Journal of Computational Physics 229 (2010) 4724–4761

Contents lists available at ScienceDirect

Journal of Computational Physics

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

ReALE: A reconnection-based arbitrary-Lagrangian–Eulerian method

Raphaël Loubère [a], Pierre-Henri Maire [b], Mikhail Shashkov [c,*], Jérôme Breil [b], Stéphane Galera [b]

[a] Institut de Mathématiques de Toulouse, CNRS, Université de Toulouse, 118 Route de Narbonne, 31062 Toulouse cedex 9, France
[b] UMR CELIA, Université Bordeaux I 351, Cours de la Libération, 33405 Talence, France
[c] Los Alamos National Laboratory, T-5, Los Alamos, NM 87545, USA

ARTICLE INFO

Article history:
Received 15 December 2009
Received in revised form 4 March 2010
Accepted 9 March 2010
Available online 18 March 2010

Keywords:
Lagrangian hydrodynamics
Cell-centered scheme
Compressible flow
Staggered scheme
Voronoi mesh
Arbitrary-Lagrangian–Eulerian
Mesh reconnection
Multi-dimensional unstructured polygonal mesh

ABSTRACT

We present a new reconnection-based arbitrary-Lagrangian–Eulerian (ALE) method. The main elements in a standard ALE simulation are an explicit Lagrangian phase in which the solution and grid are updated, a rezoning phase in which a new grid is defined, and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid. In standard ALE methods the new mesh from the rezone phase is obtained by moving grid nodes without changing connectivity of the mesh. Such rezone strategy has its limitation due to the fixed topology of the mesh. In our new method we allow connectivity of the mesh to change in rezone phase, which leads to general polygonal mesh and allows to follow Lagrangian features of the mesh much better than for standard ALE methods. Rezone strategy with reconnection is based on using Voronoi tessellation. We demonstrate performance of our new method on series of numerical examples and show it superiority in comparison with standard ALE methods without reconnection.

1. Introduction

In numerical simulations of multi-dimensional fluid flow, the relationship of the motion of the computational grid to the motion of the fluid is an important issue. One of two choices is typically made: a Lagrangian framework or an Eulerian framework. In the Lagrangian framework, the grid moves with the local fluid velocity, while in the Eulerian framework, the fluid flows through a grid fixed in space. More generally, the motion of the grid can be chosen arbitrarily. The philosophy of the arbitrary-Lagrangian–Eulerian methodology (ALE; cf. [64,19,20,70,71,81,100,93]) is to exploit this degree of freedom to improve the accuracy and efficiency of the simulation. The main elements in an standard ALE simulation are an explicit Lagrangian phase in which the solution and grid are updated, a rezoning phase in which a new grid is defined, and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid [93]. Clearly ALE also includes Lagrangian approach when the mesh is not changing during rezone phase. It is important to note that Eulerian simulation can be considered as a limiting case of ALE when the rezoned mesh coincides with the mesh at the beginning of the Lagrangian step – such implementation of Eulerian methodology is usually called Eulerian-as-Lagrange-Plus-Remap [123,101] as opposed to direct Eulerian approach [38] in which gas dynamics equations in Eulerian form are directly solved. Therefore, ALE methodology is very flexible and successfully used to solve complicated problems.

The numerical methods related to Lagrangian phase of ALE are relatively well developed (for example [20,28,29,18,31,83,30,32,40,33,109,108,110,34,91,16,104,88,89,87]), as well as the remapping phase (being considered as conservative interpolation or advection) [20,94,77,61,84,101,124,49,21,48,98].

* Corresponding author.
E-mail addresses: raphael.loubere@math.univ-toulouse.fr (R. Loubère), maire@celia.u-bordeaux1.fr (P.-H. Maire), shashkov@lanl.gov (M. Shashkov), breil@celia.u-bordeaux1.fr (J. Breil), galera@celia.u-bordeaux1.fr (S. Galera).

0021-9991/$ - see front matter © 2010 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2010.03.011

Our opinion is that the most difficult and least developed phase of ALE is the rezoning phase. A review of existing rezone strategies for ALE methods is presented in [74] including analysis of alternative approaches [81,120,45,19,57,100,24,129,9,47,128]. A review of a more general class of methods, namely moving mesh methods, is presented in [79].

Ideally the mesh has to adapt to the solution. Any adaptive scheme is composed of three main ingredients: an optimal-mesh criterion, an error estimator or error indicator, and an algorithm of the strategy for the mesh improvement. These ingredients answer to the following questions: How should the optimal mesh be defined? Where are mesh changes required? And how should the improved mesh be constructed? For standard ALE methods a strategy for mesh improvement is based on moving the spatial grid. Generally speaking the goal of rezoning is to improve the efficiency of the ALE method, that is, to achieve a given accuracy with the least amount of "work". "Work" in this context should be understood not only as the CPU time but also as memory and man-hour resources. However, to design an adaptive method one needs a quantitative assessment of optimality. The problem is that, for non-linear equations of gas dynamics in 2D and 3D, at the moment, it is not feasible to obtain such quantitative assessment. For this reason practitioners are usually using some qualitative approaches. In real complex ALE simulation the most basic goal of rezoning is simply to run calculation to completion without user intervention and still achieve reasonable accuracy (recall that we always can run ALE in Eulerian = Lagrange-Plus-Remap mode, which will be robust but less accurate). Even this goal is usually not achieved in most production ALE codes. For example, even for very popular methods based on Winslow smoothing, [128,129], practical simulations require the introduction of numerous geometrical and physics-based triggers and lockers, that is, mesh constraints that typically keep a node Lagrangian until some condition is reached e.g. element quality criterion (to detect cell distortion or collapse) or physical condition is reached in surrounding elements (for example, did the cell fully detonate?) [100,14,15,65].

As it is mentioned in [14,15], the mesh movement philosophy applied to most applications, related to high-speed multi-material flows, is to develop algorithms that will move the mesh in such a way as to maintain robustness while staying as close as possible to the Lagrangian mesh motion. The Lagrangian mesh motion naturally follows most flow features of interest such as shocks, material interfaces and steep gradients and allows users to focus zoning in materials of interest. Mesh relaxation is then used in regions of high material deformation to improve mesh quality. In standard ALE methods, which use fixed mesh topology, nodes are moved to refine mesh some areas of the problem at the expense of coarsening mesh in other parts of the problem. Generally, the increase of mesh resolution is limited, and, most importantly it can degrade the mesh quality leading to robustness problems. One of the most cited papers in mesh rezoning is [24], where authors use variational approach to combine requirements related to maintaining geometric quality of the mesh and some mesh adaptation based on equidistribution of some error indicator. Functional which is responsible for mesh smoothness essentially can be considered as a variational form of Winslow approach for which the corresponding optimization problem is well behaved. In contrast, the functional responsible for error equidistribution, if used by itself, has multiple local minima and its minimization can lead to tangled mesh. A difficulty arises when one tries to combine these – how should one weight the relative importance of these separate goals and still obtain a well behaved optimization process? In particular, the two global functionals have distinct (physical) dimensions, and so, can only be combined with some dimensional constant. At present, there is no theoretical basis for choosing this constant, thus delegating the decision to the user. In practice, bad choice of this parameter can lead to loss of accuracy (if the mesh is over-smoothed) or robustness problem (if the mesh becomes tangled).

In our opinion, the main reason for this is that standard ALE codes utilize a fixed topology mesh, defined at the outset, which in general will not be able to adapt to the dynamically evolving interface shape (or contact discontinuity) in spite of efforts at regularization (see Section 2 for more details). The most general solution to this difficulty, while preserving a Lagrangian nature of mesh motion is to relax the constraints on mesh topology and allow reconnection.[1]

The idea of using mesh reconnection to solve partial differential equations is not new. To the best of our knowledge, in context of computational gas dynamics the ideas related to mesh reconnection were first used in [99].[2] In this seminal paper the authors suggest to use a set of point Lagrangian particles and surround them with domains (parcels) to describe the media. The shape and size of these parcels are determined by the positions of the particles. The connectivity of the set of particles is not fixed but can vary with time depending on relative positions of the particles. After connectivity is established the set of neighboring particles defines the stencil on which Lagrangian equations are discretized.

In our opinion paper [99] has all basic ideas that lead to development of so-called free-Lagrange (or free-Lagrangian) methods, [55,23,121], which were very popular in 80s and early 90s The name free-Lagrange was introduced in [39] and the corresponding code was called FLAG. We need to analyze the main features of free-Lagrange approach in order to precisely position our new reconnection-based ALE method with respect to well established methods.

We start our analysis with methods, which we will call *pure* free-Lagrangian methods, [95,122,54,114,113,107,10]. In these methods, the fluid is represented by point particles surrounded by parcels. Each particle and its corresponding parcel represent a single material, that is, there is no mixed (containing several materials) parcel. Mass exchange between particles is forbidden. Flow variables are stored at the particle within each parcel, which moves in a "Lagrangian" fashion. The shape of the parcel is determined by constructing Voronoi mesh [96], in which each particle (generator, site) is en-

[1] Let us also mention interesting approach of combining standard ALE method with adaptive mesh refinement (AMR) [3]. In this, approach authors allows subdivision of the cell into smaller cells similar to standard Eulerian AMR methods. Formally, it can be considered as connectivity change. Our opinion is that objective of ALE AMR is to increase local resolution. This method has all drawbacks of standard ALE methods with fixed connectivity.

[2] In this paper we are not considering meshless methods, like smoothed particle hydrodynamics – SPH – interested readers can refer for example to [23,80,66].

closed within convex polygonal cell. By definition, in a Voronoi mesh each cell encloses all points in the domain which are closer to the corresponding particle than to any other particle. The mesh is fully reconstructed after some number of time steps to allow grid connectivity to change naturally under influence of shear or vorticity. Connectivity of Voronoi mesh uniquely defines the "neighbors" that is particles which parcels share edge with given particle. These neighbors define the stencil on which gas dynamics equations are approximated. In "pure" free-Lagrange, one approximates some form of Lagrangian equations. Several discretization approaches are possible, for example, mimetic finite difference [95], or Godunov-type methods [10]. The energy equation can be approximated in conservative form [10], or in non-conservative form as in [95]. In later case mimetic approach is used to guarantee conservation of total energy. There are also different approximations used for continuity equation. Most standard approach is to define density as ratio of particle mass (which is constant in time) and volume of the corresponding parcel. However, one also can define the volume from a special evolution equation as it is usually done in Godunov based methods. Let us emphasize that there is no "remapping" phase in "pure" free-Lagrangian methods.

Now let us discuss how pure are "pure" free-Lagrangian methods. Such analysis requires the definition of what is a Lagrangian method. One can define a Lagrangian method as a method where equations in Lagrangian coordinates are approximated. That is to say, governing equations do not have advective terms; in other words no exchange of mass, momentum and energy between these particles is allowed. In case of "pure" free-Lagrangian methods these two criteria are satisfied. But this is not enough to call method Lagrangian. For a Lagrangian method the boundary of parcel has to move in Lagrangian fashion too. Unfortunately, for "pure" free-Lagrange it is not the case because parcels are Voronoi cells and the vertices of the Voronoi cell are not moved independently in a Lagrangian way but defined from positions of the particles in a very non-linear way [95]. It is well known [55,23,121] that a Voronoi cell is not a Lagrangian object and it is rigorously proven in [118] that for "pure" free-Lagrangian methods discrete continuity equation is not consistent (order of approximation is zero) with continuous continuity equation; it also means that the so-called Geometric Conservation Law (GCL), [53,87], which can affect stability of the method, [53], is not fulfilled. It usually implies that small changes in the particle positions (especially if two particles are very close to each other) can lead to very large change of the volume. Non-Lagrangian behavior of Voronoi cell dramatically decreases accuracy near contact discontinuity. There are several approaches which intend to mitigate non-Lagrangian behavior of Voronoi cells. For example, in [95,114] special type of artificial viscosity was introduced which helps to regularize the particle positions and do not allow particles to be to close to each other. In [10] some corrections to velocity of generators are introduced and in [67] special artificial forces are introduced to correct shape of parcels. Another approach is to keep Voronoi mesh inducing connectivity, but define parcel shape using so-called median mesh [107], where the coordinates of vertices are defined by simple interpolation of a particle position and its two consecutive neighbors. It leads to non-convex shape of parcels and abrupt change in parcel volume when the connectivity of the mesh is changing (in contrast volume and shape of Voronoi cells are continuous in time, but are not Lagrangian objects). This clearly contradicts another important property of Lagrangian parcel: its shape is supposed to change continuously in time including instances when connectivity changes. Because of abrupt change of volume of the parcel some remapping is needed. The conclusion of this analysis is that even pure-Lagrangian method are neither "pure" nor "Lagrangian". Still as demonstrated, for example in [10,67,12,68,11,114] these types of methods can be very successful for particular problems, but each problem requires some special strategy to mitigate non-Lagrangian nature of Voronoi cells.

There were several other "free-Lagrangian" methods, which differ in how connectivity of the mesh is established, spatial centering of flow parameters (particle centered, parcel centered, staggered), what form of equations are used, what type of discretization is used, implicit or explicit, some of the methods having explicit remap phase, and so on. Interested reader can refer to the following papers [51,39,50,115,35,36,63,126,6,97,13,5,56]. Very educational are the roundtable discussions published in [55,121]. The conclusion is that these "free-Lagrangian" methods are even less Lagrangian than "pure" free-Lagrangian methods, which we have analyzed in previous paragraph. However, all of these methods allow change of connectivity of the mesh with time.

Our belief is that because free-Lagrangian methods are not really Lagrangian then, explicitly or implicitly, they incorporate a rezone phase, and consequently, a corresponding remap phase. Some of them explicitly states this remap phase. Nevertheless if the free-Lagrange method does not have a remap phase, errors related to it will manifest itself one way or another.

It leads us to the conclusion that methods where connectivity of the mesh can change have to be developed in reconnection-based ALE (ReALE) framework, where rezone stage includes reconnection. Let us note that similar philosophy was used in [45,1,37], even so authors of [45,1,37] do not call their method ALE or free-Lagrange.

Let us finally note that when we were writing this paper we became aware of the paper [119]. In this paper the author proposed a new formulation of continuum hydrodynamics based on an unstructured grid. In this work the mesh is defined as the Voronoi tessellation of a set of discrete mesh-generating points. The gas dynamics equations are solved using a Godunov-type finite-volume scheme on a moving Voronoi mesh. This approach corresponds to a direct ALE strategy (no explicit Lagrangian, rezone and remap phases-governing equations are written in moving coordinate system) in which the motion of the grid is ruled by prescribing the velocities of the mesh generators. The numerical results obtained with this method are quite impressive and display its robustness and accuracy. However, we note that this method may suffer from several limitations. First, this is by construction a mono-material formulation since the author is using a direct ALE strategy. We believe that its multi-material extension is far from being obvious. We also note that this ALE scheme does not fulfill GCL requirement. This flaw can lead to severe problems.

As standard for ALE method, the main elements in ReALE simulation are an explicit Lagrangian phase in which the solution and grid are updated (without changing connectivity), a rezoning phase in which a new grid is defined (which includes changing connectivity and also adding or deleting cells or vertices of the parcels), and a remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) onto the new grid. Flowchart of the entire ReALE algorithm is presented in Section 5. For ReALE all three phases are supposed to satisfy specific requirements which are different from standard ALE methods.

We assume, that at the beginning of the calculation ($t = 0$) as well at the beginning of each time step (after rezone phase) the computational mesh consists of Voronoi cells corresponding to some set of particles (generators, sites), that is, distribution of generators defines the mesh. Initialization of the mesh as well as necessary definitions related to Voronoi diagrams are given in Sections 3 and 4.

Because of reconnection in rezone phase, the Lagrangian phase of the ReALE method has to deal with discretization of the Lagrangian equations on general polygonal meshes and corresponding update of this polygonal mesh is supposed to be Lagrangian. There are several papers dealing with discretization of Lagrangian equation on general polygonal meshes [25,27,26,45,116,117]. In this paper we will use compatible mimetic finite discretizations [28,29] on staggered mesh, which historically close to [25,27,26] and newly developed cell-centered discretizations based on Godunov approach [91,89,87], which satisfy GCL (contrarily to [45], which does not satisfy GCL). Necessary details about discretizations used in Lagrangian phase of our ReALE methodology are presented in Section 6.

The rezone phase of ReALE has to include both mesh movement and reconnection procedure. In this paper we used a set of particles (generators) and the machinery of Voronoi diagrams to do both mesh movement and mesh reconnection. That is our rezone strategy consists of a special movement of generators. It is close to Lagrangian in some sense, but also include some smoothing procedure based on notion of centroidal Voronoi diagrams [43]. The rezone phase of our new ReALE approach is described in Section 7.

In the remapping phase, the Lagrangian solution is transferred (conservatively interpolated) onto the rezoned mesh. Lagrangian mesh is the result of one time step Lagrangian movement of the Voronoi mesh corresponding to the distribution of the generators at the previous time step. The new rezoned mesh is the Voronoi mesh corresponding to the positions of generators created by the rezone phase. During the rezone phase generators are moved in an "almost" Lagrangian way and because Voronoi cells are changing their shape continuously with respect to positions of the generators; rezoned and Lagrangian meshes are "close". However, in general, the connectivity of the Lagrangian and rezoned mesh are different. Consequently remapping methods have to be able to conservatively transfer flow parameters from one polygonal mesh to another. In this paper intersection (overlay) based remap is used [61,45,94,77], however, one can take advantage of how Lagrangian and rezoned meshes are constructed and design more efficient methods [76]. The remapping phase is described in Section 8.

We demonstrate the performance of our method on a set of numerical tests presented in Section 9. Finally, conclusions and future work are summarized in Section 10.

2. Motivation

To motivate our research let us consider the Rayleigh–Taylor instability problem. It consists of two ideal gases with initial densities $\rho_{top} = 2$ and $\rho_{bottom} = 1$; in both cases the adiabatic constant is $\gamma = 1.4$. Initially, the heavier gas is above the lighter gas in a rectangular vessel $[0{:}1/6] \times [0{:}1]$, with gravitational field directed vertically downward and with magnitude $g = 0.1$. The interface has been deliberately perturbed as described by formula $y_i(x) = 1/2 + 0.01\cos(6\pi x)$. Initially both gases are at rest; the pressure distribution is approximately hydrostatic and is defined in the lighter gas

$$P = 1 + \rho_{top} g 0.5 + \rho_{bottom} g(0.5 - y),$$

and in the heavier as

$$P = 1 + \rho_{top} g(1 - y).$$

It is well known that such configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. Further details of the general theory of Rayleigh–Taylor instabilities can be found in [78]. The time evolution of this problem leads to a rollup of the interface and the generation of significant vorticity. This problem is poorly suited for Lagrangian methods, and is usually tackled using Eulerian or ALE techniques. As discussed in Section 1, for standard ALE methods the mesh does not change connectivity. Improvement of the mesh on the rezone stage is achieved only by moving nodes. Here we consider a standard ALE approach based on cell-centered Lagrangian method described in [90,87], which uses Winslow approach at rezone phase [128]. In Fig. 1(a) we present meshes for time moments $t = 7, 8, 9$; in Fig. 1(b) we present vorticity color map at the same time moments. On Lagrangian stage of ALE the mesh is trying to follow the flow, but because of development vorticity it eventually leads to a tangled mesh. On rezone stage the mesh is slightly relaxed, however on the next time step of the Lagrangian phase, the mesh is trying to follow vorticity development and, again, is approaching a tangling situation. The meshes in Fig. 1(a) are the result of these competing processes.

