Computers & Fluids

journal homepage: www.elsevier.com/locate/compfluid

ReALE: A Reconnection Arbitrary-Lagrangian–Eulerian method in cylindrical geometry

Raphaël Loubère [a,*], Pierre-Henri Maire [b], Mikhail Shashkov [c]

[a] Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, France
[b] Commisariatá l'Energie Atomique CEA-CESTA, 33114 Le Barp, France
[c] Los Alamos National Laboratory, T-5, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 29 April 2010
Received in revised form 28 August 2010
Accepted 31 August 2010
Available online 21 September 2010

Keywords:
ReALE
Cylindrical geometry
Lagrangian hydrodynamics
Voronoi mesh
Arbitrary-Lagrangian–Eulerian
Mesh reconnection
Polygonal mesh

ABSTRACT

This paper deals with the extension to the cylindrical geometry of the recently introduced Reconnection algorithm for Arbitrary-Lagrangian–Eulerian (ReALE) framework. The main elements in standard ALE methods are an explicit Lagrangian phase, a rezoning phase, and a remapping phase. Usually the new mesh provided by the rezone phase is obtained by moving grid nodes without changing connectivity of the underlying mesh. Such rezone strategy has its limitation due to the fixed topology of the mesh. In ReALE we allow connectivity of the mesh to change in rezone phase, which leads to general polygonal mesh and permits to follow Lagrangian features much better than for standard ALE methods. Rezone strategy with reconnection is based on using Voronoi tesselation machinery. In this work we focus on the extension of each phase of ReALE to cylindrical geometry. The Lagrangian, rezone with reconnection and remap phases are revamped to take into account the cylindrical geometry. We demonstrate the efficiency of our ReALE in cylindrical geometry on series of numerical examples.

1. Introduction

A new reconnection-based Arbitrary-Lagrangian–Eulerian (ALE) framework called Reconnection ALE (ReALE) has been recently introduced in [1]. The main elements in standard ALE methods are an explicit Lagrangian phase, a rezoning phase, and a remapping phase. Usually the new mesh provided by the rezone phase is obtained by moving grid nodes without changing connectivity of the underlying mesh. Such rezone strategy has its limitation due to the fixed topology of the mesh and may lead to stagnation of the mesh in certain situations [1]. Contrarily to classical ALE framework, the rezone part of ReALE allows topological mesh reconnection using the machinery of Voronoi tesselation [2]. The new feature of this technique is an underlying set of generators moving with the fluid as "pseudo-Lagrangian particles". The new generator position is a combination between its Lagrangian new position and the displaced Lagrangian cell centroid. These generators, as particles, can change neighbors especially when shear or vortex motions occur. The Voronoi machinery is then used on this set of generators to define the rezone mesh: Each generator is associated to the same Voronoi cell which, accordingly, may have changed its neighborhood. This Voronoi mesh is the rezone mesh onto which the physical variables are further remapped. Consequently as the Lagrangian and rezone meshes are *a priori* different, the conservative remap phase must be modified to handle polygonal meshes possibly with different connectivity.

In [1] the 2D Cartesian geometry was only considered as to prove the feasibility of this ALE with reconnection approach. Contrarily in this work we investigate the extension of ReALE to cylindrical geometry. Although staggered and cell-centered Lagrangian schemes were considered in [1] to prove the generality of ReALE, in this work we focus on Lagrangian cell-centered discretization because the presentation and implementation are simpler. However there is no theoretical limitation in using a staggered placement of variable for ReALE in cylindrical geometry. High-order cell-centered discretization of the Lagrangian hydrodynamics equations has been described [3]; all conserved quantities, including momentum, and hence cell velocity are cell-centered. Extension to cylindrical geometry has also been studied in [4] in a control volume or area-weighted discretization. The control volume scheme conserves momentum, total energy and satisfies a local entropy inequality in its first-order semi-discrete form. The main difference between these approaches relies on the problem of preserving spherical symmetry in two-dimensional cylindrical geometry. Being given a one-dimensional spherical flow on a polar grid, equally spaced in angle, Maire [4] analyzed the ability of the schemes to maintain spherical symmetry. It turns out that the

[*] Corresponding author. Tel.: +33 561 557 652.
E-mail address: raphael.loubere@math.univ-toulouse.fr (R. Loubère).

0045-7930/$ - see front matter © 2010 Elsevier Ltd. All rights reserved.
doi:10.1016/j.compfluid.2010.08.024

control volume formulation does not preserve symmetry whereas the area-weighted formulation does similarly to staggered Lagrangian schemes [5]. In the context of ReALE the preservation of symmetric polar grid is not a goal as we are dealing with polygonal meshes by nature. We leave this issue for later investigation. However this cylindrical geometry extension of ReALE is motivated since in many application problems, such as inertial confinement problems, physical domains have axisymmetric features. This paper is organized as follows; we first recall some notion of cylindrical geometry, then in a second section we derive the cell-centered Lagrangian scheme. In the third section the rezone and remap parts are extended to cylindrical geometry. Numerical test case are provided in the fourth section where comparisons to exact solution and/or experimental solution are proposed. Finally conclusions and perspectives are drawn.

2. Cylindrical geometry

We are interested in discretizing the equations of the 2D Lagrangian hydrodynamics in cylindrical geometry, taking into account under the same form both Cartesian and cylindrical geometry. To this end, we re-use the notations introduced by Dukowicz in [6]. In the Lagrangian formalism the rates of change of mass, volume, momentum and total energy are computed assuming that the computational volumes follow the material motion. This representation leads to the following set of equations for an arbitrary moving control volume $V(t)$:

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)}\rho\,\mathrm{d}V = 0, \tag{1a}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)}\mathrm{d}V - \int_{V(t)}\nabla\cdot\boldsymbol{U}\,\mathrm{d}V = 0, \tag{1b}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)}\rho\boldsymbol{U}\,\mathrm{d}V + \int_{V(t)}\boldsymbol{\nabla}P\,\mathrm{d}V = \boldsymbol{0}, \tag{1c}$$

$$\frac{\mathrm{d}}{\mathrm{d}t}\int_{V(t)}\rho E\,\mathrm{d}V + \int_{V(t)}\nabla\cdot(P\boldsymbol{U})\,\mathrm{d}V = 0. \tag{1d}$$

where $\frac{\mathrm{d}}{\mathrm{d}t}$ denotes the material, or Lagrangian, time derivative. Here, ρ, $\boldsymbol{U}$, P, E respectively denote the mass density, velocity, pressure and specific total energy of the fluid. Eqs. 1a, 1c, 1d express the conservation of mass, momentum and total energy. The thermodynamic closure is obtained by adding the Equation Of State (EOS) of the form $P = P(\rho,\varepsilon)$, where the specific internal energy, ε, is related to the specific total energy by $\varepsilon = E - \frac{1}{2}\|\boldsymbol{U}\|^2$. We note that volume variation Eq. (1b) which is also named Geometric Conservation Law (GCL), is equivalent to the local kinematic equation

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X} = \boldsymbol{U}(\boldsymbol{X}(t),t), \quad \boldsymbol{X}(0) = \boldsymbol{x}, \tag{2}$$

where $\boldsymbol{X}$ is a point located on the control volume surface, $S(t)$, at time $t > 0$ and $\boldsymbol{x}$ corresponds to its initial position. We note that the case of Cartesian or cylindrical geometry can be combined by introducing the pseudo Cartesian frame (O,X,Y), equipped with the orthonormal basis $(\boldsymbol{e}_X,\boldsymbol{e}_Y)$, through the use of the pseudo radius $\mathscr{R}(Y) = 1-\alpha+\alpha Y$, where $\alpha = 1$ for cylindrical geometry and $\alpha = 0$ for Cartesian geometry. We remark that Y corresponds to the radial coordinate in the cylindrical case meaning that we assume rotational symmetry about X-axis, refer to Fig. 1. We note that if we refer to standard cylindrical coordinates, (Z,R), then X corresponds to Z and Y to R. In this framework, the volume V is obtained by rotating the area A about the X-axis. Thus, the volume element, $\mathrm{d}V$, writes $\mathrm{d}V = \mathscr{R}\,\mathrm{d}A$, where $\mathrm{d}A = \mathrm{d}X\mathrm{d}Y$ is the area element in the pseudo Cartesian coordinates. Note that we have omitted the factor 2π due to the integration in the azimuthal direction, namely we consider all integrated quantities to be defined per unit radian.

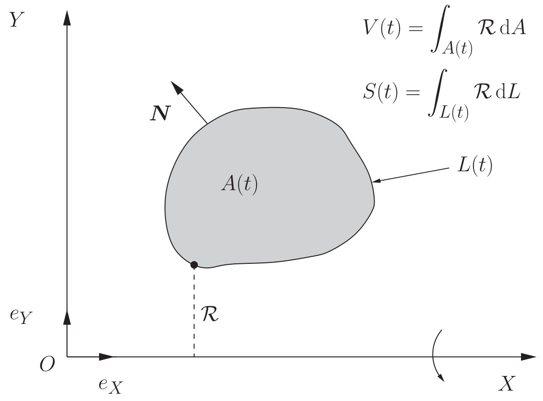
Y
$V(t) = \int_{A(t)} \mathcal{R}\,\mathrm{d}A$
$S(t) = \int_{L(t)} \mathcal{R}\,\mathrm{d}L$
N
A(t)
L(t)
e_Y
$\mathcal{R}$
O
e_X
X

Fig. 1. Notation related to the pseudo Cartesian geometry.