In standard ALE method the mesh cannot change connectivity, and mesh with fixed connectivity has some limitation in how much it can deform. In Fig. 2 we present fragment of the mesh at $t = 8$ (panel a) as well as only "horizontal" (panel b)

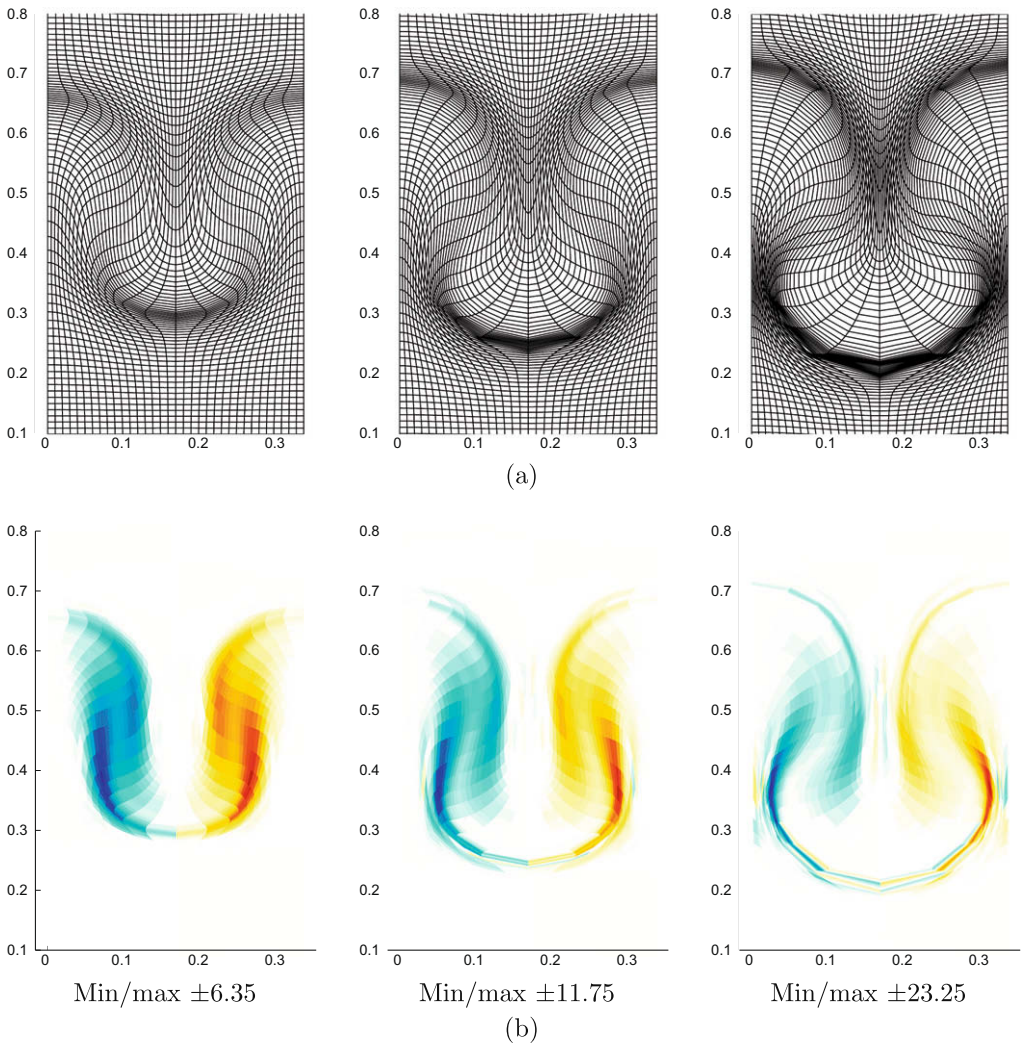

Fig. 1. Rayleigh–Taylor problem at time moments $t = 7, 8, 9$ – (a) Mesh fragments. (b) Vorticity color map. Color scale is from blue (minimal negative vorticity) to red (maximal positive vorticity). White color corresponds to zero vorticity. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

and only "vertical" (panel c) logical lines of the mesh. From this figure it is clear that logically rectangular mesh cannot deform much further in regions where vorticity is developing. At some time moment the mesh eventually stagnates in subregions where vorticity is still developing. Similar pictures can be found in [74] for ALE using staggered discretization and reference Jacobian rezone strategy.

This stagnation means that in subregions where vorticity is developing ALE method with fixed connectivity actually becomes Eulerian in its Lagrange + Remap form. It leads to excessive smoothing of flow parameters at remapping stage, and eventually to loss of accuracy in these regions. Moreover, because the mesh is logically rectangular, stagnation of the mesh in some regions leads to locking of the mesh in other regions, resulting to loss of overall accuracy. Also stagnated mesh has a low geometrical quality which additionally contributes to the loss of accuracy. One clearly sees that behavior of vorticity presented in Fig. 1(b) is not physical because it oscillates from positive to negative at neighboring cells, which is especially pronounced at $t = 9$. We believe that the resolution of this problem lies in allowing reconnection during the rezone stage of ALE method. However, allowing reconnection at rezone phase has its implications on all phases of ALE. First of all, we need to decide what is the mechanism of the reconnection. In this paper we use Voronoi tessellation machinery.

3. Voronoi tessellation

Let us consider a convex computational domain Ω in 2D. For a set of generators $\mathbf{G}_c = (x_c, y_c) \in \Omega$ the Voronoi cell Ω_c is defined as follows [96]:

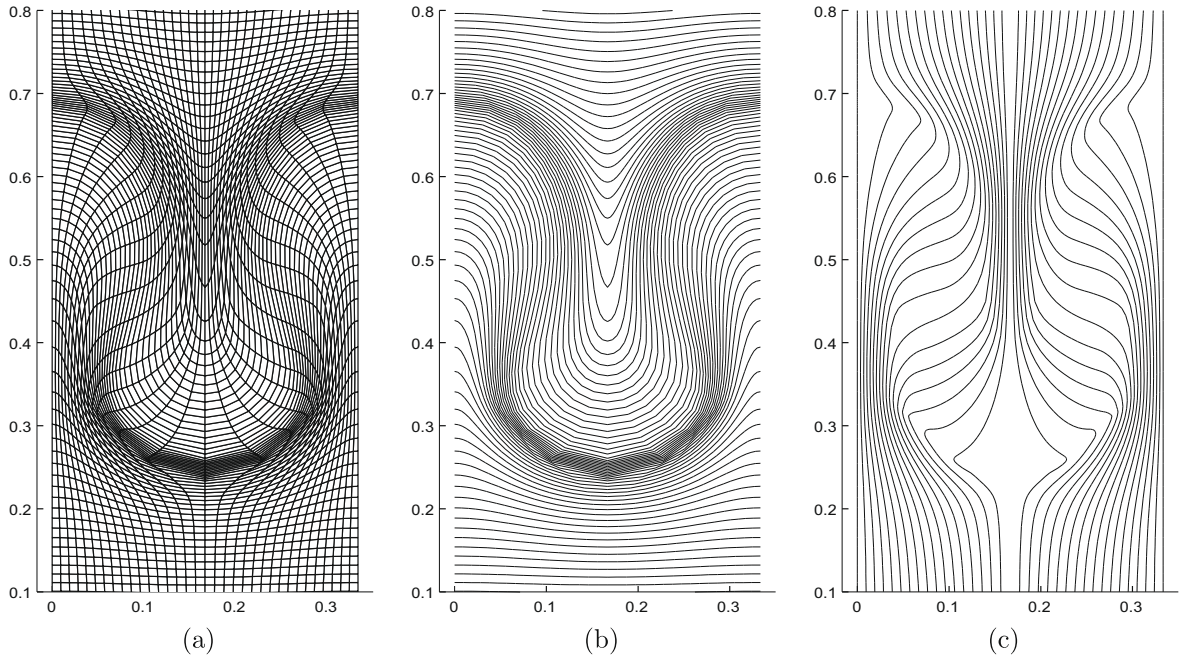

Fig. 2. Fragment of mesh at $t = 8$: (a) Mesh fragment. (b) "Horizontal" logical lines. (c) "Vertical" logical lines.

$$\Omega_c = \{\mathbf{X} = (x, y) \in \Omega : |\mathbf{X} - \mathbf{G}_c| \leqslant |\mathbf{X} - \mathbf{G}_{c'}|, \quad \text{for all } c \neq c'\} \tag{1}$$

Voronoi cell Ω_c is a convex polygon, and the set of Voronoi cells defines the tessellation of Ω, that is, it covers Ω without holes or overlaps. There are several generalizations of the definition of Voronoi cell for non-convex domains, like bounded and constrained Voronoi diagrams [7,8,72], VPS (visibility shortest path) Voronoi diagrams [96, pp. 163–156]. In this paper we will only consider convex computational domains and will not describe these generalizations. We use an incremental algorithm for the construction of Voronoi tessellation which is described in [113]. In general any other available algorithm can be used. In Fig. 3(a) we present Voronoi cells in the unit square which correspond to generators marked by $\times$. For now on, a cell Ω_c is referred to with its unique index c. By definition cell $\tilde{c}$ is a neighbor of cell c if it shares a face with it. The set of neighbors of cell c is denoted by $\mathcal{C}(c)$. The set of vertices of cell c is denoted by $\mathcal{P}(c)$. Any vertex of a Voronoi mesh is shared by three cells only. The set of faces of cell c is denotes as $\mathcal{F}(c)$, each face shares only by two cells. These relationships completely define the connectivity of the mesh.

Depending on the position of generators, the Voronoi mesh can be genuinely non-uniform. One of the possible measures of non-uniformity is how far is the centroid of a Voronoi cell from the generator corresponding to this cell. Let us introduce, $\boldsymbol{X}_c$, the centroid of the cell Ω_c as follows:

$$\boldsymbol{X}_c = \frac{1}{|\Omega_c|} \int_{\Omega_c} \boldsymbol{X}\, dV,$$

where $|\Omega_c|$ denotes the volume of the cell Ω_c.

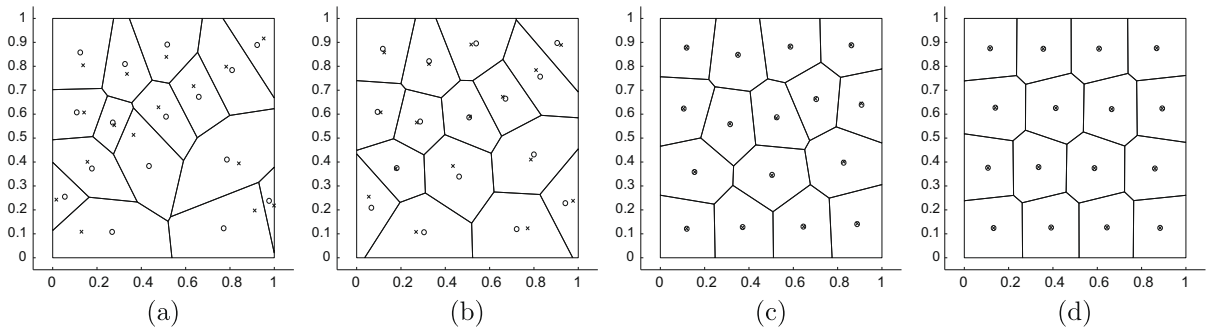

Fig. 3. Lloyd's algorithm to smooth Voronoi mesh: (a) Initial mesh – (b) mesh after one iteration – (c) mesh after 10 iterations – (d) final "converged" mesh – Crosses $\times$ correspond to positions of the generators. A circle $\circ$ is the position of the centroid of the Voronoi cell which corresponds to a unique generator.

In all panels of Fig. 3 positions of centroids are marked by circles ∘ and position of generators are marked by crosses ×. Clearly, the closer the centroids and the generators are, the more uniform the mesh is. It brings us to an important new definition: centroidal Voronoi tessellation (CVT) [43]. Voronoi tessellation is called centroidal Voronoi tessellation if the position of the cell centroid coincides with the position of the corresponding generator. CVT type of meshes are asymptotically (as number of generators goes to infinity) made of perfect hexagons inside the computational domain. Meshes of CVT type have several attractive properties, in particular discretizations of partial differential equations are usually more accurate on such meshes [69].

There is a very simple algorithm to create CVT depicted in Fig. 3. It starts with an arbitrary distribution of generators, Fig. 3(a), and constructs its corresponding Voronoi tessellation. Then it computes centroids of the constructed Voronoi cells and uses them as generators for a next iteration. The resulting mesh after one iteration is presented in Fig. 3(b). Let us note that regularity of the mesh is visibly improved. The mesh after 10 iterations is presented in Fig. 3(c). Finally the converged mesh is presented in Fig. 3(d). This algorithm is called Lloyd's method, readers can refer to [43,42] for more details.

For the purposes of our paper one iteration of Lloyd's algorithm can be considered as a mesh smoothing step, which, in some sense, is analogous to a Winslow iteration for meshes with fixed connectivity. However, in Lloyd's algorithm connectivity of the mesh may change at each iteration.

4. Initialization

Modeling of any problem starts with the creation of an initial mesh. In our approach, an initial mesh is created by the distribution of generators and the construction of the associated Voronoi mesh. As with any ALE method the initial mesh reflects the knowledge about the underlying physical problem, for example, initial distribution of materials, direction and shape of the main shocks and other important features of the flow. Let us repeat that the initial mesh has to be Voronoi mesh and has to be consistent with rezone strategy. "Consistent" means that the initial mesh has to be constructed in such a way that it will not change dramatically at rezone stage. This issue will be clarified in Sections 7 and 9. Another important issue is that, as for any ALE method, we want to track material boundaries as close as possible. Therefore, initially, we are trying to create a mesh, the cell faces of which coincide with material boundaries. Let us demonstrate some ideas on a simple example of a bubble containing one material in a rectangular computational domain filled with another material, see Fig. 4 (this is a fragment of the initial mesh for the shock–bubble interaction problem described in details in Section 9.3). For this problem it is important to have an orthogonal mesh with mesh lines aligned with coordinate directions outside the bubble because a vertical shock will approach the bubble from the right. As a consequence the shock direction will be aligned with the mesh. A square mesh is the degenerate Voronoi mesh obtained with a regular distribution of generators; these are located at the centers of the squares. In order for the boundary of the bubble to be represented by faces of Voronoi cells, the generators have to be located on the same distance from the boundary in orthogonal direction. In this case because the boundary is a circle of some radius r, generators are located also on concentric circles of smaller, $(r - \Delta r)$, an bigger, $(r + \Delta r)$ radii. Inside the bubble, generators are located on circles, which radii are equally distributed between 0 and r. On each circle we put a different number of generators such that the arc length between generators on each circle is approximately the same for all circles. It then creates a pseudo polar mesh. This mesh is a little bit extended outside the bubble. Technically to combine a square mesh and a pseudo polar mesh we first create generators which produce the square mesh, then generators which are located in disk of radius $r + 2\Delta r$ are removed and replaced by generators corresponding to the pseudo polar mesh. The resulting mesh is presented in Fig 4.

Similar techniques can be applied for fitting more complicated shapes, some ideas are presented in [102].

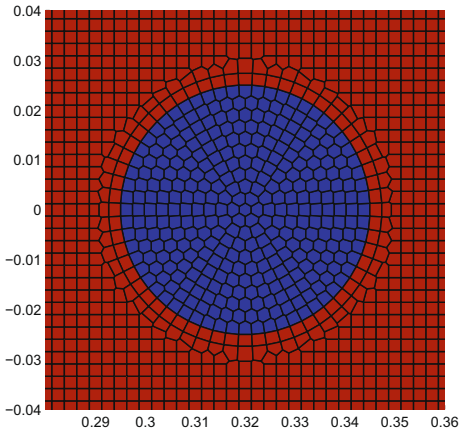

Fig. 4. Mesh for a bubble in a rectangle.

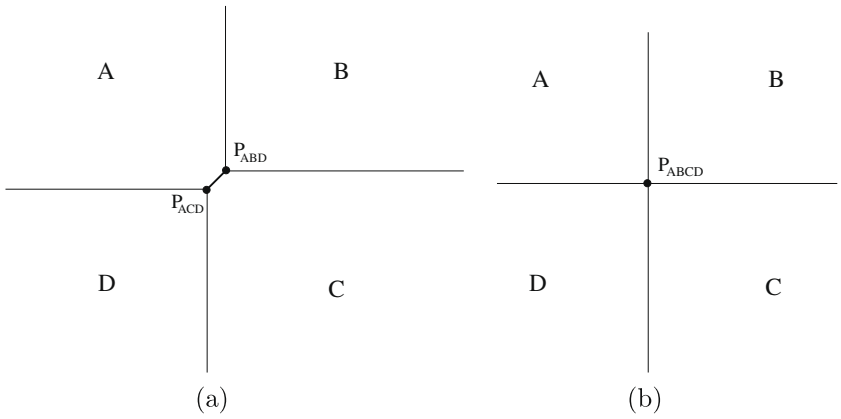

Fig. 5. Cleaning of small edges: (a) before cleaning and (b) after cleaning.

In some cases the faces of a Voronoi cell can be very small. In degenerate cases like a rectangular Voronoi mesh, some edges actually have zero length. This situation is schematically presented in Fig. 5(a). In this figure the edge P_{ACD} P_{ABC} has an almost zero length. Time step control for Lagrangian phase of ReALE method is based on minimal edge length over the cell, see Section 6. Therefore, the presence of small edges can dramatically reduce the time step. The solution lays in the introduction of a "cleaning" step that removes edges of the cell, which are small in comparison with a local characteristic length.

In situation presented in Fig. 5 cell A has neighbors $\ldots B, C, D \ldots$. Cells D and B are not a neighbors. Removing edge P_{ACD} P_{ABC} is equivalent to collapsing the two points into one point P_{ABCD} (the midpoint of segment $[P_{ACD}, P_{ABC}]$) schematically presented in Fig. 5(b). It means that point P_{ABCD} is now shared by four cells A, B, C, D. Cells A and C do not share any edge, and cells A and C do not have zero length edge anymore. We call this process small-edge cleaning. Cleaning is applied not only to zero length edges but to all edges which are small in comparison with the local characteristic length. More details on cleaning is given in Section 7. It is important to note that after the cleaning step, the computational mesh is not a Voronoi tessellation anymore. In particular each point can be shared by more than three cells.

5. Flowchart of ReALE method

In Fig. 6 we describe the flowchart of ReALE method. Initialization stage is described in previous section. The result of this stage is the mesh $I^{n=0}$, which, in general, is an unstructured polygonal mesh obtained by the cleaning of a Voronoi mesh. On the initialization stage we also define the initial condition for the degrees of freedom $U^{n=0}$ related to each particular Lagrangian scheme, for example, for cell-centered discretization, these are density, velocity and pressure of the cell (any other variable being deduced from them). First, the Lagrangian step uses the mesh I^0. At time step n the Lagrangian scheme starts with the rezoned mesh obtained from the previous time step. On Lagrangian stage the mesh is moving with the flow. The result of the Lagrangian step is the Lagrangian mesh L^{n+1} and all physical quantities on this mesh – U_L^{n+1}. Recall that Lagrangian algorithm has to deal with general polygonal meshes. Let us note that in the Lagrangian stage the generators do not play any role. During the Lagrangian step the mesh does not change connectivity. This stage is described in Section 6.

On the rezone stage we define positions of generators $\mathbf{G}_c^{n+1}$ and construct the associated Voronoi tessellation. It is followed by a cleaning step. The result of the rezone phase is a general polygonal mesh R^{n+1}. In general, connectivity of mesh R^{n+1} is different from connectivity of mesh L^{n+1}. However, the number of cells is the same. Details of the rezone stage are provided in Section 7. For the remap stage we perform a conservative interpolation of flow parameters from Lagrangian mesh L^{n+1} onto rezoned mesh R^{n+1}. Because in general these two meshes are polygonal meshes with different connectivities, one needs to use an intersection (overlay) based remap method. The remapping stage is described in details in Section 8. Finally to start new time step we set I^{n+1} as being the rezoned mesh R^{n+1}. In other words the new Lagrangian step starts with the polygonal mesh obtained as result of rezoning from the previous time step and the physical variables U^{n+1} from the remap phase.