The surface S, which bounds the volume V, is obtained by rotating, L, the boundary of the area A, about the X-axis. Thus, the surface element, $\mathrm{d}S$, writes $\mathrm{d}S = \mathscr{R}\,\mathrm{d}L$, where $\mathrm{d}L$ is the line element along the perimeter of A.

In view of subsequent spatial discretization, we shall express the volume integrals associated with the divergence and gradient operators using the Green formula. We recall that, in the pseudo Cartesian frame, the divergence operator writes

$$\begin{aligned}\nabla\cdot\boldsymbol{U} &= \frac{\partial u}{\partial X} + \frac{1}{\mathscr{R}}\frac{\partial}{\partial Y}(\mathscr{R}v) = \frac{\partial u}{\partial X} + \frac{\partial v}{\partial Y} + \alpha\frac{v}{\mathscr{R}}\\ &= \frac{1}{\mathscr{R}}\left[\frac{\partial}{\partial X}(\mathscr{R}u) + \frac{\partial}{\partial Y}(\mathscr{R}v)\right]\end{aligned}$$

where (u,v) are the components of the vector $\boldsymbol{U}$. The gradient operator writes as usual

$$\boldsymbol{\nabla}P = \frac{\partial P}{\partial X}\boldsymbol{e}_X + \frac{\partial P}{\partial Y}\boldsymbol{e}_Y.$$

By replacing the volume integral form of the divergence operator by its surface integral form and by employing the previous notations one deduces the Green formula in the pseudo Cartesian framework as

$$\int_V \nabla\cdot\boldsymbol{U}\,\mathrm{d}V = \int_L \boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L. \tag{3}$$

where $\boldsymbol{N}$ is the unit outward normal associated with the contour L. To derive the surface integral form of the gradient operator, we use the vector identity $\boldsymbol{U}\cdot\boldsymbol{\nabla}P = \nabla\cdot(P\boldsymbol{U}) - P\nabla\cdot\boldsymbol{U}$, which holds for any vector $\boldsymbol{U}$. The integration of this identity over the volume V leads to

$$\int_V \boldsymbol{U}\cdot\boldsymbol{\nabla}P\,\mathrm{d}V = \int_L P\boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L - \int_A P\nabla\cdot\boldsymbol{U}\mathscr{R}\,\mathrm{d}A.$$

Assuming a constant $\boldsymbol{U}$ vector, we finally get

$$\int_V \boldsymbol{\nabla}P\,\mathrm{d}V = \int_L P\boldsymbol{N}\mathscr{R}\,\mathrm{d}L - \alpha\boldsymbol{e}_Y\int_A P\,\mathrm{d}A, \tag{4}$$

since for a constant $\boldsymbol{U}$ vector, we have $\nabla\cdot\boldsymbol{U} = \frac{\alpha}{\mathscr{R}}\boldsymbol{U}\cdot\boldsymbol{e}_Y$. We have expressed the volume integral of the gradient operator as a function of a surface integral plus a source term, which ensures the compatibility with the surface integral form of the divergence operator. This approach leads to a discretization which is known as *Control Volume formulation* (CV). An alternative approach to define the surface integral form of the gradient operator is obtained by setting

$$\int_V \boldsymbol{\nabla}P\,\mathrm{d}V = \int_A \boldsymbol{\nabla}P\mathscr{R}\,\mathrm{d}A = \overline{\mathscr{R}}\int_A \boldsymbol{\nabla}P\,\mathrm{d}A.$$

Here, we have used the mean value theorem, hence $\overline{\mathscr{R}}$ is defined as the averaged pseudo radius $\overline{\mathscr{R}} = \frac{1}{|A|}\int_A \mathscr{R}\,\mathrm{d}A$, where $|A|$ is the surface of the area A. We remark that in the case of Cartesian geometry $\overline{\mathscr{R}} = 1$ since $\alpha = 0$. Finally, applying the Green formula, we get

$$\int_V \nabla P \, \mathrm{d}V = \overline{\mathscr{R}} \int_L P\boldsymbol{N} \, \mathrm{d}L. \tag{5}$$

We recover the Cartesian definition of the gradient operator weighted by the averaged pseudo radius. This alternative approach leads to the so-called *Area-Weighted formulation* (AW). We point out that, in this case, the compatibility between the surface integrals of the divergence and gradient operators is lost. Finally let us remark that formulae (5) and (4) coincide in the case of the Cartesian geometry since $\alpha = 0$ and $\overline{\mathscr{R}} = 1$.

3. Compatible cell-centered Lagrangian scheme

We develop a sub-cell force-based discretization over a domain $\mathscr{D}$ which is paved using a collection of polygonal cells without gap or overlaps. Such discretization has been introduced in [7,8]. Using the previous results and particularly the gradient operator definition given by (4), we rewrite the set of Eq. (1) in the control volume formulation over the moving polygonal cell $\Omega_c(t)$ as

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \int_{\partial\Omega_c(t)} \boldsymbol{U} \cdot \boldsymbol{N} \mathscr{R} \, \mathrm{d}L = 0, \tag{6a}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \int_{\partial\Omega_c(t)} P\boldsymbol{N}\mathscr{R} \, \mathrm{d}L = \alpha A_c P_c \boldsymbol{e}_Y, \tag{6b}$$

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}E_c + \int_{\partial\Omega_c(t)} P\boldsymbol{U} \cdot \boldsymbol{N}\mathscr{R} \, \mathrm{d}L = 0. \tag{6c}$$

Here, A_c is the area of the cell $\Omega_c(t)$ and m_c its constant mass. For any fluid variable ϕ, ϕ_c denotes its mass density average, i.e. $\phi_c = \frac{1}{m_c}\int_{\Omega_c(t)} \rho\phi \, \mathrm{d}V$. The area-weighted formulation is obtained using (5) for the gradient operator definition. In comparison to the control volume formulation, the previous system only differs in the momentum equation. Using the notations previously introduced, the area-weighted formulation of the momentum equation writes

$$m_c \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \overline{\mathscr{R}}_c \int_{\partial\Omega_c(t)} P\boldsymbol{N} \, \mathrm{d}L = 0, \tag{7}$$

where the cell averaged pseudo radius is $\overline{\mathscr{R}}_c = \frac{1}{A_c}\int_{A_c} \mathscr{R} \, \mathrm{d}A$. We point out that, in the case of Cartesian geometry $\overline{\mathscr{R}}_c = 1$ for all c, therefore the area-weighted formulation coincides with the control volume formulation. Moreover recalling that $m_c = V_c\rho_c$ and $\overline{\mathscr{R}}_c = V_c/A_c$ implies that (7) can be rewritten as

$$\mu_c \frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \int_{\partial\Omega_c(t)} P\boldsymbol{N} \, \mathrm{d}L = 0, \tag{8}$$

where $\mu_c = A_c\rho_c = m_c\overline{\mathscr{R}_c}$ denotes the Cartesian inertia. Consequently (8) has the same form as the momentum equation written in Cartesian geometry although the Cartesian inertia is not a Lagrangian mass (i.e it is not constant in time).

We have written a set of semi-discrete evolution equations for the cell-centered variables $\left(\frac{1}{\rho_c}, \boldsymbol{U}_c, E_c\right)$, whose thermodynamic closure is given by the EOS, $P_c = P(\rho_c, \varepsilon_c)$, where $\varepsilon_c = E_c - \frac{1}{2}\|\boldsymbol{U}_c\|^2$. The motion of the grid is ruled by the discrete trajectory equation written at each point: $\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{X}_p = \boldsymbol{U}_p(\boldsymbol{X}_p(t), t), \boldsymbol{X}_p(0) = \boldsymbol{x}_p$, where $\boldsymbol{X}_p$ denotes the position vector of point p and $\boldsymbol{U}_p$ its velocity. Let us note that by setting $\alpha = 0$ in the previous set of equations we recover the same system as in Cartesian geometry [3]. In the following we determine the numerical fluxes and the nodal velocity used to move the grid.

3.1. Geometric conservation law

Introducing $V_c = \int_{\Omega_c(t)} \mathscr{R} \, \mathrm{d}A$ the measure of the volume obtained by rotation of the polygonal cell Ω_c about X-axis, Eq. (6a) writes as the GCL

$$\frac{\mathrm{d}}{\mathrm{d}t}V_c - \int_{\partial\Omega_c(t)} \boldsymbol{U} \cdot \boldsymbol{N}\mathscr{R} \, \mathrm{d}L = 0. \tag{9}$$

Likewise in the case of Cartesian geometry, we use the fact that V_c is a function of the position vector $\boldsymbol{X}_p$ of point $p \in \mathscr{P}(c)$, where $\mathscr{P}(c)$ denotes the set of points of the Lagrangian cell Ω_c. The cylindrical corner area vector, refer to Fig. 2 is given by

$$\mathscr{A}_{pc}\boldsymbol{N}_{pc} = \frac{1}{2}\left[\frac{\mathscr{R}_{p^-} + 2\mathscr{R}_p}{3}(\boldsymbol{X}_p - \boldsymbol{X}_{p^-}) + \frac{\mathscr{R}_{p^+} + 2\mathscr{R}_p}{3}(\boldsymbol{X}_{p^+} - \boldsymbol{X}_p)\right] \times \boldsymbol{e}_z$$

where $\mathscr{A}_{pc}$ is the corner area that can be computed knowing that $\boldsymbol{N}_{pc}^2 = 1$. Noticing that the half-edge outward normals are given by $L_{pc}^{\pm}\boldsymbol{N}_{pc}^{\pm} = \pm\frac{1}{2}(\boldsymbol{X}_{p^{\pm}} - \boldsymbol{X}_p) \times \boldsymbol{e}_z$, we rewrite the previous equation as

$$\mathscr{A}_{pc}\boldsymbol{N}_{pc} = \frac{\mathscr{R}_{p^-} + 2\mathscr{R}_p}{3}L_{pc}^-\boldsymbol{N}_{pc}^- + \frac{\mathscr{R}_{p^+} + 2\mathscr{R}_p}{3}L_{pc}^+\boldsymbol{N}_{pc}^+. \tag{10}$$

As noticed by Whalen in [9], the corner area vector is the fundamental geometric object that uniquely defines the time rate of change of the cell volume as

$$\frac{\mathrm{d}}{\mathrm{d}t}V_c = \sum_{p\in\mathscr{P}(c)} \mathscr{A}_{pc}\boldsymbol{N}_{pc} \cdot \boldsymbol{U}_p. \tag{11}$$

This last result yields the definition of the discrete divergence operator over cell Ω_c as follows:

$$(\nabla \cdot \boldsymbol{U})_c = \frac{1}{V_c}\frac{\mathrm{d}}{\mathrm{d}t}V_c = \frac{1}{V_c}\sum_{p\in\mathscr{P}(c)} \mathscr{A}_{pc}\boldsymbol{N}_{pc} \cdot \boldsymbol{U}_p. \tag{12}$$

We claim that we have completely defined the volume flux in terms of the corner area vector and the nodal velocity, moreover this derivation is compatible with the mesh motion.