6. Lagrangian phase

In this section we present the discretization of the Lagrangian hydrodynamics over a general two-dimensional polygonal grid that will be the first phase of our ReALE algorithm. We present two discretizations, the first one is based on a staggered placement of the variables, whereas the second one is cell-centered. In Lagrangian hydrodynamics methods, a computational cell moves with the flow velocity. In practice, this means that the cell vertices move with a computed velocity, the cell faces being uniquely specified by the vertex positions. Thus, Lagrangian methods can capture contact discontinuity sharply in multi-material fluid flows. However, in the Lagrangian framework, one has to discretize not only the gas dynamics equations but

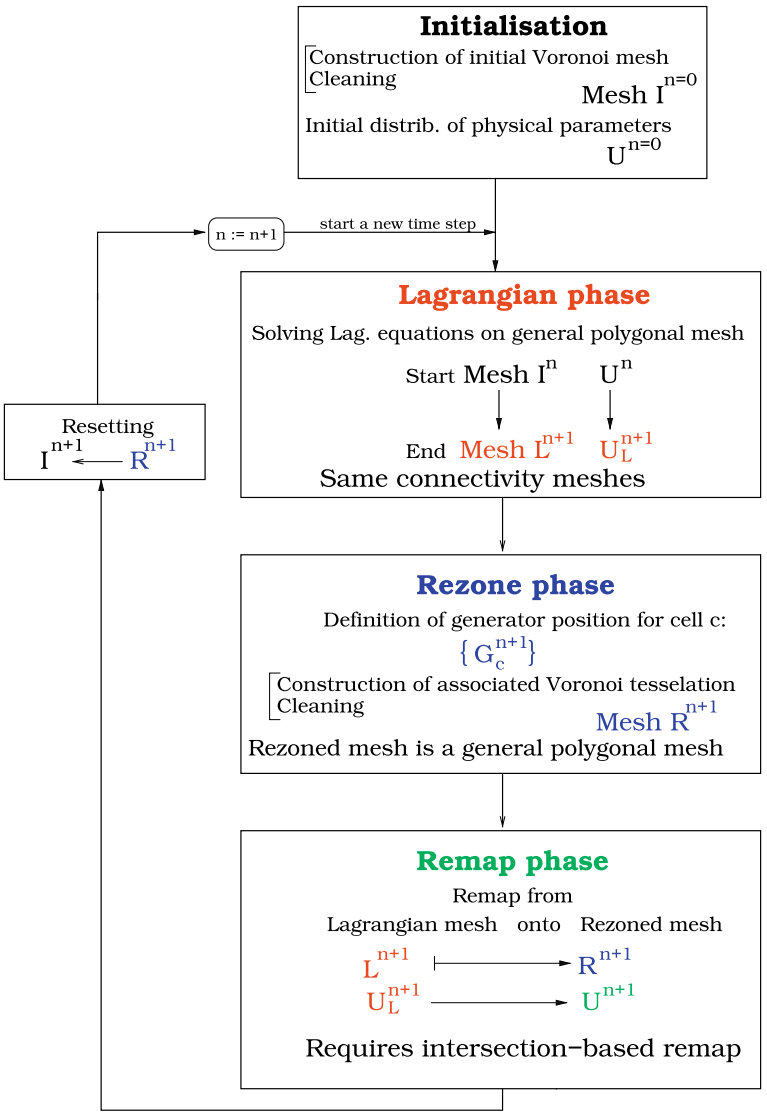

Fig. 6. Flowchart of ReALE Method.

also the vertex motion in order to move the mesh. Moreover, the numerical fluxes of the physical conservation laws must be determined in a compatible way with the vertex velocity so that the geometric conservation law (GCL) is satisfied, namely the rate of change of a Lagrangian volume has to be computed coherently with the node motion. This critical requirement is the cornerstone of any Lagrangian multi-dimensional scheme.

The most natural way to solve this problem employs a staggered discretization in which position, velocity and kinetic energy are centered at points, while density, pressure and internal energy are within cells. The dissipation of kinetic energy into internal energy through shock waves is ensured by an artificial viscosity term. Since the seminal works of von Neumann and Richtmyer [125], and Wilkins [127], many developments have been made in order to improve the accuracy and the robustness of staggered hydrodynamics [32,33,28]. More specifically, the construction of a compatible staggered discretization leads to a scheme that conserves total energy in a rigorous manner [30,29].

An alternative to the previous discretizations is to derive a Lagrangian scheme based on the Godunov method [60]. In the Godunov-type method approach, all conserved quantities, including momentum, and hence cell velocity are cell-centered. The cell-face quantities, including a face-normal component of the velocity, are available from the solution of an approximate Riemann problem at each cell face. However, it remains to determine the vertex velocity in order to move the mesh. In one of the first papers related to application of Godunov methods on general polygonal mesh in 2D [1], the flux computation was not compatible with the node displacement, and hence the GCL was not satisfied. This incompatibility generated additional spurious components in the vertex velocity field whose correction required a very expensive treatment [46]. An

important breakthrough concerning the compatibility between flux discretization and vertex velocity computation has been introduced in [40,88]. In these papers, authors present schemes in which the interface fluxes and the node velocity are computed coherently thanks to an approximate Riemann solver located at the nodes. This original approach leads to first-order conservative schemes which satisfy a local semi-discrete entropy inequality. The multi-dimensional high-order extension of these schemes are developed in [34,91,87,86].

In what follows, we recall briefly the main features of the two Lagrangian schemes, staggered and cell-centered, that will be used to construct our ReALE algorithm. The necessary details concerning the discretization can be found in previously published papers.

6.1. Governing equations

The gas dynamics equations, in Lagrangian form, write as follows:

$$\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla \cdot \boldsymbol{U} = 0, \tag{2a}$$

$$\rho \frac{d}{dt}\boldsymbol{U} + \nabla P = 0, \tag{2b}$$

$$\rho \frac{d}{dt}E + \nabla \cdot (P\boldsymbol{U}) = 0, \tag{2c}$$

where $\frac{d}{dt}$ is the material time derivative. Here, $\rho, P, \boldsymbol{U}$ and E denote the density, pressure, velocity and specific total energy of the fluid. The previous equations express the conservation of volume, momentum and total energy in a frame which moves with the fluid. The thermodynamic closure of this set of equations is obtained by the addition of an equation of state which is taken to be 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 for smooth flows, by subtracting kinetic energy equation from total energy equation, we get the time rate of change of specific internal energy

$$\rho \frac{d}{dt}\varepsilon + P\nabla \cdot \boldsymbol{U} = 0, \tag{3}$$

using (2a), this equation rewrites

$$\rho \frac{d}{dt}\varepsilon + P\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) = 0. \tag{4}$$

Recalling the Gibbs relation, $T\,dS = d\varepsilon + Pd(\frac{1}{\rho})$, where T denotes the temperature and S the specific entropy, it turns out that the previous internal energy Eq. (4) is equivalent to the conservation of entropy. We emphasize that this conclusion is valid only for smooth flows. The case of non-smooth flows, such as shock waves, is taken into account requiring that the second law of thermodynamics must be satisfied. To this end, we write the internal energy equation in the following non-conservative form:

$$\rho \frac{d}{dt}\varepsilon + P\rho \frac{d}{dt}\left(\frac{1}{\rho}\right) = \rho T \frac{d}{dt}S \geqslant 0. \tag{5}$$

This thermodynamic framework will be used in what follows to derive the staggered discretization.

Let us remark that Eq. (2a) is also named the Geometric Conservation Law (GCL) and is strongly linked to the motion of the fluid which is ruled by the trajectory equation

$$\frac{d\boldsymbol{X}}{dt} = \boldsymbol{U}(\boldsymbol{X}(t), t), \quad \boldsymbol{X}(0) = \boldsymbol{x}. \tag{6}$$

Here, $\boldsymbol{X}(t)$, denotes the position vector of a fluid particle at time $t > 0$, whose initial location was $\boldsymbol{x}$.

Here, we are deriving a discretization that is compatible with the GCL. By GCL compatibility, we mean that we are deriving a discrete divergence operator for the volume Eq. (2a) by requiring consistency of the divergence of the velocity field with the time rate of change of volume of the cell (cf. [92]). To highlight this particularly important point, it is instructive to rewrite Eq. (2a) in finite-volume form. To this end, let us consider a finite Lagrangian volume, $V_c(t)$, characterized by a constant mass, m_c, and a mass density defined as $\rho_c(t) = \frac{m_c}{V_c(t)}$. With this notation, one might substitute the following for Eq. (2a):

$$\frac{d}{dt}V_c - V_c \int_{\partial V_c(t)} \boldsymbol{U} \cdot \boldsymbol{N}\, dS = 0, \tag{7a}$$

$$\frac{d}{dt}m_c = 0, \tag{7b}$$

$$\rho_c(t) = \frac{m_c}{V_c(t)}, \tag{7c}$$

Eq. (7a) states that the time rate of change of the cell volume must be equal to the volume swept by the element boundary during its displacement with the flow velocity. One has to discretize it with great care so that the discrete GCL over a cell remains compatible with the discrete motion of the cell vertices. Namely, the time rate of change of a Lagrangian volume has to be computed coherently with the node motion. It means that change of volume computed from Eq. (7a) has to be the same as one corresponding to computing volume at new time step from geometry of the cell, that is, in discrete case from positions of vertices of the cell, which are advance in time according to trajectory Eq. (6).

Another way of expressing CGL is to require that discrete divergence operator **DIV** is defined consistently with the following formula

$$\mathbf{div}\,\mathbf{U} = \frac{1}{V}\frac{dV}{dt} \tag{8}$$

where V is volume of fluid parcel. In discrete case fluid parcel is represented by cell V_c and its volume depend on time not directly but by means of coordinates of vertices dependence on time. That is,

$$(\mathbf{DIV}\,\mathbf{U})_c = \frac{1}{V_c}\frac{dV_c}{dt} = \frac{1}{V_c}\sum_p \left(\frac{dV_c}{dx_p}\frac{dx_p}{dt} + \frac{dV_c}{dy_p}\frac{dy_p}{dt}\right) = \frac{1}{V_c}\sum_p \left(\frac{dV_c}{dx_p}u_p + \frac{dV_c}{dy_p}v_p\right), \tag{9}$$

where sum is taken over all vertices p of cell c, and x_p, y_p, u_p, v_p, coordinates and velocities of vertices correspondingly. One can find details in (cf. [92]). The essence of GCL is do discretize Eq. (7a) consistently with Eq. (9). In application to cell-centered discretization process of deriving GCL compatible discretization is described in detail in [88]. This is the cornerstone of any Lagrangian discretization.

6.2. Compatible staggered scheme

The staggered discretization used in this paper is based on a staggered placement of the variables. Namely, the kinematic variables, including the velocity, are located at the nodes while the thermodynamic variables (density, pressure and specific internal energy) are defined at the cell center. We note that this placement of the variables allows the staggered scheme to fulfill naturally the GCL compatibility requirement and at the same time to construct a discrete divergence operator.

The discretizations of momentum and specific internal energy are derived from each other by use of important concept of compatible discretization [28], which is based on detailed balance between kinetic and internal energy and uses subzonal masses and subzonal forces. This compatible hydrodynamics algorithm is thus designed to conserve momentum and total energy exactly in discrete form. The dissipation of kinetic energy into internal energy through shock waves is ensured by means of an artificial viscosity which can be edge based [33] or tensorial [28]. This mechanism leads to a dissipation that is coherent with the second law of thermodynamics. The subzonal pressure method is also used for the control of hourglass-type motion [32]. Finally, the time integration method is a predictor–corrector technique which is detailed in [30]. The extension of this compatible Lagrangian hydrodynamics algorithm to unstructured grids, where each zone is a polygon with an arbitrary number of sides, has been presented in [29].

6.3. Cell-centered scheme

This discretization employs a centered placement of the variables. That is density, pressure, momentum and total energy are piecewise constant over each cell. The interface fluxes and the nodal velocity are computed by means of a node-centered approximate Riemann solver. The resulting nodal velocity allows to calculate zone volumes in a consistent manner with their geometric definition. In this way, the GCL compatibility requirement is ensured [40,88]. The main new feature of the algorithm used here, is the introduction of four pressures on each edge, two for each node on each side of the edge [88]. This extra degree of freedom allows to construct a nodal solver which fulfills two properties. First, momentum and total energy are rigorously conserved at the discrete level. Second, a semi-discrete entropy inequality is provided, which shows that kinetic energy is correctly dissipated into internal energy through shock wave. The node based feature of this scheme makes it naturally unstructured and thus able to deal with polygonal meshes. The high-order extension is derived using a one-step time integrator, based on the Generalized Riemann Problem (GRP) methodology [87]. It consists in solving the high-order Riemann problem with piecewise linear polynomial, whereby the approximate solution is given as a time power series expansion right at the interface. The acoustic version of the GRP method has been implemented and extended to the framework of the two-dimensional node-centered Riemann solver. In this way, we get an acoustic node-centered Generalized Riemann solver which provides the time derivatives of the nodal velocity and pressures, needed for the high-order flux computation.

6.4. Time step control

The Lagrangian discretizations require a time step control to ensure the stability of the schemes. Let Δt^n denotes the current time step, the next time step, Δt^{n+1}, for both discretizations is evaluated using several criteria. The first one is a standard CFL criterion based on the characteristic time

$$\Delta t_{\text{cfl}} = C_{\text{cfl}} \min_c \frac{l_c^n}{a_c^{\star,n}},$$

where C_{cfl} is a strictly positive coefficient, l_c^n a characteristic zone length defined as the minimum of edge lengths in the cell. The generalized sound speed, $a_c^{\star,n}$, writes $a_c^{\star,n} = \sqrt{\frac{\gamma_c P_c^{\star,n}}{\rho_c^n}}$, where γ_c is the ratio of specific heats, ρ_c^n the mean zone density. The generalized pressure, $P_c^{\star,n}$, writes $P_c^{\star,n} = P_c^n$ for the cell-centered scheme and $P_c^{\star,n} = P_c^n + Q_c^n$ for the staggered scheme, where P_c^n denotes the mean zone pressure and Q_c^n the scalar part of the artificial viscosity tensors in a zone.

The second criterion ensures that a zone does not change its volume by too large an amount in a time step. To this end, we define the characteristic time

$$\Delta t_{\text{vol}} = C_{\text{vol}} \min_c \frac{1}{|(\nabla \cdot \boldsymbol{U})_c^n|},$$

where C_{vol} is a user defined coefficient and $(\nabla \cdot \boldsymbol{U})_c^n$ stands for the discrete divergence operator related to cell c.

Finally, the new value of the time step reads

$$\Delta t^{n+1} = \min\left(\Delta t_{\text{cfl}}, \Delta t_{\text{vol}}, C_{\text{mul}} \Delta t^n\right).$$

Here, C_{mul} is a multiplicative coefficient which does not allow time step to increase too fast. For numerical applications, we set $(C_{\text{cfl}}, C_{\text{vol}}, C_{\text{mul}}) = (0.25, 0.1, 1.05)$. Note that for the staggered discretization the new value of the time step is always chosen on the predictor step.

When we have to deal with polygonal cells containing small edges we need to supplement the previous time step control with a criterion that prevents the cells from being non-convex or tangling during the Lagrangian phase. To this end, let us consider a polygonal cell Ω_c and one of its vertex indexed by p. We also consider the previous and the next point of p in the counter-clockwise ordered list of points of Ω_c. We label them respectively p^- and p^+. The triangle formed with these points is denoted T_{pc}, its area writes

$$|T_{pc}| = \frac{1}{2}\left(\boldsymbol{X}_p\boldsymbol{X}_{p^+} \times \boldsymbol{X}_p\boldsymbol{X}_{p^-}\right) \cdot \boldsymbol{e}_z,$$

where $\boldsymbol{e}_z$ supplements the orthonormal basis $(\boldsymbol{e}_x, \boldsymbol{e}_y)$, i.e. $\boldsymbol{e}_z = \boldsymbol{e}_x \times \boldsymbol{e}_y$. It is well known that the cell Ω_c is convex provided that the area of T_{pc} is strictly positive for each point p of the cell. We will use this sufficient condition to predict an admissible time step. During the Lagrangian phase, the position vector of the points is updated according to

$$\boldsymbol{X}_p^{n+1} = \boldsymbol{X}_p^n + \Delta t \boldsymbol{U}_p^{n+\frac{1}{2}},$$

where $\boldsymbol{U}_p^{n+\frac{1}{2}}$ is the time-centered point velocity and Δt is the current time step. It turns out that the area of T_{pc}^{n+1} is a quadratic function of Δt. Thus, for each triangle T_{pc} we compute the strictly positive time step so that $T_{pc}^{n+1} > 0$. Next, we compute its minimum over the cell and the global minimum over the whole polygonal mesh to get the characteristic time Δt_{pol}. Finally, we modify the time step control as follows to take into account this new criterion

$$\Delta t^{n+1} = \min\left(\Delta t_{\text{cfl}}, \Delta t_{\text{vol}}, C_{\text{pol}} \Delta t_{\text{pol}}, C_{\text{mul}} \Delta t^n\right),$$

where the safety coefficient is set to $C_{\text{pol}} = 0.25$.

6.5. Multi-species thermodynamic closure

In this paragraph, we describe the multi-species thermodynamic closure model that we are using to obtain an effective equation of state for our multi-component fluid mixture. Every components are completely miscible from a continuum view point. Let us denote by the subscript f the fth component of the mixture. We suppose that each fluid follows a gamma gas law, namely its pressure, P_f, and specific internal energy, ε_f, write as function of temperature T_f

$$P_f = \frac{R}{\mathcal{M}_f}\rho_f T_f,$$
$$\varepsilon_f = \frac{R}{(\gamma_f - 1)\mathcal{M}_f} T_f,$$

where R denotes the perfect gas constant, γ_f the polytropic index of fluid f and $\mathcal{M}_f$ its molar mass. Each fluid is characterized by its mass fraction C_f, which represents the ratio between the mass of the fluid f and the total mass of the mixture. The mixture EOS closure problem requires to find the equilibrium mixture pressure, P, and temperature, T, such that

$$\frac{1}{\rho} = \sum_{f=1}^{F} \frac{C_f}{\rho_f}, \quad \text{volume conservation},$$

$$\varepsilon = \sum_{f=1}^{F} C_f \varepsilon_f, \quad \text{energy conservation},$$

$$P_f = P, \quad \forall f = 1 \cdots F, \quad \text{pressure equilibrium},$$

$$T_f = T, \quad \forall f = 1 \cdots F, \quad \text{temperature equilibrium},$$

where F denotes the total number of fluids, ρ, ε are the density and the specific internal energy of the mixture and ρ_f, ε_f the density and specific internal energy of fluid f. The solution of the previous set of equations allows to write the following effective mixture gamma gas law:

$$P = (\gamma - 1)\rho\varepsilon,$$

where γ is the effective polytropic index of the mixture, which writes

$$\gamma = 1 + \frac{\sum_{f=1}^{F} \frac{C_f}{\mathcal{M}_f}}{\sum_{f=1}^{F} \frac{C_f}{(\gamma_f - 1)\mathcal{M}_f}}. \tag{10}$$

During the Lagrangian phase, the concentration of each fluid evolves following the trivial equation $\frac{d}{dt} C_f = 0$.

7. Rezone phase

7.1. Generators displacement

Let Ω_c^n and Ω_c^{n+1} 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_c^n is denoted $\boldsymbol{G}_c^n$. In this section we define the new position of the generator at time t^{n+1}. First, we compute a Lagrangian-like displacement of the generator by setting

$$\boldsymbol{G}_c^{n+1,\text{lag}} = \boldsymbol{G}_c^n + \Delta t \boldsymbol{U}_c, \tag{11}$$

where $\boldsymbol{U}_c$ is the "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

$$\boldsymbol{U}_c = \frac{1}{|\mathcal{P}(c)|} \sum_{p \in \mathcal{P}(c)} \boldsymbol{U}_p^{n+\frac{1}{2}}.$$

Here, $\mathcal{P}(c)$ denotes the set of vertices of the Lagrangian cell Ω_c and $\boldsymbol{U}_p^{n+\frac{1}{2}}$ is the time-centered velocity of point p between times t^n and t^{n+1}.

Let us introduce, $\boldsymbol{X}_c^{n+1}$, the centroid of the Lagrangian cell Ω_c^{n+1}, according to

$$\boldsymbol{X}_c^{n+1} = \frac{1}{|\Omega_c^{n+1}|} \int_{\Omega_c^{n+1}} \boldsymbol{X}\, dV,$$

where $|\Omega_c^{n+1}|$ denotes the volume of the cell Ω_c^{n+1}. The updated position of the generator is defined by mean of a convex combination between the new Lagrangian-like position, $\boldsymbol{G}_c^{n+1,\text{lag}}$ and the centroid $\boldsymbol{X}_c^{n+1}$ of the Lagrangian cell at time t^{n+1}

$$\boldsymbol{G}_c^{n+1} = \boldsymbol{G}_c^{n+1,\text{lag}} + \omega_c \left(\boldsymbol{X}_c^{n+1} - \boldsymbol{G}_c^{n+1,\text{lag}} \right), \tag{12}$$

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_c^{n+1}. We note that for $\omega_c = 0$ we get a Lagrangian-like motion of the generator whereas for $\omega_c = 1$ we obtain a centroidal-like motion, which tends to produce a smoothed mesh. This latter case is equivalent to perform one Lloyd iteration [43,42]. It remains to determine ω_c.

7.2. Computation of the ω parameter

The first role of the ω parameter is to construct a convex combination between the Lagrangian grid and the centroidal Voronoi grid, that is why we want it to be in $[0,1]$. Moreover, we want it to vanish smoothly for rigid rotation and translation and recover the pure Lagrangian motion in these cases. In a nutshell, we want it to be Galilean invariant. To construct a parameter that fulfills the previous requirements we utilize mechanical objects that characterize properly the mechanical features of the flow.

A good candidate for this task is the right Cauchy–Green tensor, [22], evaluated between two consecutive Lagrangian time steps. It can be regarded as quantifying the ratio of squared lengths of infinitesimal fibers between the initial and the initial configurations. To be more precise, let $d\boldsymbol{X}_1 = L_1 \boldsymbol{N}_1$ denotes a material fiber of length L_1 in the initial configuration, where $\boldsymbol{N}_1$ is

a unit vector. This elementary fiber is stretched and rotated to $d\boldsymbol{X}_2 = L_2\boldsymbol{N}_2$, in the final configuration, where $\boldsymbol{N}_2$ is a unit vector. Introducing F as the deformation gradient between the initial and the final configuration, one has $d\boldsymbol{X}_2 = \mathsf{F}d\boldsymbol{X}_1$. Therefore, one can deduce that the length of the fiber in the final configuration is given by

$$L_2^2 = d\boldsymbol{X}_2 \cdot d\boldsymbol{X}_2 = L_1\mathsf{F}\boldsymbol{N}_1 \cdot L_1\mathsf{F}\boldsymbol{N}_1 = L_1^2\mathsf{F}^t\mathsf{F}\boldsymbol{N}_1 \cdot \boldsymbol{N}_1. \tag{13}$$

Hence, the ration of the squared lengths writes as

$$\left(\frac{L_2}{L_1}\right)^2 = \mathsf{C}\boldsymbol{N}_1 \cdot \boldsymbol{N}_1,$$

where $\mathsf{C} = \mathsf{F}^t\mathsf{F}$ is the Cauchy–Green tensor, which is symmetric definite positive and fulfills the requirements of Galilean invariance. Here, the subscripts 1 and 2 refer to the initial and final configurations corresponding to the beginning and the end of the Lagrangian time step.

Returning to our standard notations, we construct ω_c using invariants of the right Cauchy–Green strain tensor associated with deformation of 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 deformation gradient tensor F

$$\mathsf{F} = \frac{\partial \boldsymbol{X}^{n+1}}{\partial \boldsymbol{X}^n},$$

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 deformation gradient tensor is nothing but the Jacobian matrix of the map that connects the Lagrangian configurations of the flow at time t^n and t^{n+1}, in the two-dimensional case its components write

$$\mathsf{F} = \begin{pmatrix} \frac{\partial \boldsymbol{X}^{n+1}}{\partial \boldsymbol{X}^n} & \frac{\partial \boldsymbol{X}^{n+1}}{\partial Y^n} \\ \frac{\partial Y^{n+1}}{\partial \boldsymbol{X}^n} & \frac{\partial Y^{n+1}}{\partial Y^n} \end{pmatrix}.$$

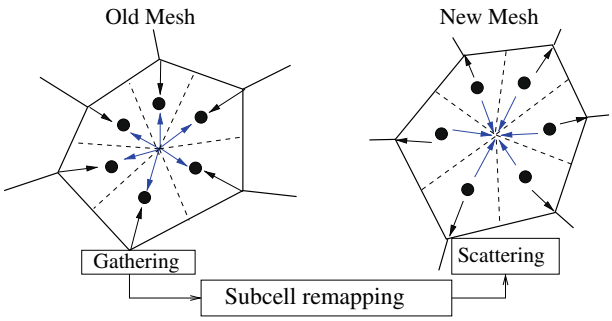

Fig. 7. Subcell remapping is performed in three phases – First, variables are gathered on subcells (left). Second a subcell-based conservative remapping is performed. Third, a scattering stage redistributes subcell-based conservative variables on nodes and cell centers (right).

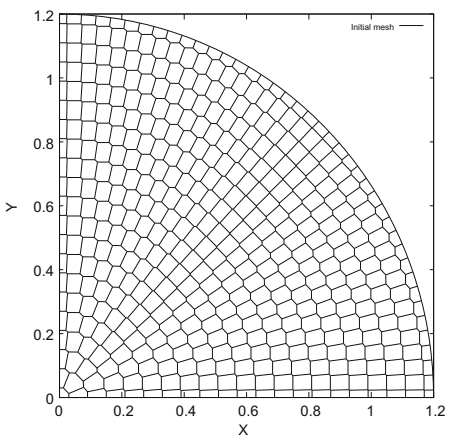

Fig. 8. Initial mesh for Sedov problem.

The right Cauchy–Green strain tensor, C, is obtained by right-multiplying F by its transpose, i.e.

$$C = F^t F.$$

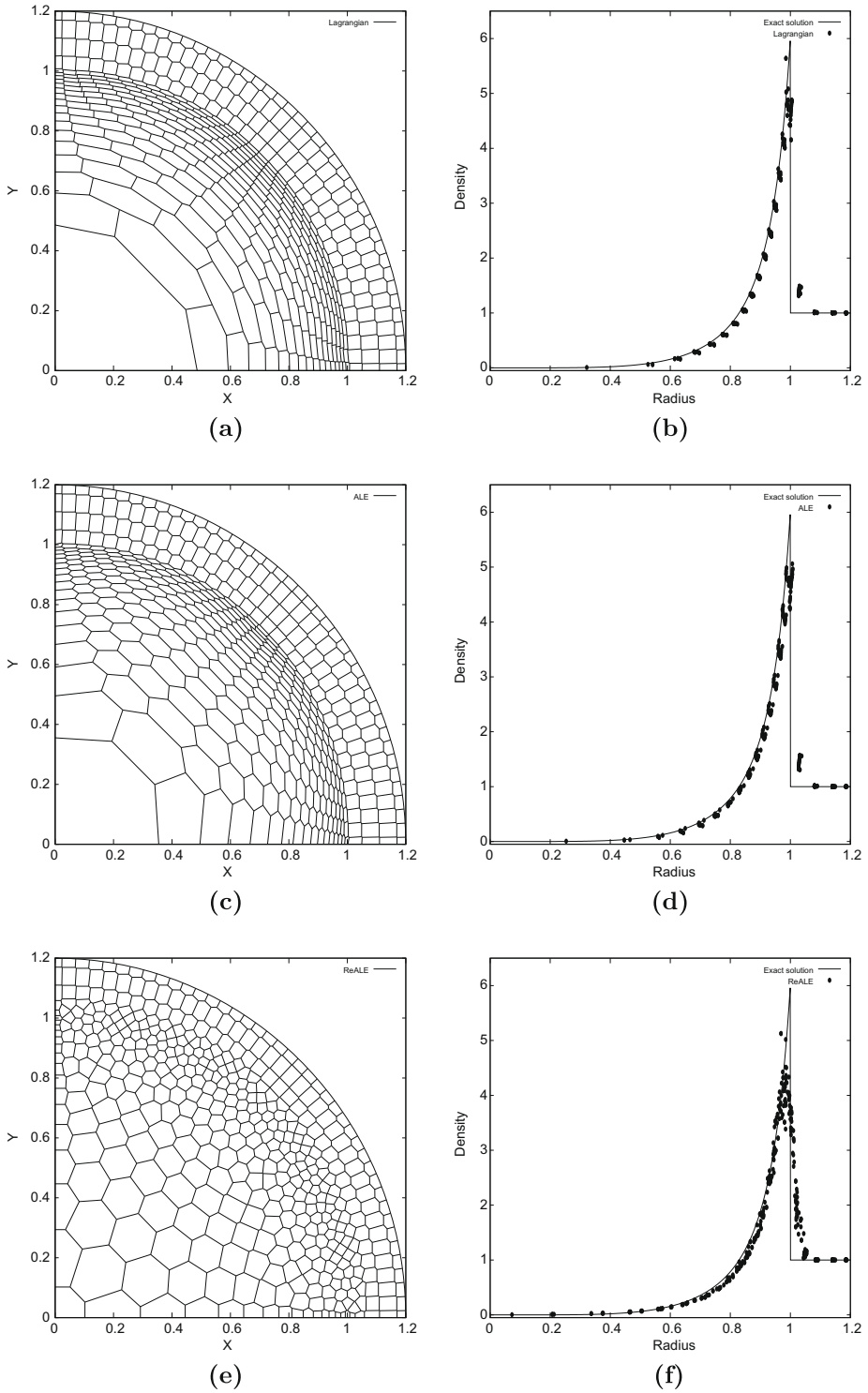

Fig. 9. Sedov problem at time $t = 1.0$ for Lagrangian, ALE and ReALE strategies – Cell-centered CHIC code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Lagrangian – (c) and (d) ALE – (e) and (f) ReALE.

In our case, C 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, which are denoted λ_1 and λ_2 with the convention $\lambda_1 \leqslant \lambda_2$. These eigenvalues 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-aver-

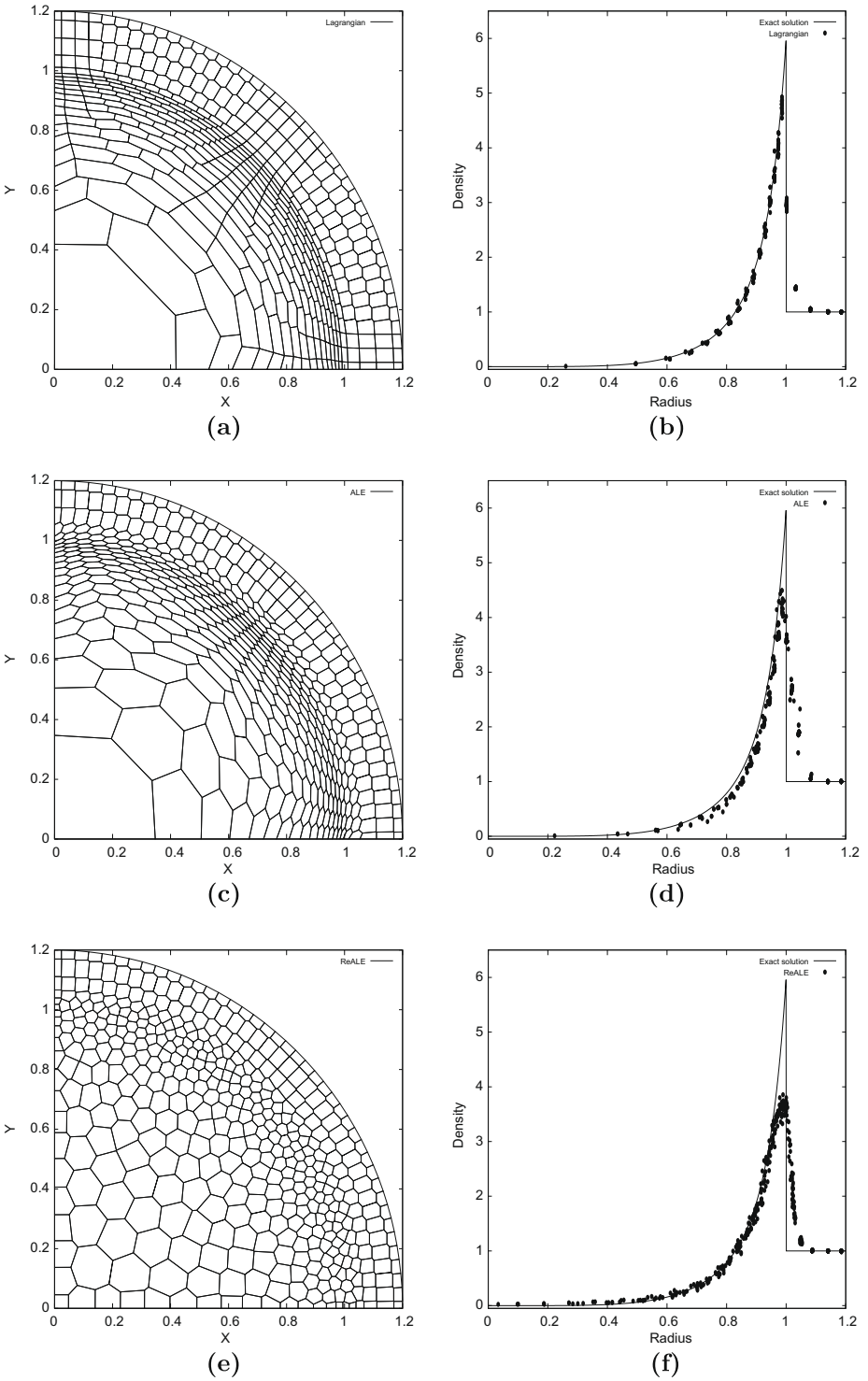

Fig. 10. Sedov problem at time $t = 1.0$ for Lagrangian, ALE and ReALE strategies – Staggered ALE INC. code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Lagrangian – (c) and (d) ALE – (e) and (f) ReALE.

aged value of the Cauchy–Green tensor by setting $C_c = F_c^t F_c$. Noticing that the two rows of the F matrix correspond to the gradient vectors of the X and Y coordinates, we can set $F^t = [\nabla_n X^{n+1}, \nabla_n Y^{n+1}]$, where for any functions

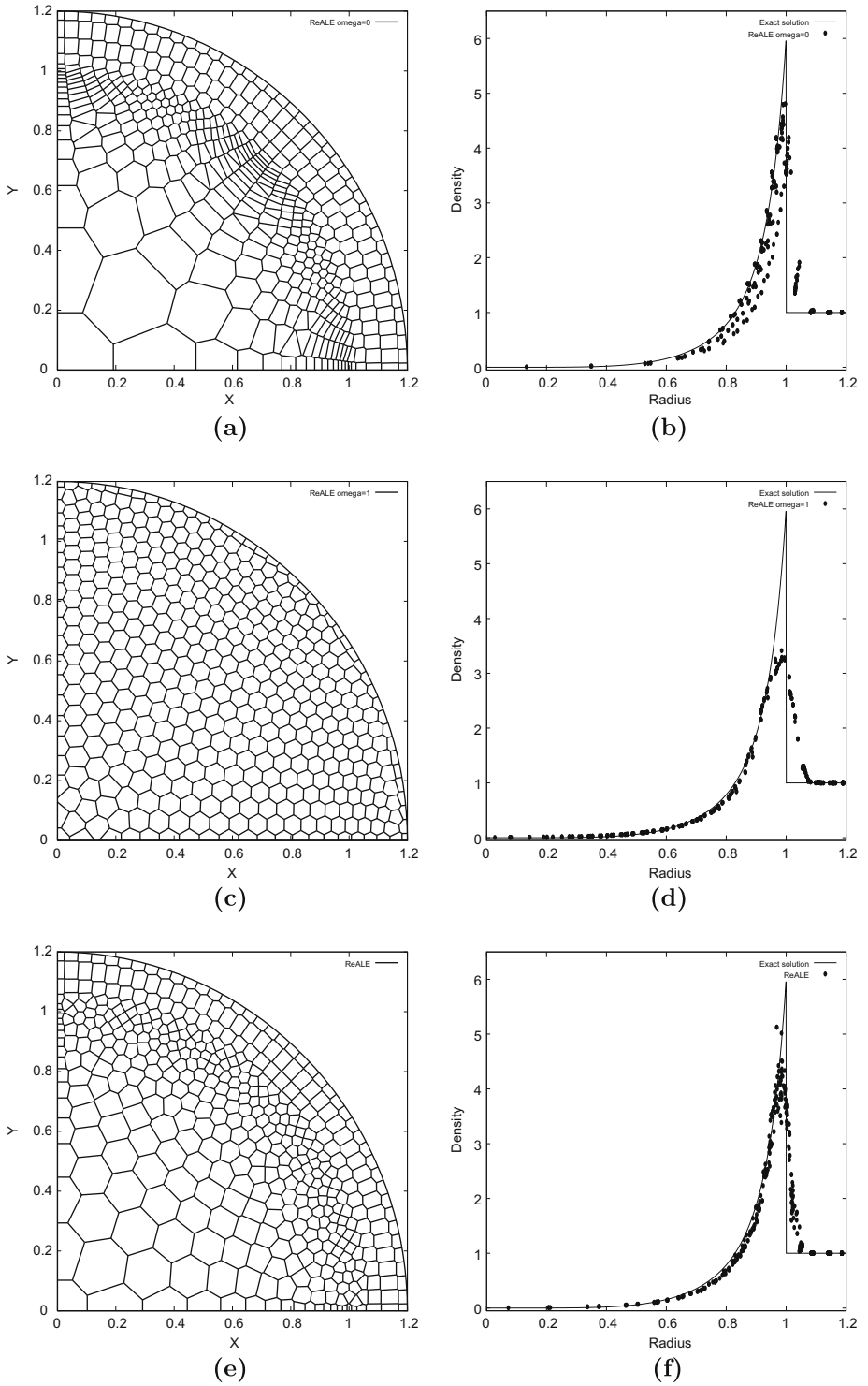

Fig. 11. Sedov problem at time $t = 1.0$ for different generator displacement strategies – Cell-centered CHIC code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Quasi-Lagrangian generator motion $\omega_c = 0$ – (c) and (d) Quasi-centroidal generator motion $\omega_c = 1$ – (e) and (f) Deformation-tensor based generator motion $\omega_c = f(\lambda_1, \lambda_2)$.