3.2. Sub-cell force-based discretization

Let us discretize momentum and total energy equations by means of sub-cell forces. To this end we use the partition of each polygonal cell Ω_c into sub-cells Ω_{pc}, where $p \in \mathscr{P}(c)$ (see Fig. 2). The sub-cell force that acts from sub-cell onto point is defined as

$$\boldsymbol{F}_{pc} = \int_{\partial\Omega_{pc}\cap\partial\Omega_c} P\boldsymbol{N}\mathscr{R} \, \mathrm{d}L. \tag{13}$$

We also use the sub-cell based partition to approximate the total energy flux as

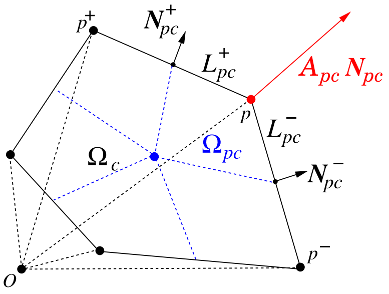

Fig. 2. Polygonal cell Ω_c in cylindrical geometry. Given the half-edge outward normals $L_{pc}^{\pm}\mathrm{N}_{pc}^{\pm}$ at point p and two consecutive points p^-, p^+ one defines the cylindrical corner area vector as $\mathscr{A}_{pc}\mathrm{N}_{pc} = \frac{\mathscr{R}_{p^-}+2\mathscr{R}_p}{3}L_{pc}^-\mathrm{N}_{pc}^- + \frac{\mathscr{R}_{p^+}+2\mathscr{R}_p}{3}L_{pc}^+\mathrm{N}_{pc}^+$. The partition into sub-cells Ω_{pc} is shown.

$$\int_{\partial\Omega_c} P\boldsymbol{U}\cdot\boldsymbol{N}\mathscr{R}\,\mathrm{d}L = \sum_{p\in\mathscr{P}(c)}\left(\int_{\partial\Omega_{pc}\cap\partial\Omega_c} P\boldsymbol{N}\mathscr{R}\,\mathrm{d}L\right)\cdot\boldsymbol{U}_p = \sum_{p\in\mathscr{P}(c)}\boldsymbol{F}_{pc}\cdot\boldsymbol{U}_p.$$

Substituting the previous results into system (6c) yields

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathscr{P}(c)}\mathscr{A}_{pc}\boldsymbol{N}_{pc}\cdot\boldsymbol{U}_p = 0, \tag{14a}$$

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{U}_c + \sum_{p\in\mathscr{P}(c)}\boldsymbol{F}_{pc} = \alpha P_c A_c \boldsymbol{e}_Y, \tag{14b}$$

$$m_c\frac{\mathrm{d}}{\mathrm{d}t}E_c + \sum_{p\in\mathscr{P}(c)}\boldsymbol{F}_{pc}\cdot\boldsymbol{U}_p = 0. \tag{14c}$$

We have expressed the numerical fluxes in terms of the corner area vector, the sub-cell force and the nodal velocity. The last two remain to be determined to complete the discretization. This task is achieved by investigating the thermodynamic consistency and the conservation of the sub-cell force-based discretization [4]. To ensure a local entropy inequality, it is sufficient to postulate the following form for the sub-cell force:

$$\mathsf{F}_{pc} = \mathscr{A}_{pc}P_c\mathsf{N}_{pc} - \mathsf{M}_{pc}(\mathsf{U}_p - \mathsf{U}_c). \tag{15}$$

Here M_{pc} is a sub-cell based 2 × 2 matrix such that: M_{pc} is symmetric, and, M_{pc} is positive semi-definite. The physical dimension of M_{pc} corresponds to an area times a density times a velocity. We remark that entropy production within cell c is directly governed by the general form of the sub-cell matrix M_{pc} and the velocity jump between the nodal and the cell-centered velocity, $\boldsymbol{\Delta U}_{pc} = \boldsymbol{U}_p - \boldsymbol{U}_c$. Finally total energy conservation is ensured provided that for all point p

$$\sum_{c\in\mathscr{C}(p)}\mathsf{F}_{pc} = \mathbf{0}. \tag{16}$$

We remark that this last equation is the same condition than the one obtained in Cartesian geometry for any compatible cell-centered or staggered sub-cell based discretization. Moreover under this condition, and, up to the boundary terms and the radial source term contributions, momentum is conserved over the entire domain. This result is remarkable in the sense that it is written under the same form regardless the geometry.

The last unknowns of the scheme, namely the sub-cell matrix M_{pc} and the node velocity $\boldsymbol{U}_p$, are obtained thanks to a node-centered Riemann solver.

3.3. Node-centered Riemann solver

The node-centered solver that provides the grid velocity is obtained as a consequence of total energy conservation. Substituting the sub-cell force (15) into (16) gives for all point p

$$\mathsf{M}_p\mathsf{U}_p = \sum_{c\in\mathscr{C}(p)}(\mathscr{A}_{pc}P_c\mathsf{N}_{pc} + \mathsf{M}_{pc}\mathsf{U}_c), \tag{17}$$

where M_p is the sum of the corner matrices around node p, which is defined as $\mathsf{M}_p = \sum_{c\in\mathscr{C}(p)}\mathsf{M}_{pc}$. We construct the natural extension of the Cartesian cell-centered scheme [3] to cylindrical geometry by defining the corner matrix as

$$\mathsf{M}_{pc} = z^-_{pc}\mathscr{R}^-_{pc}L^-_{pc}\left(\mathsf{N}^-_{pc}\otimes\mathsf{N}^-_{pc}\right) + z^+_{pc}\mathscr{R}^+_{pc}L^+_{pc}\left(\mathsf{N}^+_{pc}\otimes\mathsf{N}^+_{pc}\right), \tag{18}$$

where $\mathscr{R}^\pm_{pc} = \frac{1}{3}\left(\mathscr{R}_{p^\pm} + 2\mathscr{R}_p\right)$. We recall that $z^\pm_{pc}$ are the generalized non-linear corner impedances given by $z^\pm_{pc} = \rho_c[a_c + \Gamma_c|(\mathsf{U}_p - \mathsf{U}_c)\cdot\mathsf{N}^\pm_{pc}|]$, where a_c is the isentropic sound speed and Γ_c is a material dependent parameter, which is given by $\frac{\gamma+1}{2}$ in case of a gamma gas law. Note that this formula is the two-dimensional extension of the 2-shock swept mass flux defined for one-dimensional approximate Riemann problem initially proposed by Dukowicz [10] for shock wave. We also mention that we recover the acoustic approximation by setting $\Gamma_c = 0$. One can easily check that this definition leads to a symmetric positive definite M_{pc} matrix. Therefore, M_p is also symmetric positive definite and thus always invertible, which defines a unique nodal velocity $\boldsymbol{U}_p$ by inverting Eq. (17). Let us mention that this solver preserves the spherical symmetry in the case of a one-dimensional spherical flow computed on an equal angle polar grid.

The high-order extension of our control volume discretization, both in time and space, is obtained by using the Generalized Riemann Problem (GRP) methodology in the acoustic approximation (see [4] for the details). Moreover an extension of this cell-centered Lagrangian scheme in area-weighted formulation is also available [4]. For multi-species computation one simply considers the iso-pressure, iso-temperature closure model. Each fluid is characterized by its mass fraction C_f, and during the Lagrangian phase, the concentration of each fluid evolves following the trivial equation $\frac{d}{dt}C_f = 0$ (refer to [1]).

4. Rezone and remap in cylindrical geometry

As mentioned in [1] ReALE consists in modifying the rezone and remap phases of an ALE code assuming that the Lagrangian scheme can handle polygonal mesh. The cell-centered Lagrangian scheme previously described in its control volume or area-weighted version is well suited for this purpose. Therefore it is adopted as the first phase of our ReALE algorithm. The extension of the rezone and remap phases is presented in the following subsections.

4.1. Rezone phase through Voronoi machinery

In cylindrical geometry the simulation is performed on an actual 2D mesh. Any notion of mesh symmetry is therefore to be considered in the plane (Z,R). Consequently, the rezone phase is assumed to operate on this plane behaving "as a Cartesian plane". Therefore neither the generator displacement nor the Voronoi machinery is modified compared to [1]. However because ReALE cornerstone lays in the generator displacement and for the sake of clarity we recall these steps.

Let Ω^n_c and Ω^{n+1}_c denotes the Lagrangian cells at time t^n and $t^{n+1} = t^n + \Delta t$, where Δt is the current time step. The position vector of the generator of the Lagrangian cell Ω^n_c is denoted G^n_c (see Fig. 3). We will define the new position of the generator at time t^{n+1}. First, we compute a Lagrangian-like displacement of the generator by setting

$$\mathsf{G}^{n+1,\mathrm{lag}}_c = \mathsf{G}^n_c + \Delta t\mathsf{U}_c, \tag{19}$$

where $\boldsymbol{U}_c$ is the "pseudo-Lagrangian" velocity of the generator within the cell. This velocity is computed so that the generator remains located in the new Lagrangian cell. To this end we define this velocity to be the average of the velocities of the points of the cell, namely $\mathsf{U}_c = \frac{1}{|\mathscr{P}(c)|}\sum_{p\in\mathscr{P}(c)}\mathsf{U}^{n+\frac{1}{2}}_p$. Here $\mathsf{U}^{n+\frac{1}{2}}_p$ is the time-centered velocity of point p between times t^n and t^{n+1}. Any other formula could be used, as instance by weighting the point velocity by the distance between G^n_c and X^n_c. Let us introduce the centroid of the Lagrangian cell $\mathsf{X}^{n+1}_c = \frac{1}{|\Omega^{n+1}_c|}\int_{\Omega^{n+1}_c}\mathsf{X}\,\mathrm{d}V$, where $|\Omega^{n+1}_c|$ denotes the volume of the cell Ω^{n+1}_c. The updated position of the generator is defined by mean of a convex combination between the new Lagrangian-like position, $\mathsf{G}^{n+1,\mathrm{lag}}_c$ and the centroid X^{n+1}_c of the Lagrangian cell

$$\mathsf{G}^{n+1}_c = \mathsf{G}^{n+1,\mathrm{lag}}_c + \omega_c\left(\mathsf{X}^{n+1}_c - \mathsf{G}^{n+1,\mathrm{lag}}_c\right), \tag{20}$$

where $\omega_c \in [0;1]$ is a parameter that remains to determine.With this convex combination, the updated generator lies in between its Lagrangian position at time t^{n+1} and the centroid of the Lagrangian cell Ω^{n+1}_c.We note that for $\omega_c = 0$ we get a Lagrangian-like

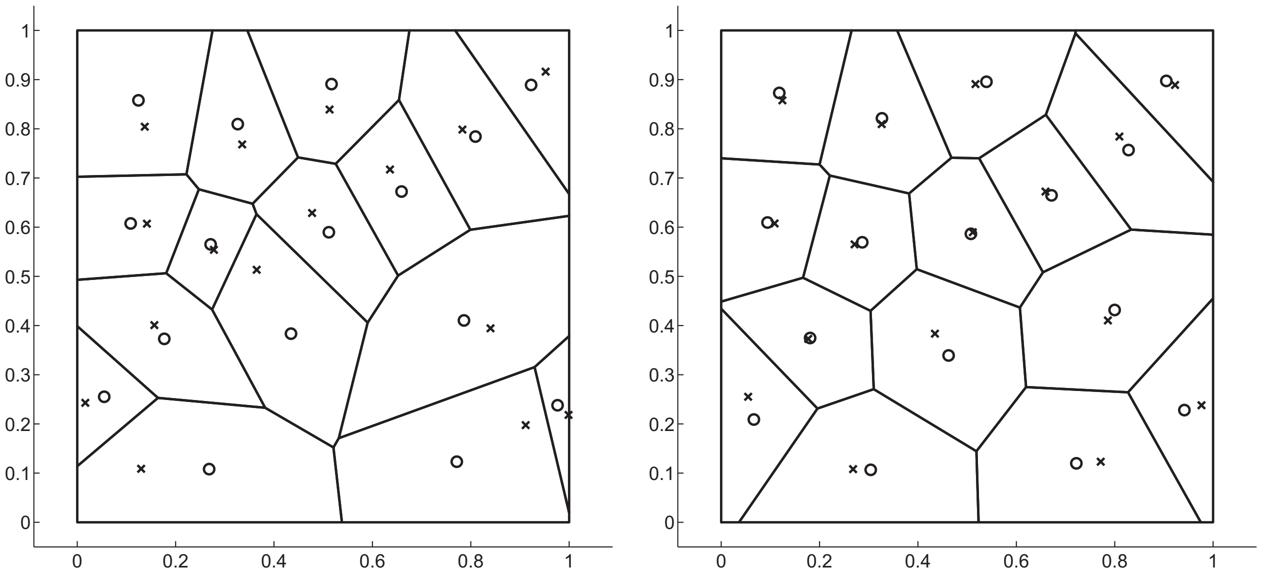

Fig. 3. Examples of Voronoi tesselation with cell generators $\boldsymbol{G}_c$ (×) and cell centroids $\boldsymbol{X}_c$ (○).

motion of the generator whereas for $\omega_c = 1$ we obtain a centroidal-like motion, which tends to produce a smooth mesh.[1] We compute ω_c requiring that the generator displacement satisfies the principle of material frame indifference, that is for pure uniform translation or rotation we want ω_c to be zero.To this end, we construct ω_c using invariants of the right Cauchy–Green strain tensor associated to the Lagrangian cell Ω_c between times t^n and t^{n+1}.Let us recall some general notions of continuum mechanics to define this tensor.First, we define the Cartesian deformation gradient tensor $\mathsf{F} = \frac{\partial \mathsf{X}^{n+1}}{\partial \mathsf{X}^n}$, where $\boldsymbol{X}^{n+1} = (X^{n+1}, Y^{n+1})^t$ denotes the vector position of a point at time t^{n+1} that was located at position $\boldsymbol{X}^n = (X^n, Y^n)^t$ at time t^n.The Cartesian deformation gradient tensor is the Jacobian matrix of the map that connects the Lagrangian configurations at time t^n and t^{n+1}.The right Cauchy–Green strain tensor, $\mathsf{C} = \mathsf{F}^t\mathsf{F}$, is a 2×2 symmetric positive definite tensor.We notice that this tensor reduces to the unitary tensor in case of uniform translation or rotation. It admits two positive eigenvalues, λ_1 and λ_2 with the convention $\lambda_1 \leqslant \lambda_2$.These can be viewed as the rates of expansion in a given direction during the transformation.To determine ω_c, we first construct the cell-averaged value of the deformation gradient tensor, F_c, and then the cell-averaged value of the Cauchy–Green tensor by setting $\mathsf{C}_c = \mathsf{F}_c^t\mathsf{F}_c$.Noticing that the two rows of the F matrix correspond to the gradient vectors of the X and Y-coordinates, we can set $\mathsf{F}^t = [\nabla_n X^{n+1}, \nabla_n Y^{n+1}]$, where for any functions$\psi = \psi(\boldsymbol{X}^n)$, we have $\nabla_n\psi = \left(\frac{\partial\psi}{\partial X^n}, \frac{\partial\psi}{\partial Y^n}\right)^t$. With these notations, one defines the cell-averaged value of the gradient of the ψ function over the Lagrangian cell Ω_c^n

$$
\begin{aligned}
(\nabla_n\psi)_c &= \frac{1}{|\Omega_c^n|}\int_{\Omega_c^n}\nabla_n\psi\,\mathrm{d}V = \frac{1}{|\Omega_c^n|}\int_{\partial\Omega_c^n}\psi\mathsf{N}\,\mathrm{d}S \\
&\simeq \frac{1}{|\Omega_c^n|}\sum_{p=1}^{|\mathscr{P}(c)|}\frac{1}{2}\left(\psi_p^n + \psi_{p+1}^n\right)L_{p,p+1}^n\mathsf{N}_{p,p+1}^n \qquad (21)
\end{aligned}
$$

where $\psi_p^n \equiv \psi(\mathsf{X}_p^n)$ and $L_{p,p+1}^n\mathsf{N}_{p,p+1}^n$ is the unit outward normal to the edge $\left[\mathsf{X}_p^n, \mathsf{X}_p^{n+1}\right]$. In the previous equation, we have first used the Green formula then an approximation of the integral using the trapezoidal rule on a polygonal cell. Applying (21) to $\psi = X^{n+1}$ and $\psi = Y^{n+1}$ we get a cell-averaged expression of the gradient tensor F and, consequently deduce the cell-averaged value of the tensor C_c. Knowing this symmetric positive definite tensor in each cell, we compute its real positive eigenvalues $\lambda_{1,c}$, $\lambda_{2,c}$ and define the parameter $\omega_c = \frac{1-\alpha_c}{1-\alpha_{\min}}$, where $\alpha_c = \frac{\lambda_{1,c}}{\lambda_{2,c}}$ and $\alpha_{\min} = \min_c\alpha_c$. We emphasize the fact that for uniform translation or rotation $\lambda_{1,c} = \lambda_{2,c} = 1$ and $\omega_c = 0$, therefore the motion of the generator is quasi Lagrangian and we fulfill the material frame indifference requirement. For other cases, ω_c smoothly varies between 0 and 1. Once the new generator position G_c^{n+1} is computed one constructs the corresponding Voronoi mesh using the Voronoi machinery. This mesh needs a last treatment as this Voronoi mesh may have arbitrary small edges. Such edges can drastically and artificially reduce the time step, and, more important can lead to a lack of robustness. Even if in theory such faces could be kept, we prefer to remove/clean them, see [1].