$\psi = \psi(X^n), \nabla_n \psi = (\frac{\partial \psi}{\partial X^n}, \frac{\partial \psi}{\partial Y^n})^t$. With these notations, let us define the cell-averaged value of the gradient of the ψ function over the Lagrangian cell Ω_c^n

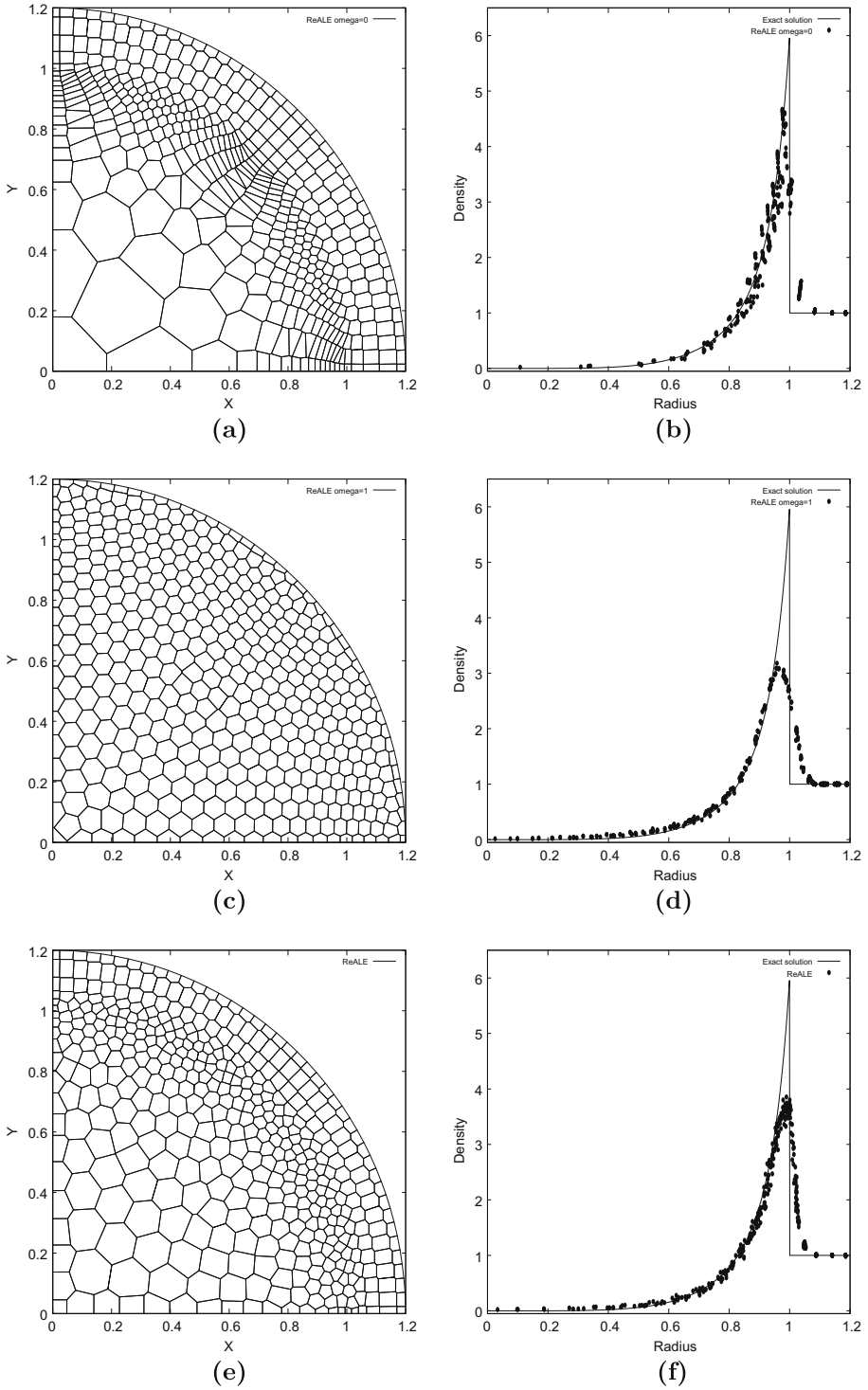

Fig. 12. Sedov problem at time $t = 1.0$ for different generator displacement strategies – Staggered ALE INC. code – Left column: Mesh. Right column: Density as a function of radius for all cells *vs.* the exact solution (line) – (a) and (b) Quasi-Lagrangian generator motion $\omega_c = 0$ – (c) and (d) Quasi-centroidal generator motion $\omega_c = 1$ – (e) and (f) Deformation-tensor based generator motion $\omega_c = f(\lambda_1, \lambda_2)$.

$$(\nabla_n\psi)_c = \frac{1}{|\Omega_c^n|}\int_{\Omega_c^n}\nabla_n\psi\, dV = \frac{1}{|\Omega_c^n|}\int_{\partial\Omega_c^n}\psi \mathsf{N}\, dS.$$

Here, we have used the Green formula and $\boldsymbol{N}$ is the unit outward normal to the boundary of the cell Ω_c^n referred as to $\partial\Omega_c^n$. Knowing that this cell is a polygon, we make use of the trapezoidal rule to obtain the following approximation for the previous integral

$$(\nabla_n\psi)_c = \frac{1}{|\Omega_c^n|}\sum_{p=1}^{|\mathcal{P}(c)|}\frac{1}{2}\left(\psi_p^n+\psi_{p+1}^n\right)L_{p,p+1}^n\mathsf{N}_{p,p+1}^n, \tag{14}$$

where ψ_p^n is the value of ψ evaluated at point X_p^n and $L_{p,p+1}^n\mathsf{N}_{p,p+1}^n$ is the outward normal to the edge $[\mathsf{X}_p^n,\mathsf{X}_p^{n+1}]$. Applying (14) to $\psi = X^{n+1}$ and $\psi = Y^{n+1}$ we get a cell-averaged expression of the gradient tensor F and then deduce from it the cell-averaged value of the right Cauchy–Green tensor C_c.

Knowing this symmetric positive definite tensor in each cell, we compute its real positive eigenvalues $\lambda_{1,c}$, $\lambda_{2,c}$. We finally define the parameter ω_c as follows:

$$\omega_c = \frac{1-\alpha_c}{1-\alpha_{\min}}, \tag{15}$$

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. Note that more complex formulae for ω_c are also possible, however we limit ourselves to the previous simple formula and postpone deeper investigations in this area for future papers.

In Section 9 we demonstrate sensitivity of choice of parameter ω_c on geometrical quality of the mesh and accuracy of ReALE calculations on the example of Sedov problem. We also present color map for values of the ω_c for 2D Riemann problem, which demonstrate its ability to follow main features of the flow. These results justify the choice of ω_c described by Eq. (15).

7.3. Cleaning

Once the new position of generators G_c^{n+1} are computed one constructs the corresponding Voronoi mesh. This mesh needs a last treatment as this Voronoi mesh may have arbitrary small faces (edges). Such faces can drastically and artificially reduce the time step, and, more important can lead to a lack of robustness. Consequently one defines a cutoff length $L_c^\varepsilon = \varepsilon L_c$, where L_c is a characteristic length of the cell and ε a small parameter. Any face f of cell c of length L_f smaller than the cutoff length L_c^ε, is removed from the Voronoi mesh , which lead to corresponding change in the connectivity. More specifically the vertices of such a face are merged (see Fig. 5 in Section 4); one vertex is then discarded from the vertex list and the connectivity structure. In our calculations we have chosen $\varepsilon = 0.01$ and $L_c = \frac{\sum_{f\in\mathcal{F}(c)}L_f}{|\mathcal{F}(c)|}$ being the average of face lengths of cell c with $\mathcal{F}(c)$ is the set of edges of cell c.

This "cleaned" polygonal mesh is no more of Voronoi kind but is well suited from a computational point of view.

In current code results of cleaning procedure depend on order in which cells are processed, that is, cleaning is order dependent. For unstable flows like Rayleigh–Taylor problem, it can lead to symmetry breaking, as one can see in Section 9.4.2.

One clearly can make cleaning procedure more symmetric and order "independent". For example, we can use Jacobi like cleaning procedure, when on the first pass one marks edges to be eliminated based on the minimal cut off length from two neighboring cells and then, on second pass, eliminates them all at the same time and makes corresponding changes in data structures.

One need to recognize that there is always be potential to developing non-symmetry in unstable flows like Rayleigh–Taylor example because of round off error. It can happen even without any cleaning.

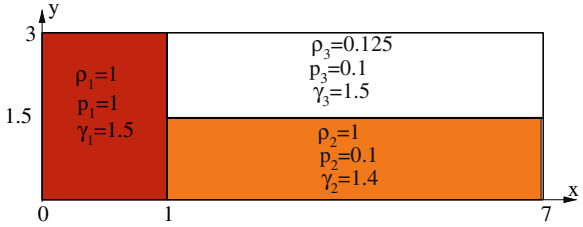

Fig. 13. Triple point problem.

8. Remap phase

The remapping phase consists of 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, moreover conservation of mass, momentum and total energy must be ensured, and, second-order accuracy conservative interpolation must be performed.

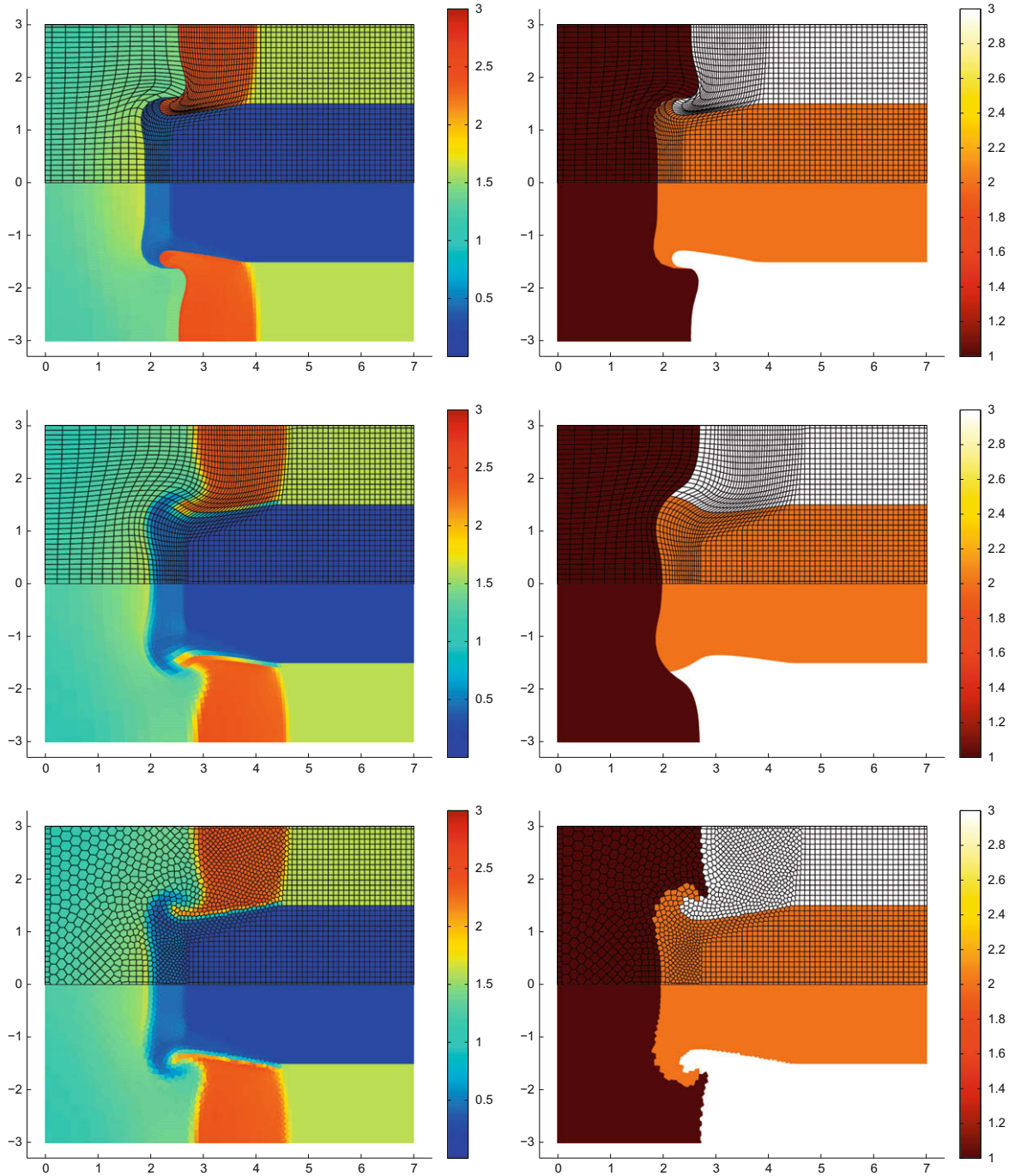

Fig. 14. Triple point problem at time t_{fail} for which Lagrangian scheme fails – Cell-centered CHIC code – Left column: Internal energy and mesh. Right column: Cell color corresponds to the initial domain (Ω_1, Ω_2 and Ω_3) – Top: Lagrangian method; Middle: ALE method; Bottom: ReALE method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

As the old (Lagrangian) and new (rezoned) polygonal meshes may not have the same connectivity, the remapping phase of our ReALE codes is based on exact intersection of *a priori* two different polygonal meshes.

Primary variables are cell-centered density, velocity and specific total energy for the cell-centered discretization whereas they are subcell density, nodal velocity and cell-centered specific internal energy for staggered discretization. Conservative quantities are cell-centered mass, momentum and total energy for the cell-centered approach whereas they are subcell mass, momentum and total energy for the staggered discretization.

8.1. Cell-centered based remap

If the primary variables are located at the same position, as it is the case for the cell-centered Lagrangian scheme, then the remapping phase is fairly simple [77,61,49]. The quantities on the old Lagrangian mesh are cell-centered density, velocity and total energy that must be transfered on the rezoned mesh. 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 [17] is performed to enforce physically justified bounds. Conservative quantities, namely mass, momentum and total energy, are obtained by integration of these representations. New conservative quantities are calculated by integration over polygons of intersection of new (rezoned) and old (Lagrangian) meshes. Finally, primary variables are simply recovered by division by new volume $\tilde{V}_c$ (for density) or new mass $\tilde{m}_c$ (for momentum and energy).

8.2. Subcell-centered based remap

Some difficulties arise when staggered location of variables is used, as for any staggered Lagrangian scheme. Conservative quantities are therefore not located on the same "entity"; mass is located at subcells, momentum at points. As a consequence, total energy is not properly defined at a given location. In [84] a gathering-remapping-scattering algorithm has been devel-

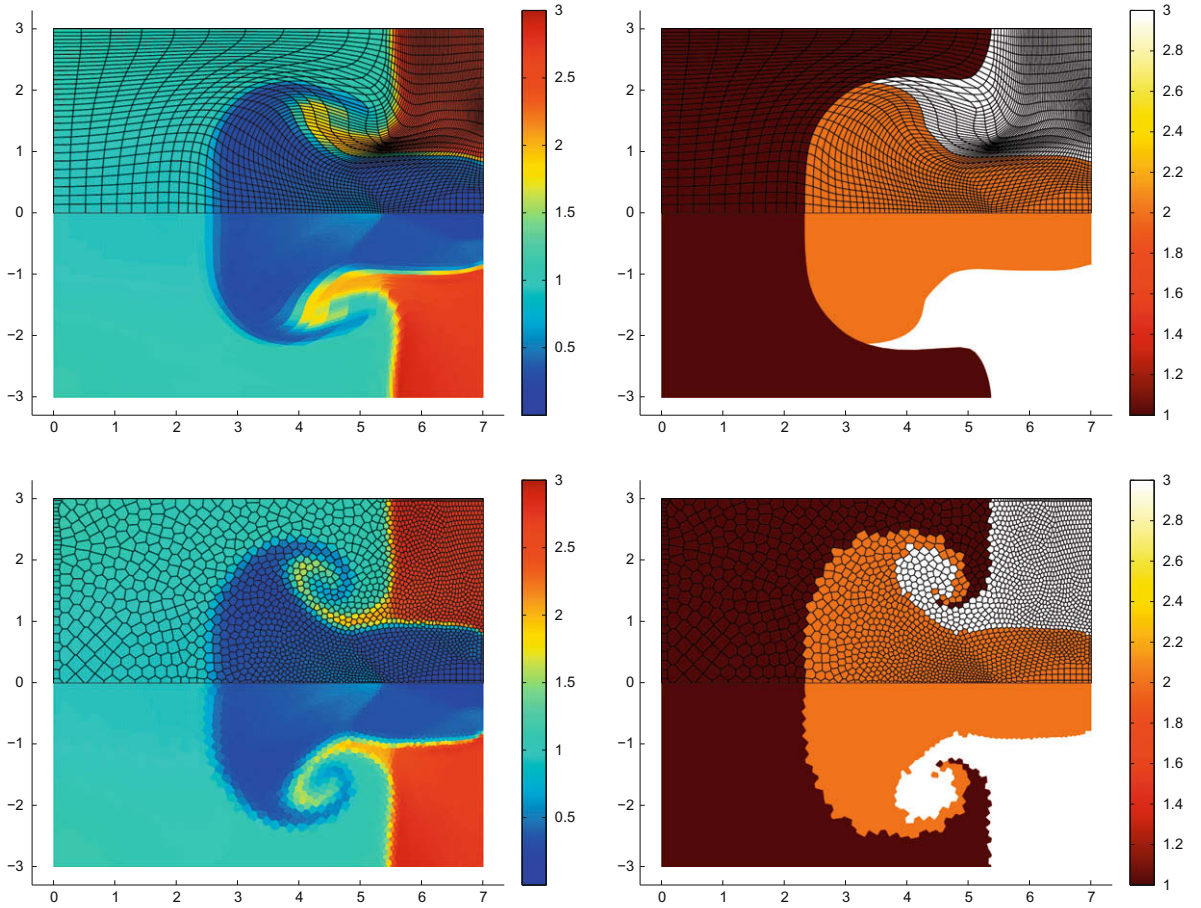

Fig. 15. Triple point problem at final time to observe stagnation of ALE mesh – Cell-centered CHIC code – Left column: Internal energy and mesh. Right column: Cell color corresponds to the initial domain (Ω_1,Ω_2 and Ω_3) – Top: ALE method; Bottom: Cell-centered (CHIC-based) ReALE method. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

oped (see Fig. 7 for a sketch); the main idea being to gather any conservative variable at subcell level, remap on subcell base similarly to the cell-centered remapper from the previous paragraph, and scatter back primary variables from subcells (density) to cell (internal energy) or node (velocity). More specifically this method consists in the following three stage algorithm:

- Gathering: Mass, momentum, internal and kinetic energies are defined on subcells from subcell density, nodal velocity and cell-centered specific internal energy in such a way that conservation is preserved.
- Subcell remapping: Conservative remapping from Lagrangian mesh subcells onto rezoned mesh subcells.
- Scattering: Conservative recovery of primary variables (subcell density, nodal velocity, cell-centered internal energy) on the rezoned mesh.

This remapping stage is followed by a repair technique to ensure physically justified bound preservation [77,112,85].

8.3. Remapping of concentrations

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 Ω_c^{n+1}, $m_{f,c} = \int_{\Omega_c^{n+1}} \rho C \, dV$. We note that $m_c = \sum_{f=1}^{F} m_{f,c}$ since $\sum_{f=1}^{F} C_{f,c} = 1$. Then, the mass of each fluid is interpolated conservatively onto the rezoned grid following the methodology previously described for the cell-centered quantities. We denote its new value by $\tilde{m}_{f,c}$. At this point we notice that $\tilde{m}_c \neq \sum_{f=1}^{F} \tilde{m}_{f,c}$, 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 $\widetilde{C}_{f,c} = \frac{\tilde{m}_c}{\tilde{m}_{f,c}}$ and impose the renormalization $\widetilde{C}_{f,c} \leftarrow \frac{\widetilde{C}_{f,c}}{\sum_{f=1}^{F} \widetilde{C}_{f,c}}$ so that $\sum_{f=1}^{F} \widetilde{C}_{f,c} = 1$. We point out that this renormalization does not affect the global mass conservation.