4.2. Remap phase by exact-intersection

The remapping phase is a conservative interpolation of physical variables from the Lagrangian polygonal mesh at the end of the Lagrangian step onto the new polygonal mesh after the rezone step. The remapping phase must provide valid physical variables to the Lagrangian scheme, and conservation of mass, momentum and total energy must be ensured. Moreover at least a second-order accuracy remapping has to be constructed. In ReALE the rezoned mesh may have a different connectivity than the Lagrangian mesh. Therefore the remapping phase of ReALE is based on exact-intersection of *a priori* two different polygonal meshes. Primary variables are cell-centered density, velocity and specific total energy. Conservative quantities are cell-centered mass, momentum and total energy. First piecewise linear representations of cell-centered variables ρ_c, $\rho_c\mathbf{U}_c$, $\rho_c E_c$ are constructed on the Lagrangian mesh. Then a slope limiting process [12] is performed to enforce physically justified bounds. This phase does not change compared to the Cartesian geometry. Then conservative quantities, namely mass, momentum and total energy, are obtained by integration of these representations in cylindrical geometry. Moreover for multi-species computation each mass fraction is remapped.

4.2.1. Control volume based remap

In control volume formulation, volume integrations are performed using the true cylindrical volume, $V = \int_\Omega \mathscr{R}\,\mathrm{d}A$, that is to say the measure of the volume obtained by rotation of the surface Ω about the Z-axis. New conservative quantities are calculated by

[1] This latter case is equivalent to perform one Lloyd iteration [11].

integration over polygons of intersection of new (rezoned) and old (Lagrangian) meshes. Let us consider one non empty polygon resulting from the intersection between an old cell $\Omega_c^{\rm old}$ and a new cell $\Omega_d^{\rm new}$ namely $\Omega_{cd} = \Omega_c^{\rm old} \bigcap \Omega_d^{\rm new}$. Then the mass embedded into this polygon is obtained by integration over Ω_{cd} of the piecewise linear limited representation of cell-centered density $\rho_c(\boldsymbol{X})$

$$\Delta m_{cd} = \int_{\Omega_{cd}} \rho_c(\mathrm{X})\mathscr{R}\,\mathrm{d}A. \tag{22}$$

Due to the linear representation of $\rho_c(\boldsymbol{X})$, the previous equation exhibits the integrals of $\mathscr{R},\ \mathscr{R}^2,\ \mathscr{R}Z$ which are reduced by the Green theorem to the boundary integrals, and subsequently evaluated from the coordinates of the Ω_{cd} region vertices. As a consequence the mass in any new cell $\Omega_d^{\rm new} = \bigcup_c \Omega_{cd}$ is simply obtained by summation

$$m_d^{\rm new} = \int_{\Omega_d^{\rm new}} \rho(\mathrm{X})\mathscr{R}\,\mathrm{d}A = \int_{\bigcup\limits_c \Omega_{cd}} \rho(\mathrm{X})\mathscr{R}\,\mathrm{d}A = \sum_{c\setminus\Omega_{cd}\neq\emptyset} \Delta m_{cd}. \tag{23}$$

Momentum and total energy are calculated likewise. Finally, primary variables in Ω_d are recovered by division by new volume $V_d^{\rm new}$ (for density) or new mass $m_d^{\rm new}$ (for momentum and energy).

To use the multi-species EOS, we need to remap the concentrations of the F fluids from the Lagrangian grid onto the rezoned one. To this end, we first compute the mass of fluid f in the Lagrangian cell $\Omega_c^{\rm old},\ m_{f,c} = \int_{\Omega_c^{\rm old}} \rho C\mathscr{R}\,\mathrm{d}A$. We note that $m_c^{\rm old} = \sum_{f=1}^F m_{f,c}$ since $\sum_{f=1}^F C_{f,c} = 1$. Then, the mass of each fluid is conservatively interpolated onto the rezoned grid following the methodology previously described. We denote its new value by $m_{f,c}^{\rm new}$. At this point we notice that $m_c^{\rm new} \neq \sum_{f=1}^F m_{f,c}^{\rm new}$, this discrepancy comes from the fact that our second-order remapping does not preserve linearity due to the slope limiting. Hence, we define the new concentrations $C_{f,c}^{\rm new} = \frac{m_{f,c^{\rm new}}}{m_c^{\rm new}}$ and impose the renormalization $C_{f,c}^{\rm new} \leftarrow \frac{C_{f,c}^{\rm new}}{\sum_{f=1}^F C_{f,c}^{\rm new}}$ so that $\sum_{f=1}^F C_{f,c}^{\rm new} = 1$. We point out that this renormalization does not affect the global mass conservation.

4.2.2. Area-weighted based remap

The difference between control volume and area-weighted formulation lays in the form of the momentum equation. As previously mentioned Eq. (7) has the same form as in Cartesian geometry modulo the presence of the Cartesian inertia $\mu_c = m_c\overline{\mathscr{R}_c}$. Consequently the remapping of the momentum equation in area-weighted cylindrical geometry is performed as in Cartesian geometry. Then the momentum embedded into $\Omega_{cd} = \Omega_c{}^{\rm old} \bigcap \Omega_d^{\rm new}$ is obtained by integration of the piecewise linear limited representation of cell-centered momentum $(\rho\boldsymbol{U})_c(\boldsymbol{X})$

$$\Delta \mathrm{W}_{cd} = \int_{\Omega_{cd}} (\rho\mathrm{U})_c(\mathrm{X})\mathscr{R}\,\mathrm{d}A = \overline{\mathscr{R}_c}\int_{\Omega_{cd}} (\rho\mathrm{U})_c(\mathrm{X})dXdY,$$

where the integration is performed over the Cartesian volume $dXdY$. The momentum in a new cell $\Omega_d^{\rm new}$ is given by

$$\mathrm{W}_d^{\rm new} = \int_{\Omega_d^{\rm new}} (\rho\mathrm{U})_c(\mathrm{X})\mathscr{R}\,\mathrm{d}A = \int_{\bigcup\limits_c \Omega_{cd}} (\rho\mathrm{U})_c(\mathrm{X})\mathscr{R}\,\mathrm{d}A = \sum_{c\setminus\Omega_{cd}\neq\emptyset} \Delta \mathrm{W}_{cd}.$$

The new velocity in cell $\Omega_d^{\rm new}$ is finally given by

$$\mathrm{U}_d^{\rm new} = \frac{\mathrm{W}_d^{\rm new}}{\mu_d} = \frac{\mathrm{W}_d^{\rm new}}{m_d^{\rm new}}\overline{\mathscr{R}_d}^{\rm new},$$

where $m_d^{\rm new}$ has been remapped using the true cylindrical volume and $\overline{\mathscr{R}_d}^{\rm new}$ has been recomputed on the new area $A_d^{\rm new}$.

5. Numerical tests

In this section we present the numerical results obtained by the cylindrical cell-centered ReALE code based on CHIC code, [3]. Let us remind that any vector is written in the (Z,R) space and that multi-species test cases are run with concentration equations. The first test is the well-known Sedov test case; it is used as a sanity check as no physical vorticity is expected to occur and therefore reconnection-based methods are not required. The second is a helium bubble shock interaction in cylindrical geometry, it is run in order to show the predictive capabilities of ReALE technique. This test generates vorticity which is a classical cause of failure for Lagrangian schemes. For a fixed-connectivity ALE code, it usually leads to a conflict between the Lagrangian motion with a tendency to tangle the mesh and, the mesh-regularization motion with a tendency to avoid bad quality cells. Such a conflict produces a stagnation of the mesh that reconnection technique is intended to cure [1]. Experimental results of the shock/bubble interaction are compared to the simulations. The last test problem is the rise of a light bubble under gravity for which the same type of vortex motion is expected. As no mesh symmetry is supposed to be preserved, we run the code in its control volume formulation for the last two test cases. Only the Sedov problem is run in area-weighted formulation to show the ability of the code to handle this formulation.

5.1. Sedov problem

Let us consider the Sedov blast wave problem with spherical symmetry. This problem models an intense explosion in a perfect gas with a diverging shock wave. The computational domain is $\Omega = [0,1.2] \times [0,1.2]$. The initial conditions are characterized by $(\rho_0,P_0,\boldsymbol{U}_0) = (1,10^{-6},\boldsymbol{0})$ for a perfect gas with polytropic index set to $\gamma = \frac{7}{5}$. We set an initial delta-function energy source at the origin prescribing the pressure in the cell containing the origin as $P_{\ \rm or} = (\gamma - 1)\rho_0 \frac{\mathscr{E}_0}{V_{\rm or}}$, where $V_{\rm or}$ denotes the volume of the cell and $\mathscr{E}_0$ is the total amount of released energy. Choosing $\mathscr{E}_0 = 0.425536$, the solution consists of a diverging shock whose front is located at radius $R = 1$ at time $t = 1$. The peak density reaches the value 6. Symmetry boundary conditions are applied on the axis. The initial mesh is a degenerate Voronoi mesh obtained from 50×50 uniformly distributed generators and four more generators on the corners of the domain. This test does not need ALE, and *a fortiori* ReALE, technique; pure Lagrangian schemes usually perform well. However this is used to assess the validity of ReALE approach. We present the density and mesh in Fig. 4 left-panel. Moreover density is presented as a function of the cell radius for any cell against the exact solution (straight line) in the right-panel of Fig. 4. The final Lagrangian mesh presents expanded cells in the rarefaction wave and compressed ones after the shock wave. On this sanity check ReALE technique in cylindrical geometry is able to produce a smooth mesh and accurate results.

5.2. Helium bubble shock interaction

The computational domain is $\Omega = [0;0.65] \times [0;0.089]$ which represents a cylinder of diameter 0.178 and initial length 0.65. The spherical helium bubble is represented as a disk defined by the coordinates of its center $(Z_c,R_c) = (0.320,0)$ and its radius $R_b = 0.0225$ (see Fig. 5). We prescribe wall boundary conditions at each boundary except at $Z = 0.65$, where we impose a piston-like boundary condition defined by the inward velocity $\boldsymbol{V}^\star = (u^\star,0)$. The incident shock wave is defined by its Mach number, $M_s = 1.25$. The bubble and the air are initially at rest. The initial data for helium are $(\rho_1,P_1) = (0.182,10^5)$, its molar mass is $\mathscr{M}_1 = 5.269 \times 10^{-3}$ and its polytropic index is $\gamma_1 = 1.648$. The initial

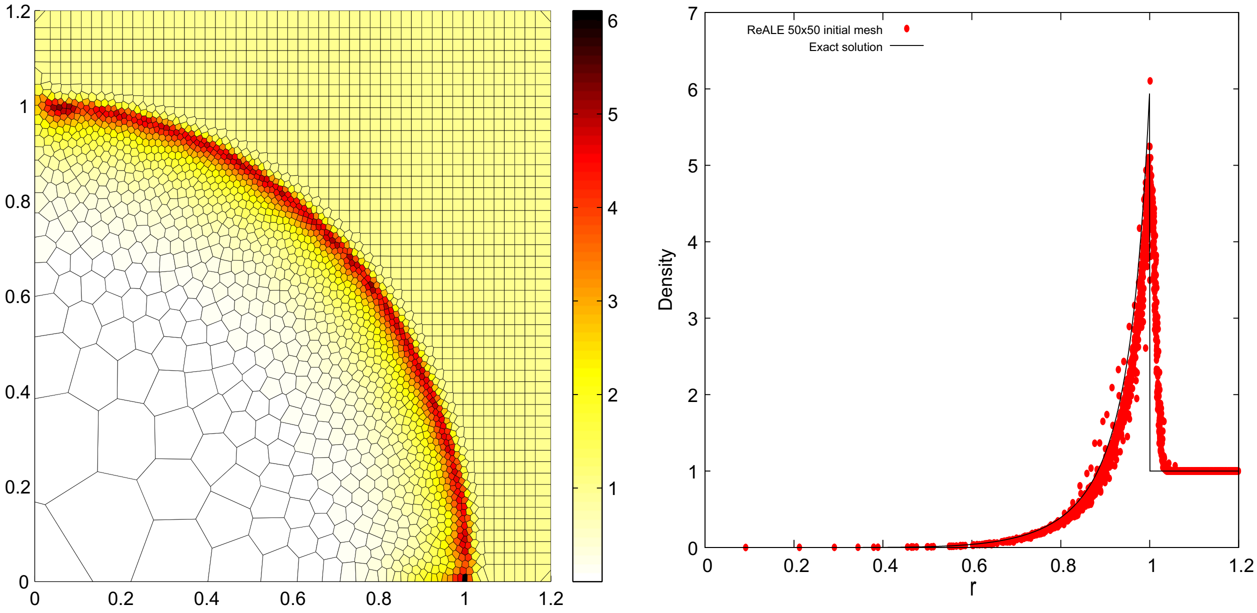

Fig. 4. Sedov problem in cylindrical geometry at time $t = 1.0$ for ReALE – left: mesh and density – right: density as a function of radius for all cells *vs* the exact solution (line).

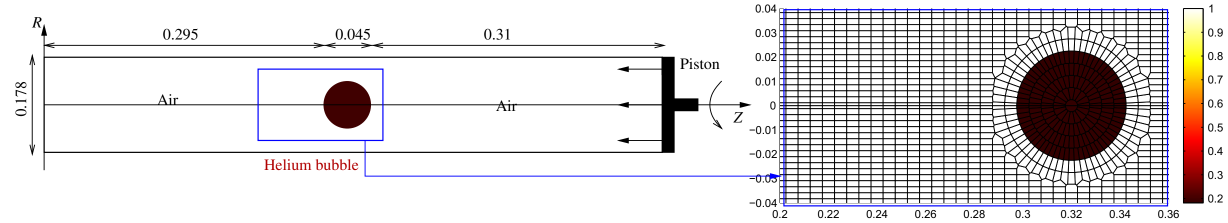

Fig. 5. Helium bubble shock interaction. A right piston compresses some air at rest by sending a $M = 1.25$ shock wave that passes through an helium bubble in cylindrical geometry — left: initial bubble, the computational domain is $\Omega = [0; 0.65] \times [0; 0.089]$ that is reflected against the Z-axis. Right: zoom on the initial mesh and density that are reflected against the z-axis.

data for air are $(\rho_2, P_2) = (1, 10^5)$, its molar mass is $\mathscr{M}_2 = 28.963 \times 10^{-3}$ and its polytropic index is $\gamma_2 = 1.4$. Specific internal energies are $\varepsilon_1 = 8.4792 \times 10^5$ and $\varepsilon_2 = 2.5 \times 10^5$. Using the Rankine–Hugoniot relations, we find that the velocity of the piston is given by $u^\star = -140.312$. The incident shock velocity is $D_c = -467.707$. The incident shock wave hits the bubble at time $t_i = 657.463 \times 10^{-6}$. The stopping time for our computation is $t_{end} = t_i + 1594 \times 10^{-6} = 2251.463 \times 10^{-6}$. The mesh is built with a set of 4856 generators designed to produce a Voronoi mesh that has a mesh line which exactly matches the bubble boundary, see Fig. 5. In all figures the top part is the actual computational domain that is mirrored around the Z-axis for visualization purposes. In Fig. 6 are displayed density and mesh around the bubble for five intermediate times and the final time of the simulation, namely $t_a = t_i + 20 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, $t_f = t_i + 600 \times 10^{-6}$ and $t_g = t_i + 1594 \times 10^{-6}$. These correspond to five shadow-photographs of experimental results from [13] (Fig. 8 of page 53) that we reproduced in Fig. 6 (right-panels). Let us note that the final time has a different color scale and that the visualization window follows the bubble. We observe a quite good agreement with the experimental results even for this coarse mesh; the timing of the shock waves and the shape of the deformed bubble fit the shadow-graphs of the experimental results. Of great importance is the fact that the bubble detaches from the Z-axis at t_f and more clearly at t_g, this can be also guessed from the experimental shadow-graphs. Finally in Fig. 7 are displayed the density waves present in the full domain at intermediate times $t_b = t_i + 82 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, and $t_e = t_i + 1007 \times 10^{-6}$. The dark zones are the inside bubble and the air that has not been yet attained by the initial shock wave. Multiple reflections and refractions can be observed in the density wave patterns.

5.3. Rise of a light bubble under gravity

This problem consists in the rise of a light bubble in a heavy gas bubble under gravity [14]. The statement of the problem is sketched in Fig. 8. The computational domain is $\Omega = [-15, 20] \times [0, 15]$. This domain is split into three regions filled with air (ideal gas EOS with $\gamma = 1.4$) at rest. One defines for each point (Z, R) its radius $\mathrm{R} = \sqrt{R^2 + Z^2}$ and angle $\boldsymbol{\theta}$ so that $Z(\boldsymbol{R}, \boldsymbol{\theta}) = \boldsymbol{R}\cos\boldsymbol{\theta}$. The initial data are given in Zone I (inside the bubble), Zone II (transition layer) and Zone III (exterior) by

Zone I $\boldsymbol{R} \leqslant R_1$	Zone II $R_1 < \boldsymbol{R} < R_2$	Zone III $\boldsymbol{R} \geqslant R_2$
$\varepsilon_1 = 3 \times 10^3$ $p_1 = 0.6$	$\varepsilon_2(\boldsymbol{R})$ $p_2(\boldsymbol{R}, \boldsymbol{\theta})$	$\varepsilon_3 = 15.6$ $p_3(\boldsymbol{R}, \boldsymbol{\theta}) = 0.6e^{-Z(\boldsymbol{R}, \boldsymbol{\theta})/\varDelta}$

$R_1 = 6.6$ is the radius of the light bubble, $R_2 = 8.5$ is the radius of the transition layer towards the atmosphere, $\varDelta = 63.7$ is the inhomogeneity parameter for the atmosphere. In Zone II a linear transition is applied between the values of p and ε of Zones I and III