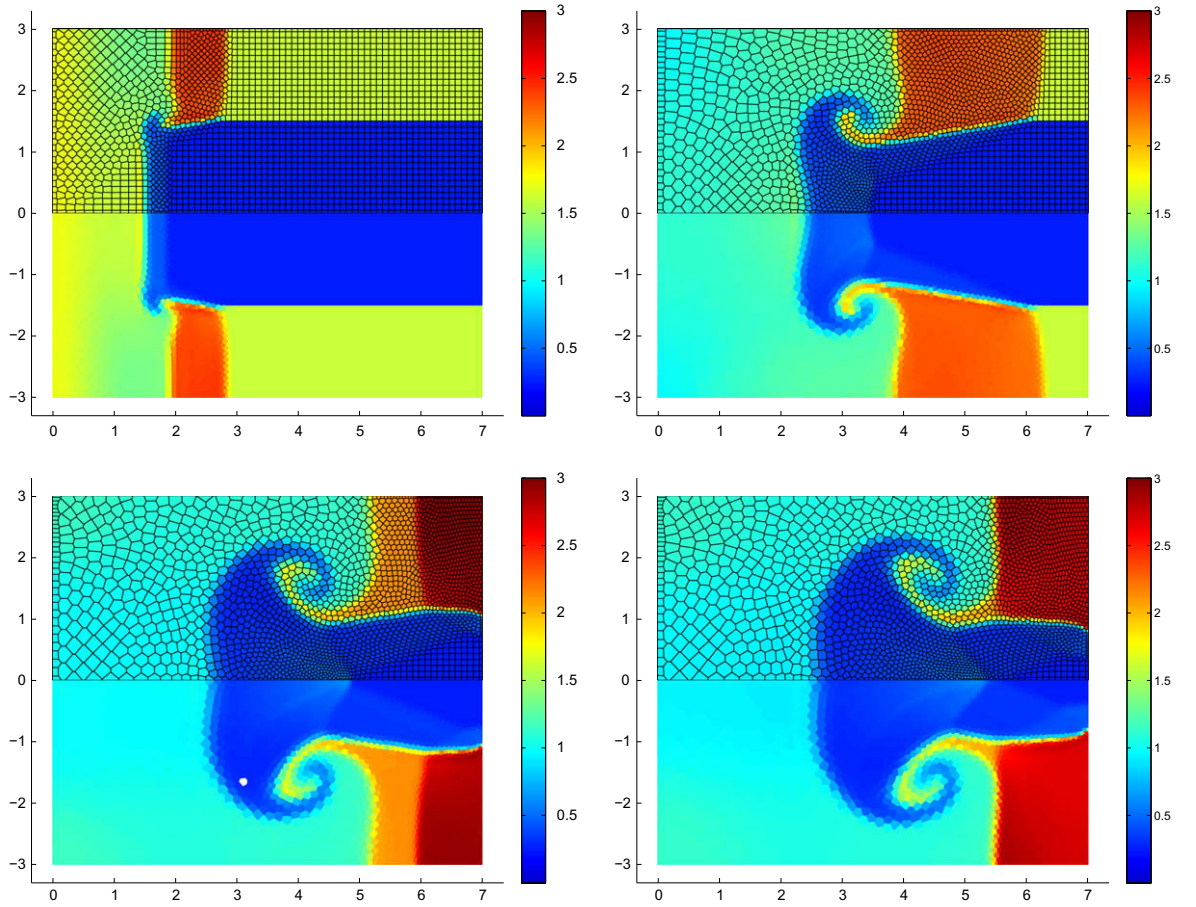

Fig. 16. Triple point problem at several times for ReALE – Cell-centered CHIC results for internal energy and mesh –From top-left to bottom-right: times $t = 1, 3, 4.5, 5$.

9. Numerical tests

In this section we present the numerical results obtained by the cell-centered ReALE code based on CHIC ALE code [87], and, the staggered ReALE code based on ALE INC(ubator) [82]. All calculations are performed in Cartesian geometry – (x,y).

The first test is the well-known Sedov test case in planar geometry, 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 "triple point" problem. It involves interaction of the shock with obstacle, which leads to vorticity formation. Most of the ReALE studies are performed using this problem. Third problem, shock interaction with a helium bubbler, is run in order to show the predictive capabilities of ReALE technique-we compare numerical results with experimental data. Finally, a Rayleigh–Taylor instability is run in order to assess the feasibility of capturing physical instability in an almost Lagrangian fashion. For this problem we compare our results with results obtained by the front tracking code (FronTier – [41,58,59]) and a an implicit large eddy simulation (ILES) incompressible Eulerian code (RTI3D – [4]).

All these tests, besides the Sedov test, generate high vorticity which is a classical cause of failure for Lagrangian schemes. For ALE codes with fixed connectivity it usually leads to a conflict between: a physics-based vortex-like motion with a tendency to tangle the mesh and, a geometrical-based opposite motion enforced by the rezoning to avoid bad geometric quality cells. Such a conflict leads to a stagnation of the mesh that reconnection is intended to cure.

9.1. Sedov problem

Let's consider the Sedov blast wave problem in Cartesian coordinates. This problem models an intense explosion in a perfect gas; it is an example of a diverging shock wave.

The computational domain is Ω, a quarter of a disk of radius 1.2 centered at the origin.

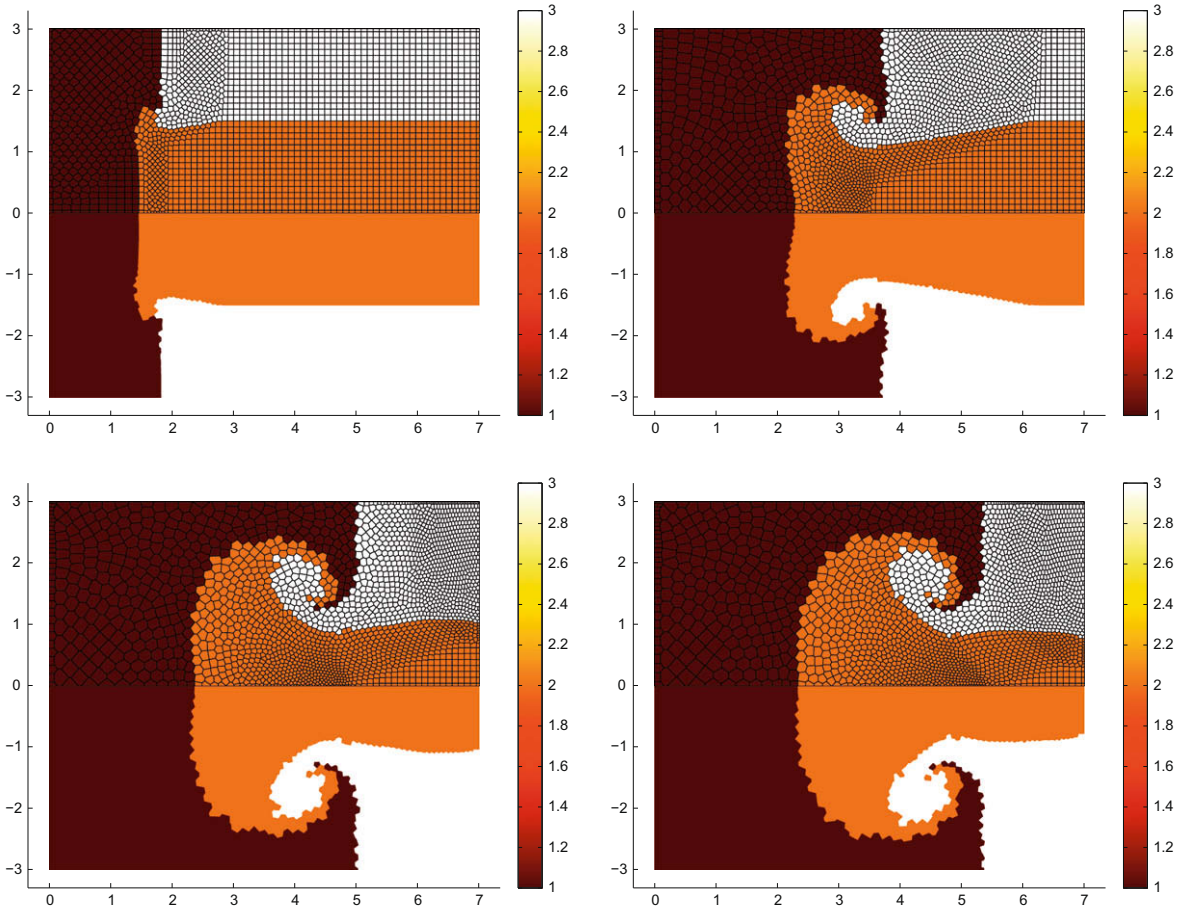

Fig. 17. Triple point problem at several times for ReALE – Cell-centered CHIC results – Cell color corresponds to the initial domain. From top-left to bottom-right: times $t = 1, 3, 4.5, 5$. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The initial conditions are characterized by $(\rho_0, P_0, \boldsymbol{U}_0) = (1, 10^{-6}, 0)$ for a perfect gas with polytropic index set to $\gamma = \frac{7}{5}$. We model an initial delta-function energy source at the origin by prescribing internal energy in the cell adjacent to the origin (see Fig. 8)

$$\varepsilon_{\text{or}} = \frac{\mathcal{E}_0}{V_{\text{or}}}, \tag{16}$$

where V_{or} denotes the volume of the cell and $\mathcal{E}_0$ is the total amount of released energy. For $\mathcal{E}_0 = 0.244816$, the front of a diverging shock is located at the radius $R = 1$ at the time $t = 1$. The peak density reaches the value 6. Symmetry boundary conditions are applied on the axis whereas zero velocity boundary condition is applied at radius 1.2.

The initial polygonal mesh is a Voronoi tessellation computed using 441 generators, see Fig. 8. The positions of generators are arranged similarly to how it described in Section 4 for meshing bubble, Fig. 4.

To run this test we do not need ALE, and *a fortiori* ReALE, technique; pure Lagrangian schemes usually perform well. However, we will present the Lagrangian, ALE and ReALE results (both for the cell-centered and staggered code) for the sake of comparison. Different generator motions are also compared.

9.1.1. Lagrangian, ALE and ReALE results

We present the meshes in Fig. 9 (resp. Fig. 10) panels (a)–(c)–(e) for the cell-centered CHIC-based methods (resp. for the staggered ALE INC.-based methods). The density for these methods is presented as a function of the cell radius for all cells in Figs. 9 and 10 panels (b), (d), (f); it is plotted against an exact solution (straight line). As known, Lagrangian schemes behave properly for the Sedov test case as seen on Figs. 9 and 10 panel (a) and (b). The final Lagrangian mesh presents expanded cells in the rarefaction wave and compressed ones after the shock wave. ALE techniques (panels (c) and (d)) can improve the smoothness of the mesh; some numerical diffusion is added during the remapping phase as can be seen on panels (d) of Figs. 9 and 10. ReALE technique used with deformation-tensor based generator motion (see panels (e) and (f)) is able to produce a smooth mesh and density profiles comparable with standard ALE.

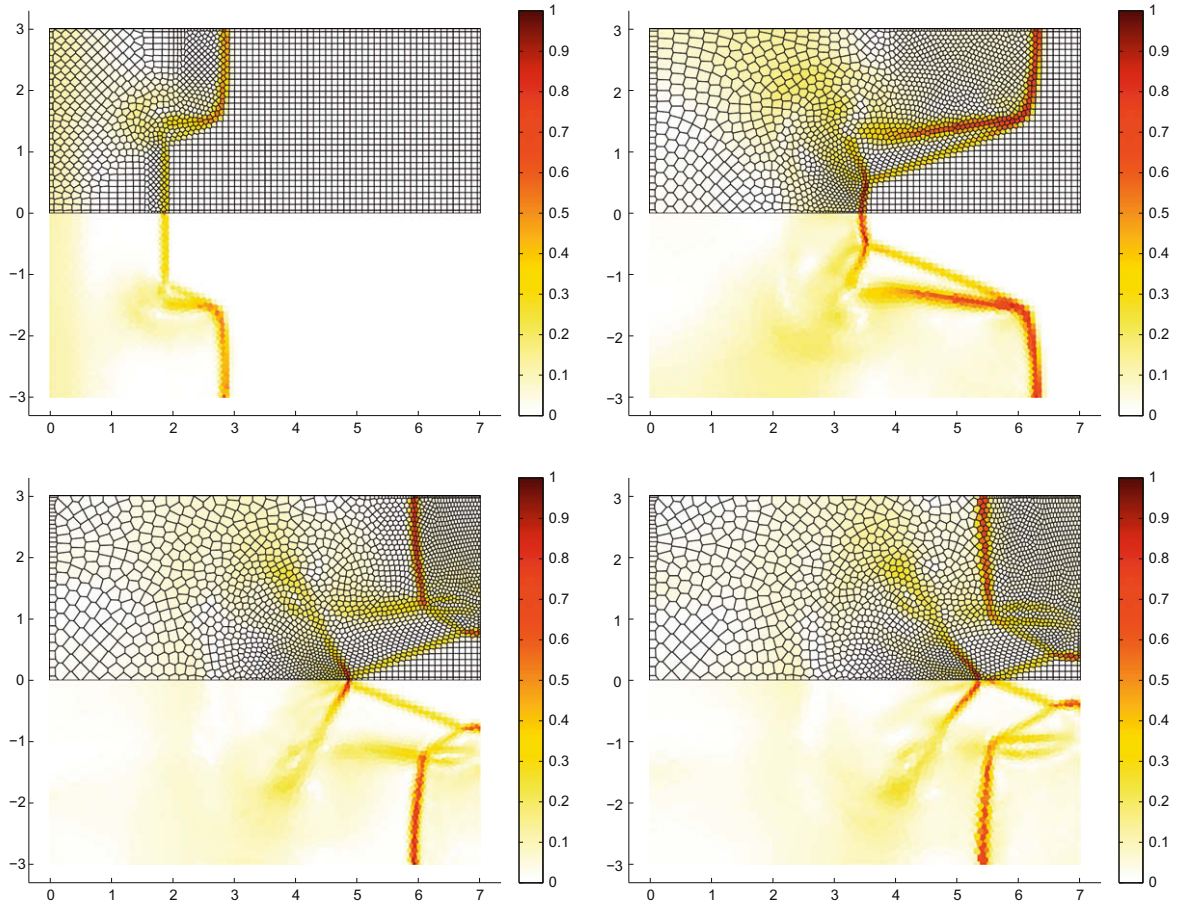

Fig. 18. Triple point problem at several times for ReALE – Cell-centered CHIC results for variable ω_c – From top-left to bottom-right: times $t = 1$, 3, 4.5, 5.

The CHIC-based methods give higher pick density value in comparison with ALE INC.-based methods because formally Lagrangian phase in CHIC method is second-order accurate and Lagrangian phase in ALE INC. method is first-order accurate. Small differences in ALE results (panels (c) and (d)) can be attributed to slightly different rezone strategy.

9.1.2. Generator displacement

The same problem is run with the cell-centered CHIC-based ReALE code and staggered ALE INC.-based ReALE code with three different generator the quasi-Lagrangian one ($\omega_c = 0$), the quasi-centroidal one ($\omega_c = 1$) and the motion based on the deformation tensor ($\omega_c = f(\lambda_1, \lambda_2)$).

The results for CHIC-based ReALE code presented in Fig. 11. As expected the quasi-Lagrangian generator motion leads to non-smooth mesh which, however, adapted quite well to the flow. One can mention that symmetry is not well preserved, which in particular can be attributed to non-smoothness of the mesh (Fig. 11(a) and (b)). On the other hand, the quasi-centroidal generator motion (Fig. 11(c) and (d)), leads to a smooth mesh that is not adapted anymore to the fluid flow; density is over-smoothed due to excessive remapping. Finally the motion based on the deformation tensor described in Section 7 (Fig. 11(e) and (f)), produces a locally smooth mesh and keeps finer cell region after the shock wave passes through, and coarser cell region after expansion.

In Fig. 12 we present the same results for the staggered ALE INC.-based ReALE, code and the same conclusions apply.

Results presented here justify rezone strategy based on the analysis of the deformation tensor described in Section 7 and the rest of the paper in all numerical ReALE simulations we use only this rezone strategy.

The Sedov test case is not extremely demanding neither for Lagrangian nor ALE scheme. Contrarily, the next test cases involve generation of the vorticity and are intended to demonstrate the capabilities of the new ReALE method.

9.2. Triple point problem – two material Riemann problem

This problem is a three state two material 2D Riemann problem in a rectangular vessel. The triple point problem simulation domain is $\Omega = [0;7] \times [0;3]$ as described in Fig. 13. Ω is split into three regions filled with two perfect gases leading to

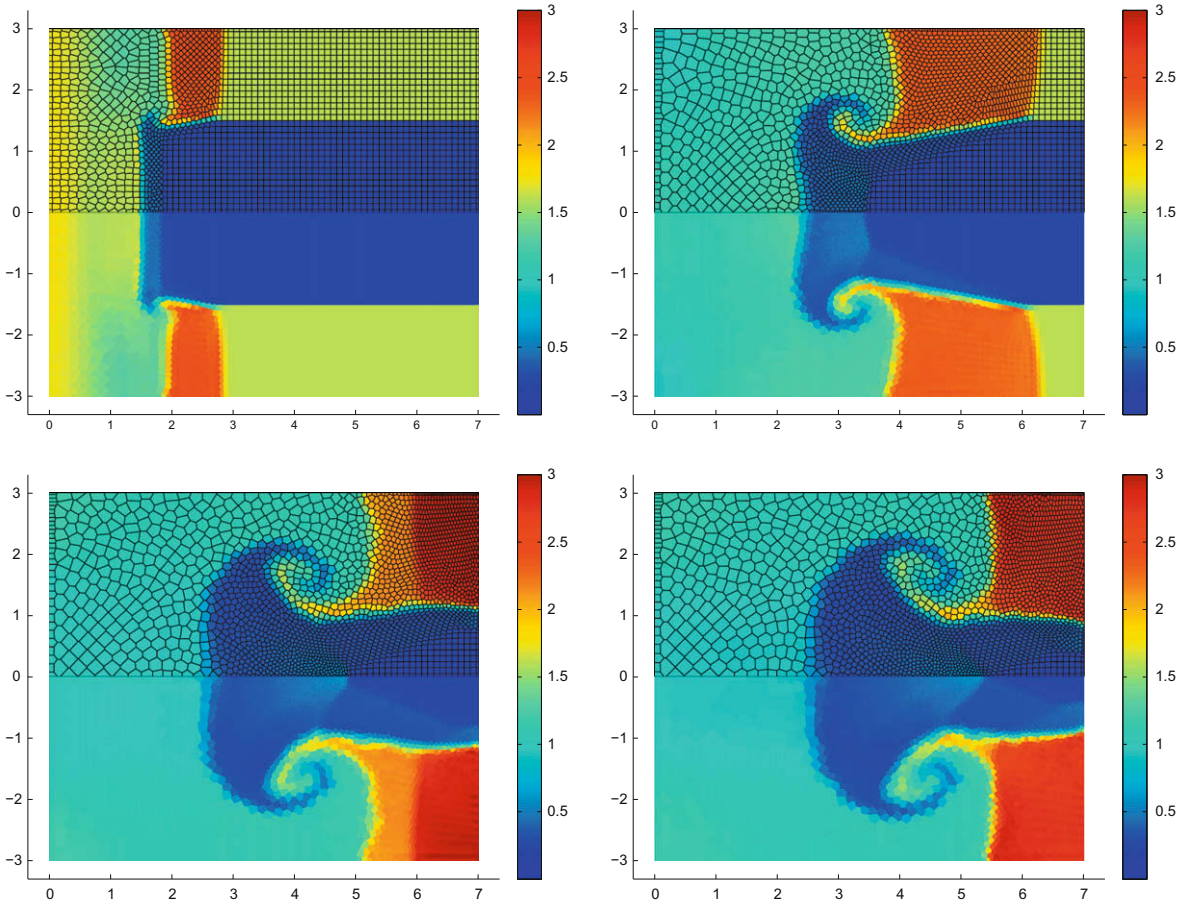

Fig. 19. Triple point problem at several times for ReALE – Staggered-ALE INC. results – internal energy and mesh. From top-left to bottom-right: times $t = 1$, 3, 4.5, 5.

a two material problem. The high pressure high density state is $\Omega_1 = [0;1] \times [0;3]$, the low pressure high density state is $\Omega_2 = [1;7] \times [0;1.5]$ and the low pressure low density is $\Omega_3 = [1;7] \times [1.5;3]$. The initial densities are $\rho_1 = \rho_3 = 1$, $\rho_2 = 0.125$, the initial pressures are $p_1 = 1$, $p_3 = p_2 = 0.1$, the initial velocity is zero everywhere. The perfect equation of state is used with $\gamma_1 = \gamma_3 = 1.5$, $\gamma_2 = 1.4$.

Due to the discrepancy in density, two shocks in domains Ω_2 and Ω_3 propagate with different speeds. This creates a shear along initial contact discontinuity and a vorticity formation. We note that the Lagrangian computation fails before vortex is developed due to the mesh tangling. Capturing the vorticity is the difficult part of such simulation when standard ALE method is used.

Initially 72×32 generators are positioned on a perfect quadrangular grid leading to 2304 degenerate Voronoi cells. The generators are located in such a way that, initially there is no mixed cells and the triple point coincides with a vertex of the mesh. This mesh is intentionally coarse, such that differences between the methods can be visually observed. The boundary conditions are reflective ones.

The fluid flow after the breakup of the initial discontinuity is characterized by a left facing rarefaction wave and two right facing shock waves separated by an "horizontal" contact discontinuity. These two shocks travels with different speeds since the densities of the materials are different. This leads to a strong vortex formation. The final time is $t = 5$.

9.2.1. Results obtained by cell-centered ReALE method

Lagrangian scheme failure. Time $t_{fail} \simeq 1.67$ corresponds, more or less, to the time after which any Lagrangian scheme inexorably fails. In Fig. 14 we present the Lagrangian (top panels), ALE (middle panels) and ReALE (bottom panels) results for the cell-centered CHIC code for this time moment. The internal energy and meshes are displayed in the left panel (top part is mesh and internal energy, bottom part is color map for internal energy). In the right panels of Fig. 14 we present the mesh for each method such that the cells have been colored according to in which domain (Ω_1 in dark-red, Ω_2 in orange, Ω_3 in white) they were initially located. The Lagrangian scheme produces results that follow the motion of the fluid

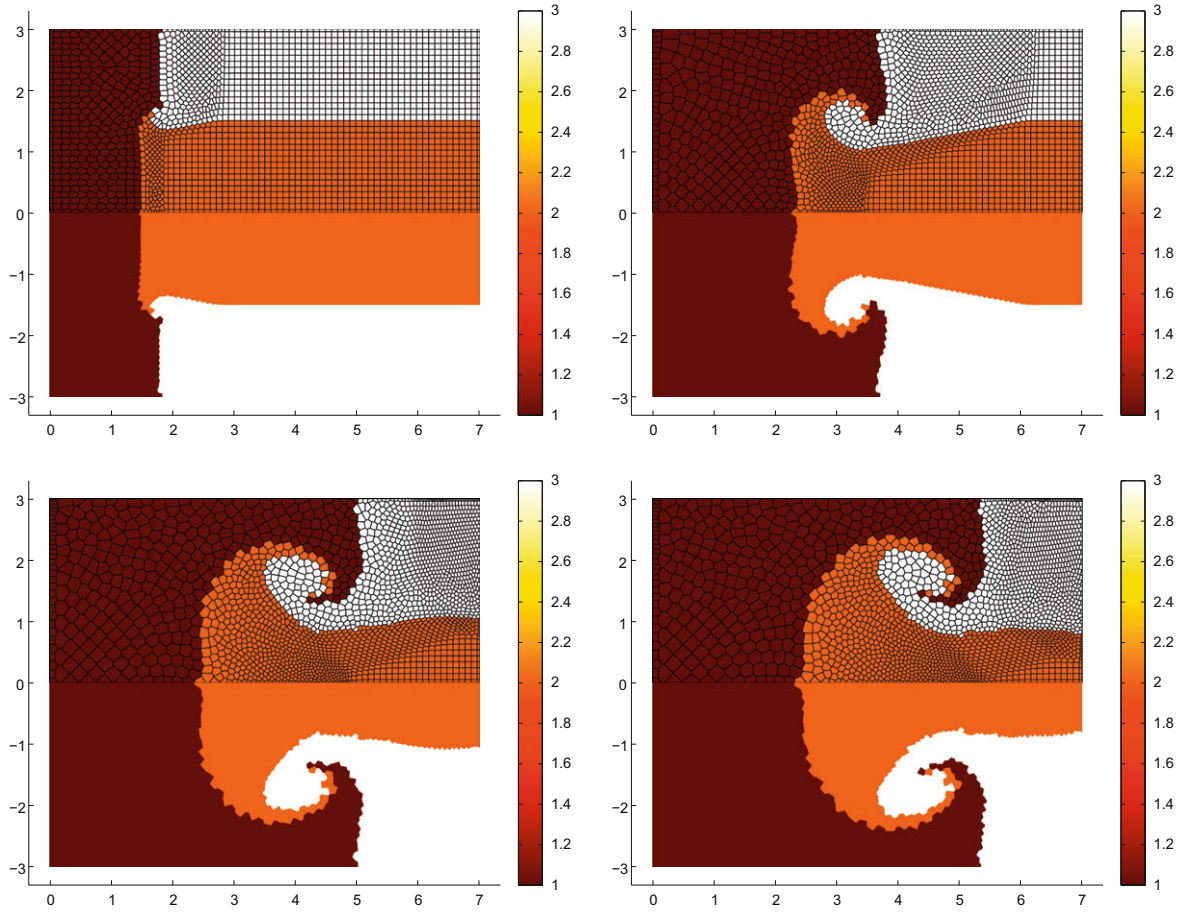

Fig. 20. Triple point problem at several times for ReALE – Staggered ALE INC. results – Cell color corresponds to the initial domain. From top-left to bottom-right: times $t = 1$, 3, 4.5, 5. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

but the scheme fails because of mesh tangling in vicinity of the vortex. The ALE method does not allow the mesh to follow the fluid motion and the mesh does not tangle. The ReALE seems to allow the cells to follow the vortex-shaped fluid motion, and up to this moment color map related to the initial location for Lagrangian and ReALE methods are very close, Fig. 14. This is indication that our rezone strategy is able to keep "centers" of the cells very close to its Lagrangian positions.

It is important to note that due to our choice of parameter ω_c, participating in movement of generators in ReALE method, meshes in front of shocks have not changed.

"Stagnation" of the mesh in ALE method. ALE and ReALE perform up to the final time $t = 5$. In Fig. 15 we present results for ALE and ReALE for final time-arrangements are the same as in Fig. 14. Results of the ALE calculation clearly show a mesh "stagnation" behavior as can be seen on the initial domain color map, whereas the ReALE allows the cells to be carried along with the vortex.

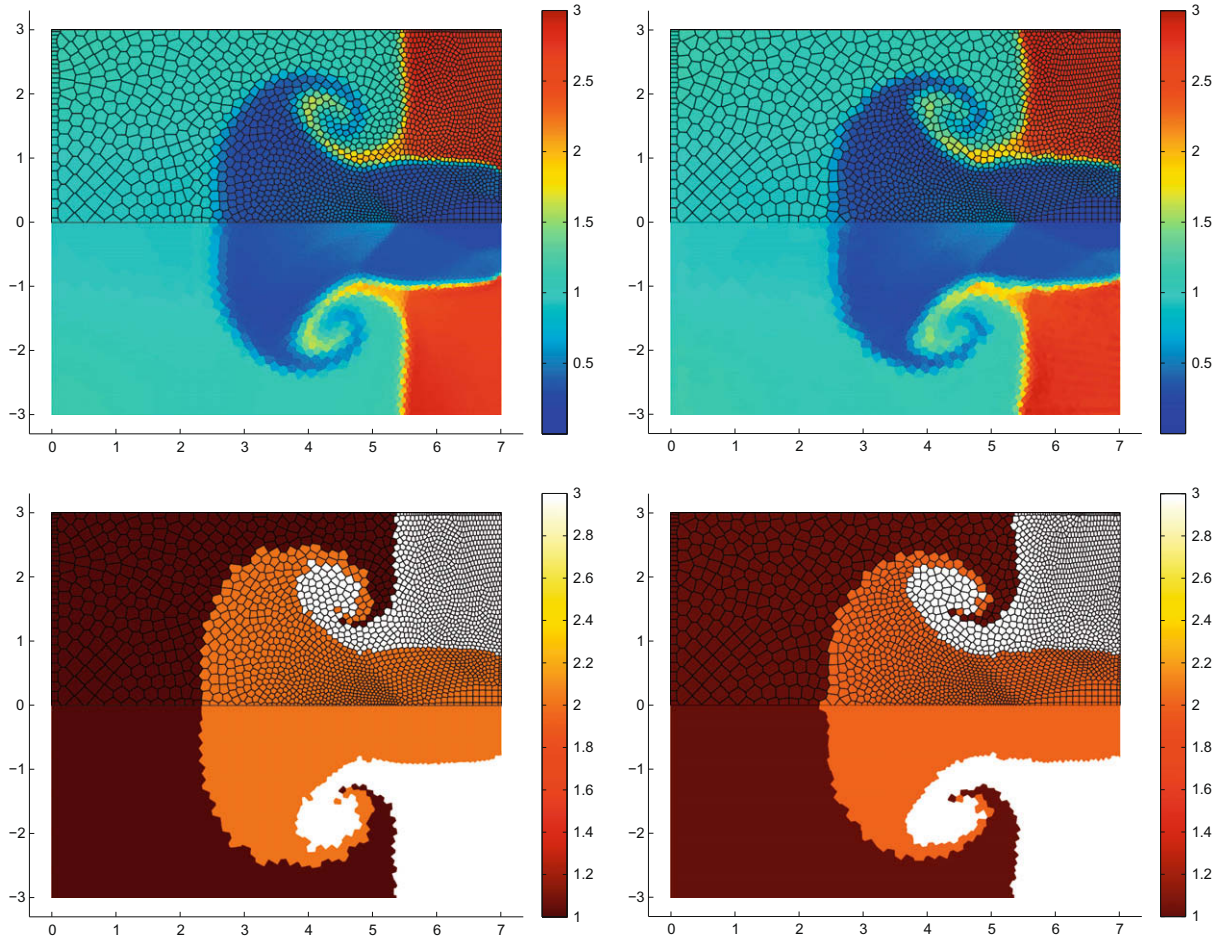

Fig. 21. Triple point problem. Comparison of cell-centered (left panels) and staggered (right panels) ReALE methods at the final time moment. Top panels – internal energy and mesh, bottom panels – coloring by initial region. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

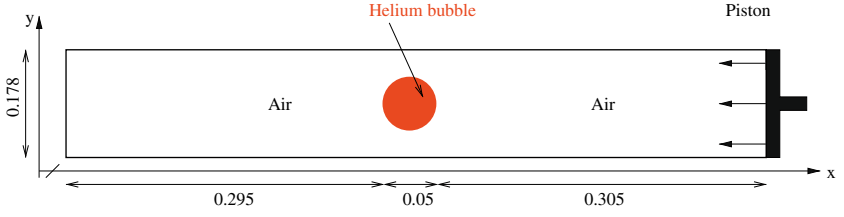

Fig. 22. Shock/Bubble interaction. A piston (right-boundary) moving to the left and compresses air initially at rest sending a shock wave that passes through an helium bubble.

In ALE calculation (top panels of Fig. 15), the mesh is "freezes" near the vortex center and the computation continues in an almost Eulerian fashion (as Lagrange + Remap because the rezone phase systematically backs up the Lagrangian t^{n+1} mesh onto the previous t^n Lagrangian mesh).

In ReALE calculation (bottom panels of Fig. 15), mesh is not Lagrangian, but it preserves Lagrangian character of the flow, because mesh follows the vortex as generators are carried within the fluid in an almost Lagrangian fashion (bottom-right panel in Fig. 15).[3] As a consequence ReALE has a better accuracy, which can be seen comparing internal energy color maps for ALE and ReALE. In ReALE results one can see roll up formation and in ALE results it is not that pronounced. Let as note that quantitative analysis of accuracy will be presented in the separate paper.

ReALE results. In Fig. 16 we present dynamics of the ReALE simulation by showing the mesh and the specific internal energy at different time moments $t = 1$ (beginning of vortex development), $t = 3$ (before the fastest shock reaches right wall), $t = 4.5$ (after shock reflection), $t = 5$ (after reflected shock reaches contact discontinuity). The meshes with initial domain coloring are presented in Fig. 17.

Features of the flow and ω_c factor. In Fig. 18 are presented the ω_c factors for different snapshots of the simulation. The ω_c factor shows the deformation encountered by the mesh between two last time steps. Comparison of these plots with structure of the main waves presented in Fig. 16 clearly demonstrate that ω_c dynamically detect main features of the flow. It is important to note that ω_c is very small in high vorticity regions (as it intended to be) and therefore movement of the generators in these areas is almost Lagrangian.

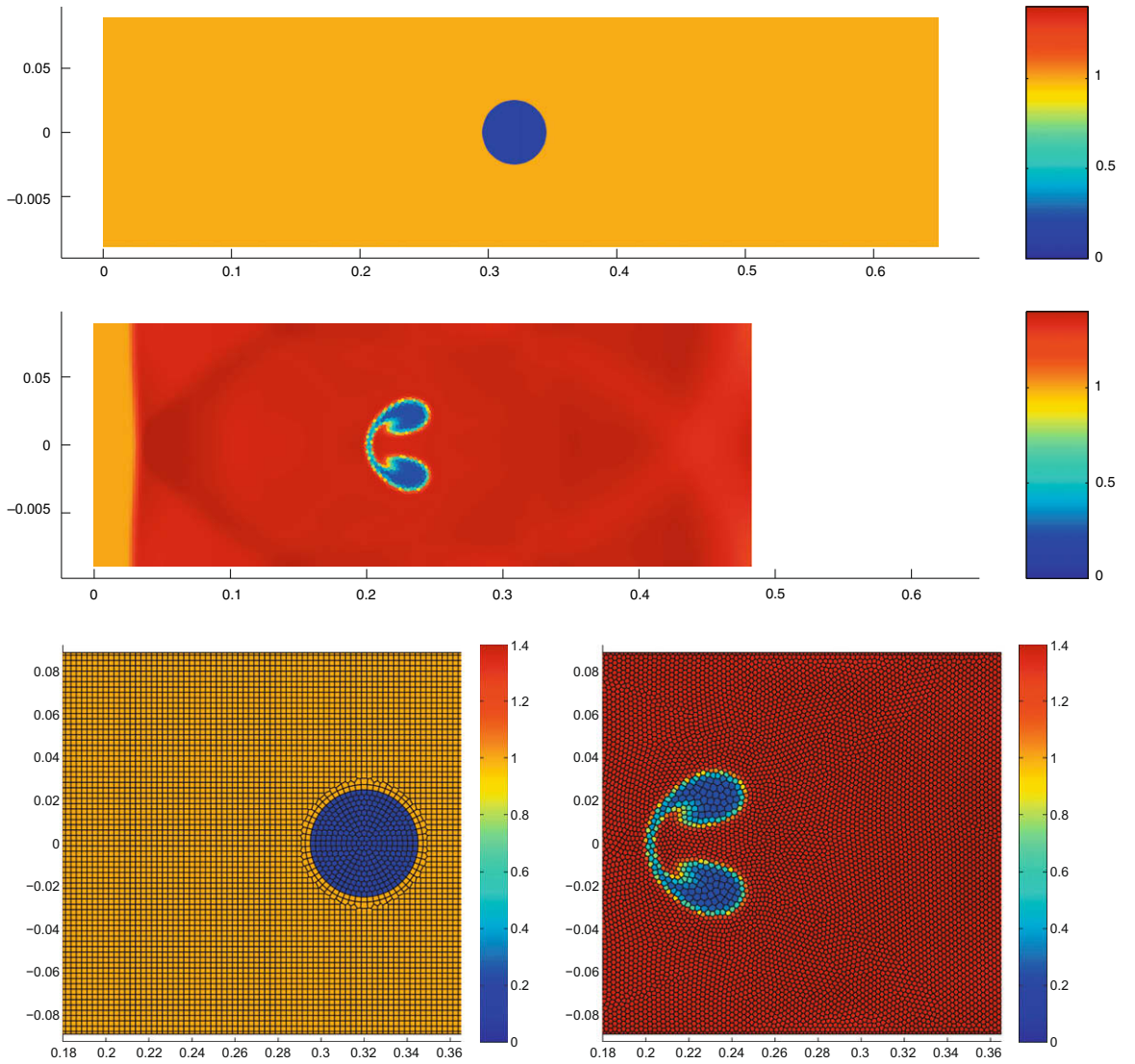

Fig. 23. Shock/Bubble interaction problem. Top: Initial density; Middle: Final density; Bottom: Zoom on the initial bubble and final bubble density and mesh.

9.2.2. Comparison of cell-centered and staggered ReALE methods

In this section we give brief comparison of ReALE calculations based on cell-centered CHIC method with results obtained by ReALE method based on staggered ALE INC. method.

In Fig. 19 we present internal energy and mesh plots for ALE INC.-based ReALE method for different time moments. It has to be compared with CHIC-based ReALE results presented in Fig. 16. In Fig. 20 we present coloring by initial region for staggered method.

Finally, we present side by side comparison of two methods in Fig. 21.

One can see only very small differences between cell-centered and staggered ReALE methods.

Results presented in this section and results for Sedov problem allows us to conclude that ReALE approach can be used for either cell-centered or staggered discretization basic Lagrangian scheme.