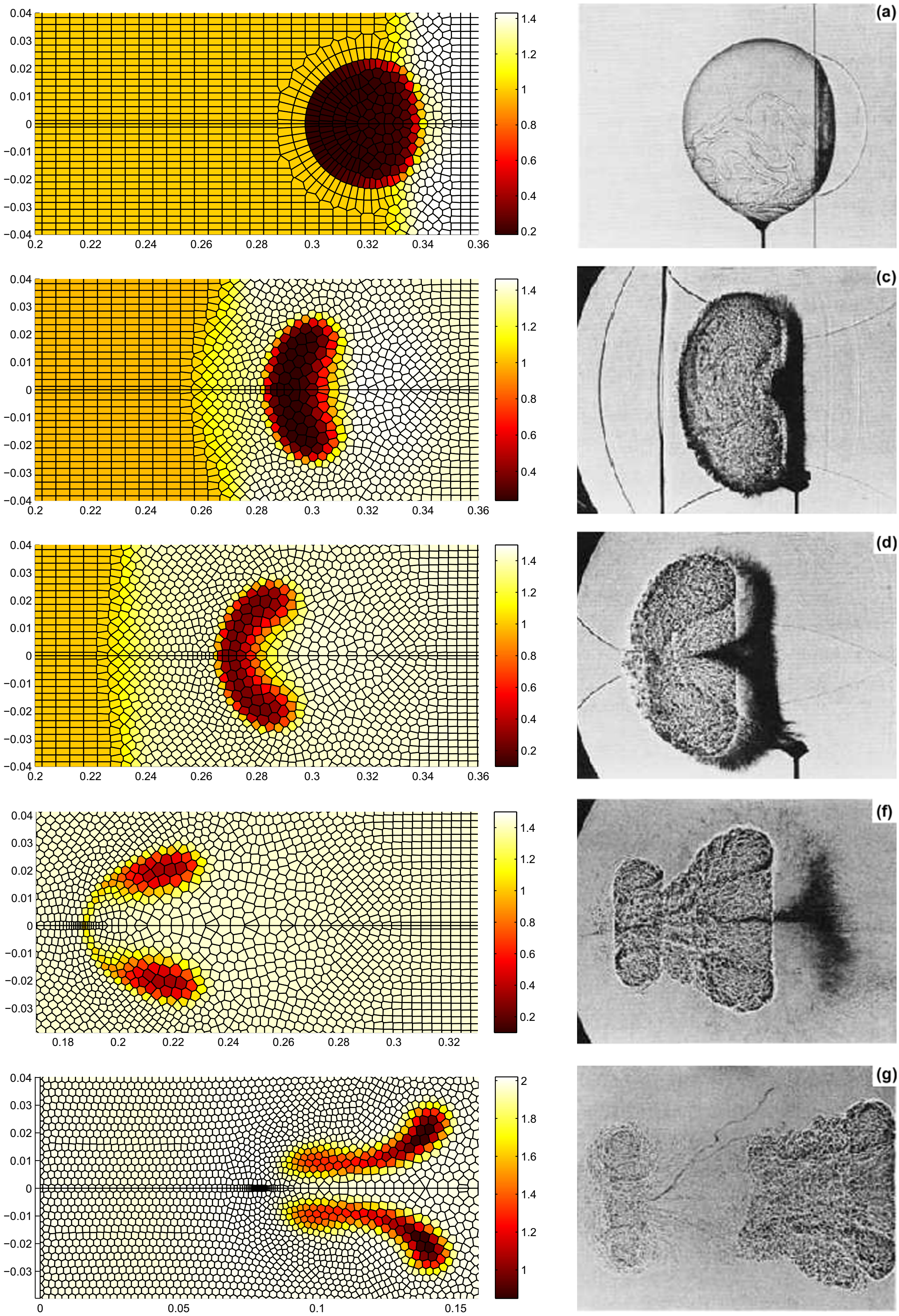

Fig. 6. $M = 1.25$ shock interaction with a spherical helium bubble — mesh and density — from top to bottom: $t_a = t_i + 20 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, $t_f = t_i + 600 \times 10^{-6}$ and $t_g = t_i + 1594 \times 10^{-6}$ where $t_i = 657.463 \times 10^{-6}$ is the time of the shock/bubble interaction. Left: ReALE in cylindrical geometry. Right: Experimental results from [13].

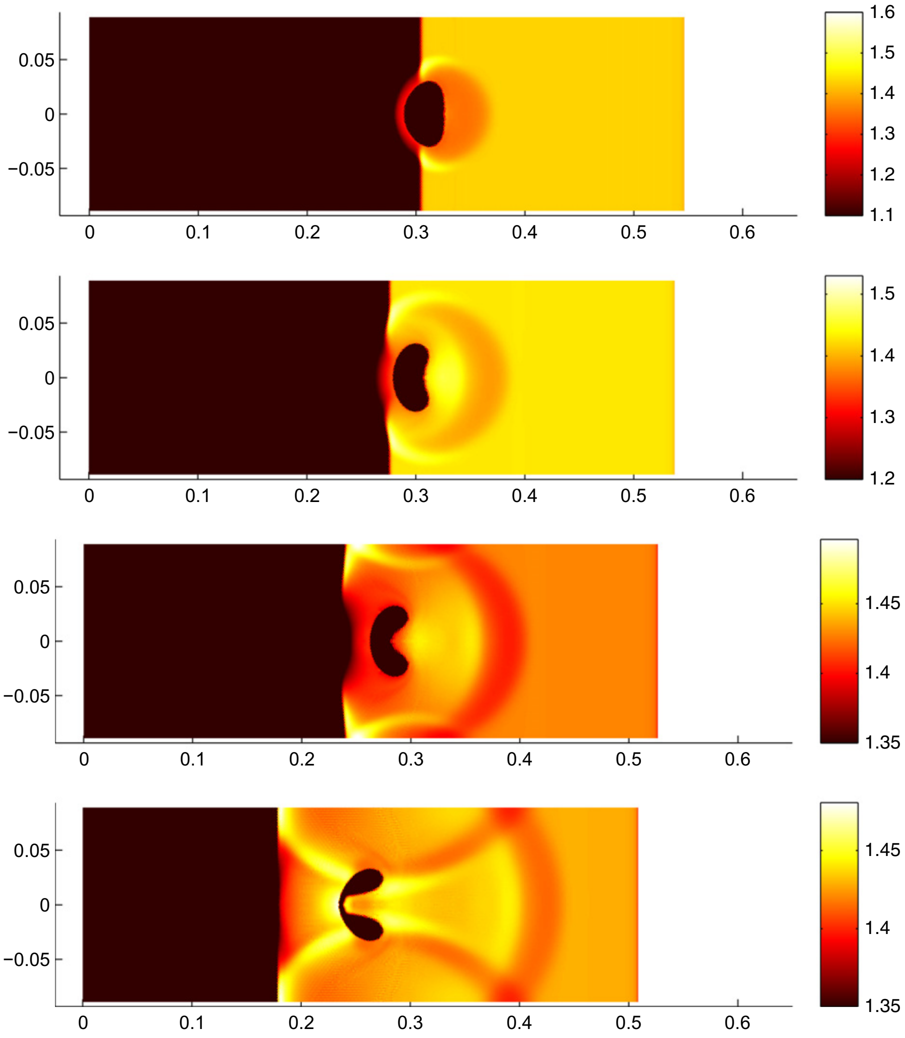

Fig. 7. $M = 1.25$ shock interaction with a spherical helium bubble — density waves in the domain — from top to bottom: $t_b = t_i + 82 \times 10^{-6}$, $t_c = t_i + 145 \times 10^{-6}$, $t_d = t_i + 223 \times 10^{-6}$, $t_e = t_i + 1007 \times 10^{-6}$ where $t_i = 657.463 \times 10^{-6}$ is the time of the shock/bubble interaction. ReALE in cylindrical geometry.

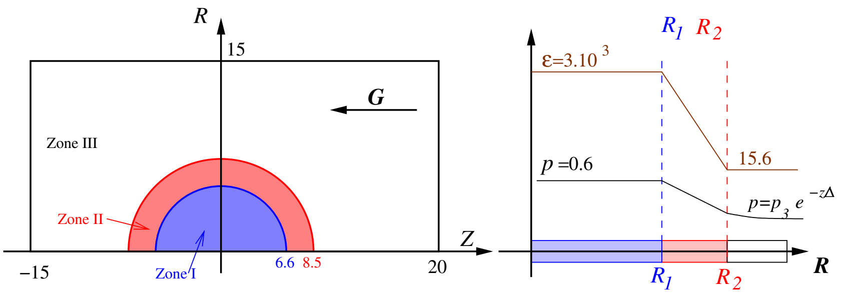

Fig. 8. Rise of a light bubble under gravity. The light bubble (Zone I) has a radius of R_1 and a transition layer is initialized between R_1 and R_2 (Zone II). The rest of the domain $\boldsymbol{R} > R_2$ is some air at rest (Zone III) where $\mathrm{R} = \sqrt{R^2 + Z^2}$. Gravity is oriented in the Z direction. The pressure and internal energy profiles are sketched in the right panel.

$$p_2(\mathrm{R}, \boldsymbol{\theta}) = (1 - \alpha(\mathrm{R}))p_1 + \alpha(\mathrm{R})p_3(\mathrm{R}, \boldsymbol{\theta}),$$
$$\varepsilon_2(\mathrm{R}) = (1 - \alpha(\mathrm{R}))\varepsilon_1 + \alpha(\mathrm{R})\varepsilon_3,$$

with $\alpha(\boldsymbol{R}) = (\boldsymbol{R} - R_1)/(R_2 - R_1)$. Gravity is set downward the Z direction with magnitude $G = 9.8 \times 10^{-2}$ see Fig. 8.The bubble rises in the Z direction because of the density gradients and velocity. In its motion it further deforms into a classical mushroom shape. The final time is $t_{\mathrm{final}} = 14$. The initial mesh is made of a total of 1901 cells split into roughly 1200 quadrangles outside Zone II and 653 polygonal cells inside refer to Fig. 9 top-middle panel. Walls boundary conditions are assumed everywhere besides for the Z-axis where symmetry boundary condition is applied. In Fig. 9 left column are plotted the density and mesh for the time moments $t_0 = 0$, $t_1 = 1$, $t_2 = 8$ and $t_3 = 14$. As for the bubble/shock problem the visualization is performed after reflection against the Z-axis. The colored vorticity and vector velocity fields are

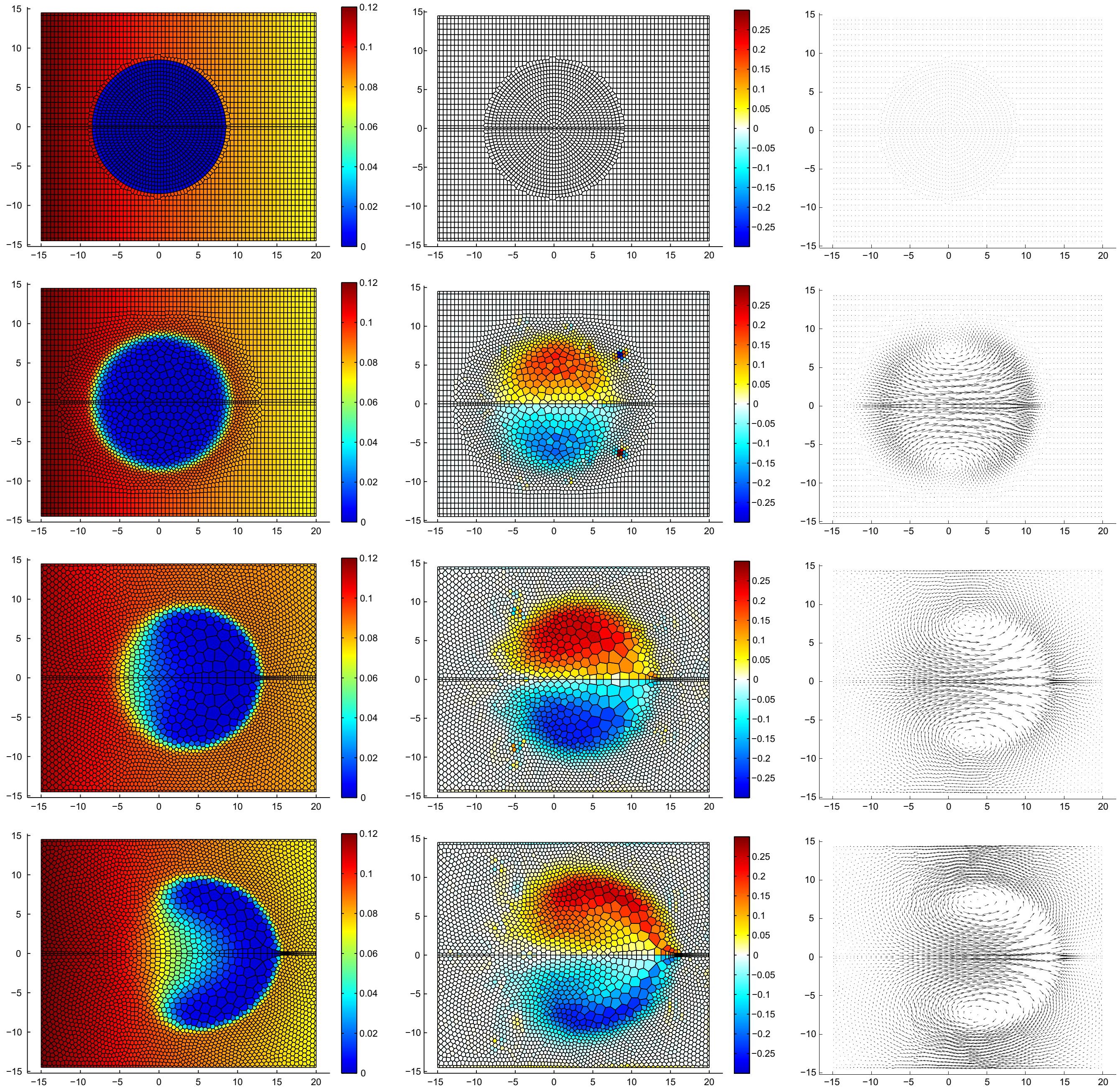

Fig. 9. Rise of a light bubble under gravity — ReALE in cylindrical geometry — left column: density and mesh — middle column: vorticity and mesh — right column: veolcity vectors — time $t_0 = 0$, $t_1 = 1$, $t_2 = 8$, $t_3 = 14$.

shown in Fig. 9 middle and right columns respectively. As expected the bubble rises upwards the Z direction. It adopts a mushroom shape as can be seen in Fig. 9 at $t_3 = 14$. Time $t_1 = 1$ shows how the ReALE technology starts to adapt the mesh while waves are emanating from the bubble. At time $t_2 = 8$ the bubble starts to deform while the cells ahead the tip of the bubble are highly compressed; this process is pursued up to final time. Vorticity and velocity vector plots confirm that the fluid undergoes a vortex-like motion. Such a vortex-like motion has a natural tendency to highly stretch cells in classical ALE simulation without reconnection leading to inaccuracy or failure of the simulation. Contrarily ReALE is able to undergo such motion while allowing cells to change neighborhood. These ReALE results are in agreement with the numerical results provided in [14]. In this paper the authors use a different numerical method that leads to a non smooth polygonal mesh as shown in Fig. 4.19 of page 111. ReALE technique form this point of view seems superior as our mesh keeps a general good geometrical quality.

6. Conclusion and perspectives

In this paper we investigate the extension to cylindrical geometry of the recently developed Reconnection Arbitrary-Lagrangian–Eulerian (ReALE) technology [1]. This extension is fairly obvious; indeed, the cell-centered Lagrangian scheme was already available in cylindrical geometry using a control volume or area-weighted formulation. Moreover, the rezone technology using Voronoi machinery with moving generators introduced in [1] can be used likewise. The last part of our ReALE code, namely the remapping part, is more demanding as it must utilize a control volume based exact-intersection of *a priori* two different polygonal meshes provided by the Lagrangian and rezone phases. In the control volume formulation true volume integrals are used to remap mass, momentum and total energy whereas in area-weighted formulation the momentum is remapped as in Cartesian geometry. Multi-fluid is treated with concentration equations that must be remapped likewise.

We show that the extension of ReALE to cylindrical geometry produces good results on numerical test cases. First we run the Sedov problem as a sanity check. Then we simulate an helium bubble shock interaction problem. We compare our multi-species simulation against experimental shadow-graphs proving the validity and accuracy of the ReALE technology in cylindrical geometry. The last problem is the rise of a light bubble under gravity that presents vortex-like motion. Unlike ReALE, a classical fixed-connectivity ALE code usually presents difficulties to capture such a motion.

In the near future we plan to investigate the association of ReALE with interface reconstruction in planar and cylindrical geometries. Moreover we will investigate the possible extension of ReALE to 3D.

Acknowledgments

This work was performed under the auspices of the National Nuclear Security Administration of the US Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and supported by the DOE Advanced Simulation and Computing (ASC) program. The authors acknowledge the partial support of the DOE Office of Science ASCR Program. The authors thank A. Solovjov for allowing to use his code for Voronoi mesh generation. The authors thank M. Kucharik, J. Dukowicz, F. Adessio, H. Trease, G. Ball, A. Barlow, P. Váchal, V. Ganzha, B. Wendroff, J. Campbell, D. Burton V. Tishkin, A. Favorskii, V. Rasskazova, N. Ardelyan, S. Sokolov for stimulating discussions over many years. The first author acknowledges the help of S. Galera and J. Breil in using CHIC ReALE code.

References

[1] Loubère R, Maire P-H, Shashkov M, Breil J, Galera S. ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method. JComput Phys 2010;229: 4724–61.

[2] Du Q, Faber V, Gunzburger M. Centroidal Voronoi tesselations: applications and algorithms. SIAM Review 1999;41:637–76.

[3] Maire P-H. A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes. J Comput Phys 2009;228:2391–425.

[4] Maire P-H. A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry. J Comput Phys 2009;228:6882–915.

[5] Caramana EJ, Burton DE, Shashkov MJ, Whalen PP. The construction of compatible hydrodynamics algorithms utilizing conservation of total energy. J Comput Phys 1998;146:227–62.

[6] Adessio FL, Carroll DE, Dukowicz KK, Johnson JN, Kashiwa BA, Maltrud ME, et al. CAVEAT: a computer code for fluid dynamics problems with large distortion and internal slip. Technical report LA-10613-MS, Los Alamos National Laboratory; 1986.

[7] Maire P-H. A unified sub-cell force-based discretization for cell-centered Lagrangian hydrodynamics on polygonal grid. Int J Numer Meth Fluids 2010, doi:10.1002/fld.2328.

[8] Maire P-H. Contribution to the numerical modeling of inertial confinement fusion, private communication.

[9] Whalen P. Algebraic limitations on two dimensionnal hydrodynamics simulations. J Comput Phys 1996;124:46–54.

[10] Dukowicz JK. A general non-iterative Riemann solver for Godunov's method. J Comput Phys 1985;61:119–37.

[11] Du Q, Emelianenko M, Gunzburger M. Convergence of the Lloyd algorithm for computing centroidal Voronoi tesselations. SIAM J Numer Anal 2006;44:102–19.

[12] Barth TJ. Numerical methods for gasdynamic systems on unstructured meshes. In: Kroner D, Ohlberger M, Rohde C, editors. An introduction to recent developments in theory and numerics for conservation laws. Proceedings of the international school on theory and numerics for conservation laws, lecture notes in computational science and engineering. Berlin: Springer; 1997. p. 195–284.

[13] Haas J-F, Sturtevant B. Interaction of weak-shock waves. J Fluid Mech 1987;181:41–76.

[14] Sofronov I, Rasskazova V, Nesterenko L. The use of nonregular nets for solving two-dimensional non-stationary problems in gas dynamics. In: Yanenko N, Shokin Y, editors. Numerical methods in fluid dynamics. Moscow: MIR; 1984. p. 82–121.