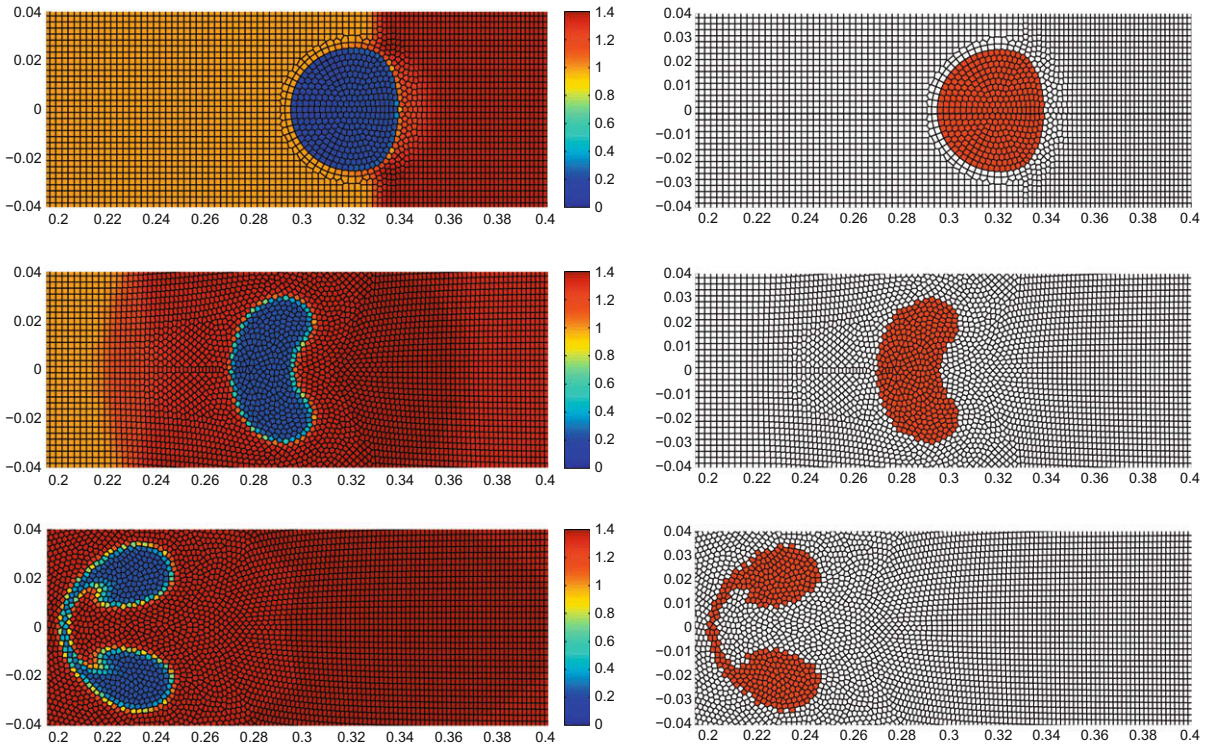

Fig. 24. Shock/Bubble interaction problem at $t = 1005.15 \times 10^{-6}$ (top panel), $t = 1101.44 \times 10^{-6}$ (middle panel) and $t = t_{end} = 1342.153 \times 10^{-6}$ (bottom panel) – Left column: Density and mesh. Right column: Coloring by the initial domains (white: air, red: bubble). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

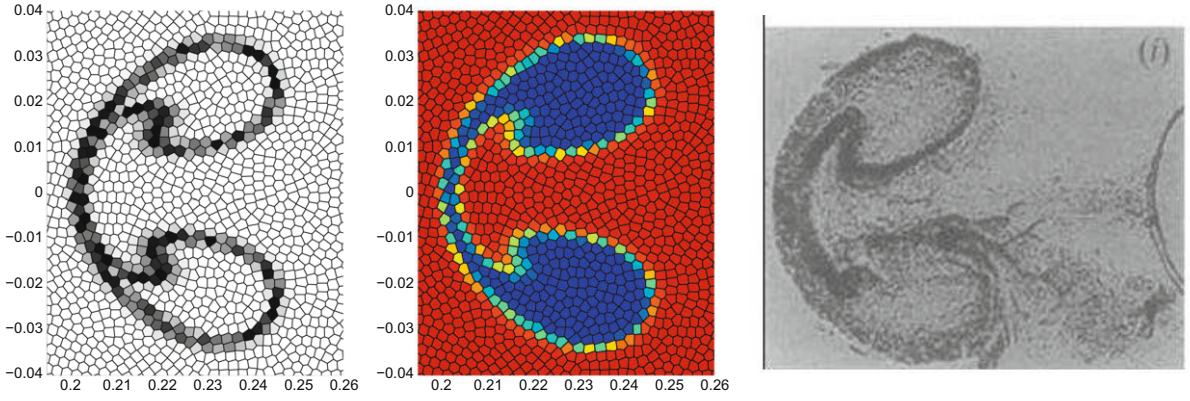

Fig. 25. Shock/Bubble interaction problem at $t_{end} = 1342.153 \times 10^{-6}$— Left: Zoom of color map for $I_c = C_{h,c}(1 - C_{h,c})$: grayscale goes from $I_c = 0$ (white) to $I_c = 1/4$ (black); Middle: Zoom of color map for density: colorscale goes from $\rho_c = 0.23$ (blue) to $\rho_c = 1.42$ (red); – Right: Schlieren image from experimental data in [62]. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

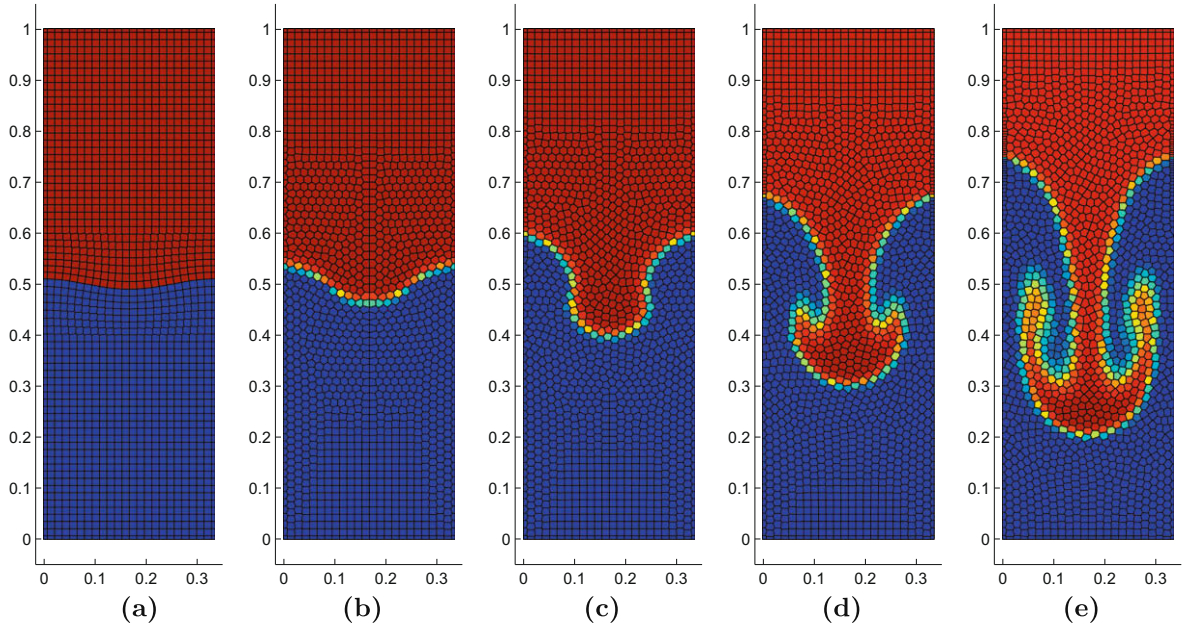

Fig. 26. Rayleigh–Taylor instability – ReALE results at times $t = 0, 3, 5, 7, 9$ (from (a) to (e)) – Mesh and density – Color scale is from blue ($\rho = 1$) to red ($\rho = 2$). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Detailed comparison of cell-centered and staggered discretizations is beyond of the goal. Actually it will require full new paper. At this moment it will be premature to recommend one discretization over the other. Our goal in this paper is to demonstrate that ReALE methodology can be used for both types of discretizations. For the sake of space, in the reminder of the paper we will present only results for cell-centered discretization.

9.3. Bubble shock interaction

The computational domain is $\Omega = [0; 0.65] \times [0; 0.178]$. The bubble is a disk defined by the coordinates of its center $(x_c, y_c) = (0.320, 0)$ and its radius $R_b = 0.025$ (see Fig. 22). We prescribe reflective boundary conditions at each boundary except the right-boundary (initially at $x = 0.65$); right-boundary is the piston which moves inward with velocity $\boldsymbol{V}^\star = (u^\star, 0)$. The incident shock wave is defined by its Mach number, $M_s = 1.22$. The bubble and the air are initially at rest. The initial data for Helium bubble is $(\rho_1, P_1) = (0.182, 10^5)$, its molar mass is $\mathcal{M}_1 = 5.269 \times 10^{-3}$ and its polytropic index is $\gamma_1 = 1.648$. The initial data for air is $(\rho_2, P_2) = (1, 10^5)$, its molar mass is $\mathcal{M}_2 = 28.963 \times 10^{-3}$ and its polytropic index is $\gamma_2 = 1.4$. It is two material problem. 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 x-velocity of the piston is given by $u^\star = -124.824$. The x-component of the incident shock velocity is $D_c = -456.482$. The incident shock wave hits the bubble at time $t_i = 668.153 \times 10^{-6}$. The stopping time for our computation is $t_{end} = t_i + 674 \times 10^{-6} = 1342.153 \times 10^{-6}$. It corresponds to the time for which experimental Schlieren graph (results from [62]) is displayed in [103] (Fig. 9(i)). Initial mesh is constructed with a set of 19,061 generators and designed to produce a Voronoi mesh that has a mesh lines which matches the bubble boundary (see bottom left panel in Fig. 23, where a zoom of the initial mesh around the bubble is presented). The description of initial mesh generation given in Section 4. Initial density is presented in top panel in Fig. 23. The middle panel presents the final density which corresponds to the deformed bubble. The bottom panels of this figure show a zoom on the initial and final bubble location and the underlying initial and final meshes.

In Fig. 24 (left column) we present density at several intermediate time moments for a zoomed region around the bubble ($t = 1005.15 \times 10^{-6}$ (top panel), $t = 1101.44 \times 10^{-6}$ (middle panel) and $t = t_{end} = 1342.153 \times 10^{-6}$ (bottom panel)). Color maps related to coloring by initial region are presented in right column of Fig. 24 (white color for the cells originally in the air; red color for the cells originally in the bubble). One can conclude that Lagrangian motion is well preserved by ReALE method.

In Fig. 25 (left panel) we present the cell-centered value $I_c = C_{h,c}(1 - C_{h,c})$ at final time $t_{end} = 1342.153 \times 10^{-6}$, where $C_{h,c}$ is the concentration of helium in cell c. Originally $C_{h,c}$ is equal to 1 if generator c belongs to helium bubble and 0 otherwise. Therefore, I_c is initially 0 for all cell. As the simulation advances, mixed cells are created close to the interface between the different materials, leading to values $I_c \neq 0$. The I_c map show how much interface is diffused; grayscale spreads from $I_c = 0$ (white) to $I_c = 1/4$ (black). As seen in left panel in Fig. 25 interface region is spread only over one or two cells.

Experimental results for this test case can be found in [62] and high-resolution numerical experiments performed with an AMR Eulerian code in [103]. In Fig. 25 (middle and right panels) a comparison of the Schlieren image of the experimental results from [62] (right panel) with our numerical results (middle panel) at $t_{end} = 1342.153 \times 10^{-6}$.

Presented numerical results show that qualitatively ReALE technique can reproduce experimental results (Fig. 25). We are planning to perform quantitative comparison with available experimental results as well as with high-resolution numerical results presented in [103] in special paper.

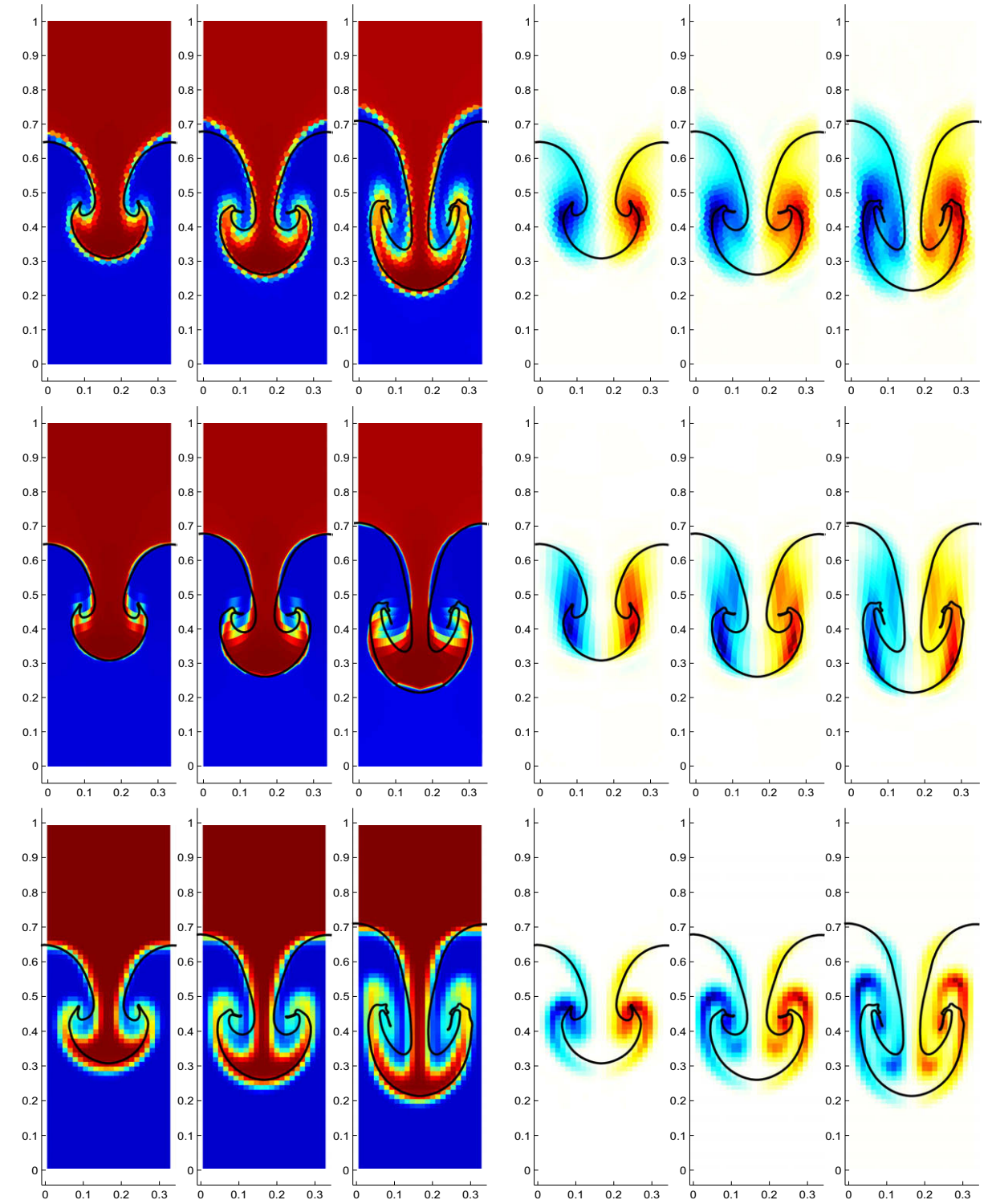

Fig. 27. Rayleigh–Taylor instability – Top panel: ReALE results. Middle panel: ALE results. Bottom panel: RTI3D Eulerian code results – Left to Right: Density at $t = 7, 8, 9$, Vorticity at $t = 7, 8, 9$. Color scale is from blue (minimal negative vorticity) to red (maximal positive vorticity). White color corresponds to zero vorticity. Frontier interface is the thick black line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

9.4. Rayleigh–Taylor instability

Our final test in this paper is Rayleigh–Taylor instability problem. It consists of two ideal gases with densities $\rho_{top} = 2$ and $\rho_{bottom} = 1$; in both cases the adiabatic constant $\gamma = 1.4$. Initially, the heavier gas is above the lighter gas in rectangular vessel $\Omega = [0{:}1/3] \times [0{:}1]$, with gravitational field directed vertically downward and with magnitude $g = 0.1$. The interface has been deliberately perturbed as described by formula $y(x) = \frac{1}{2} + \frac{1}{100}\cos(6\pi x)$. Initially both gases are at the rest; the pressure distribution is approximately hydrostatic and is defined in the lighter gas

$$p_{bottom}(x,y) = 1 + \frac{1}{2}\rho_{top} g + \rho_{bottom} g\left(\frac{1}{2} - y\right),$$

and in the heavier gas:

$$p_{top}(x,y) = 1 + \rho_{top} g(1 - y).$$

It is well known that this configuration is unstable and as time progresses, the heavier gas will sink and the lighter gas will rise through the formation of bubbles and spikes. This problem does not involve any shock wave, but the vorticity is so high that pure Lagrangian schemes eventually fail.

9.4.1. Initial phase of instability

The ReALE simulation starts with a Voronoi cleaned mesh obtained *via* 24×72 generators (24 in x-direction, *72* in y, see Fig. 26(a)). These are initially set so that the interface $y(x)$ is well approximated by edges of cells (see Fig. 26(a)).

In Fig. 26 we present the density and mesh for several time moments: $t = 0$, 3, 5, 8, 9.

As a matter of comparison and verification we use results obtained by others methods:

- FronTier (front tracking) code [41,58,59] is used to get a reference solution for the interface between the two fluids. The results of this code are used by the courtesy of J. Grove of the Los Alamos National Laboratory. FronTier is run with much more finer resolution (106×320 cells). This interface is plotted with a black thick lines in Fig. 27.
- RTI3D code (an implicit large eddy simulation (ILES) incompressible Eulerian code based on control volume approach, using a second-order Van-Leer method for volume fraction and momentum advection, see [4]). It is used with a grid resolution of 24×74 fixed cells. This Eulerian code provides a qualitative spatial and temporal behavior of vorticity. The results of this code are used by the courtesy of M.J. Andrews of the Los Alamos National Laboratory.
- Finally the CHIC-based ALE code without reconnection is used with an initial logically rectangular grid of 24×72.

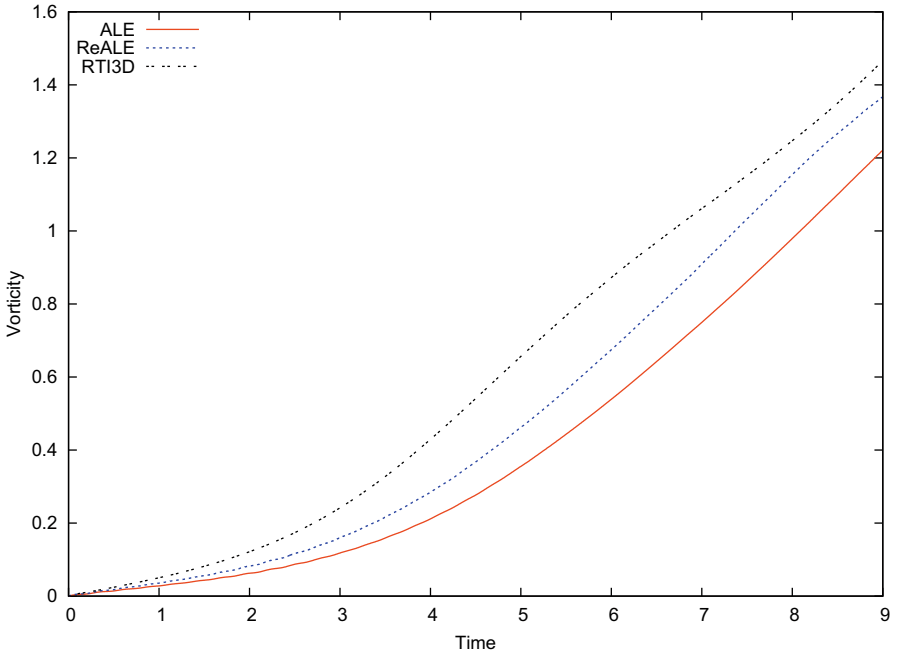

Fig. 28. Time evolution of the mean vorticity for the Rayleigh–Taylor instability – ReALE, ALE and RTI3D results are displayed.

In Fig. 27 we present the density (t = 7, 8, 9) and the vorticity (t = 7, 8, 9) for the ReALE, ALE codes (top and middle panels) and the RTI3D code (bottom panel). The FronTier interface is plotted as a thick black line on the top of each graph. The color scale for density is the same for all calculations, it is from 1 to 2. The general shape is resolved reasonably well by all three methods The tip of the interface is better resolved by ReALE method, Fig. 27 – left panels.

In the right panels in Fig. 27 we present color maps for vorticity, $\mathbf{curl}_c^h\mathbf{u}$ which is finite difference approximation of **curlu** in the cell Ω_c based on Green formula. For vorticity we have following ranges. For time t = 7 – ReALE: −3.53/3.55; ALE: −3.04/

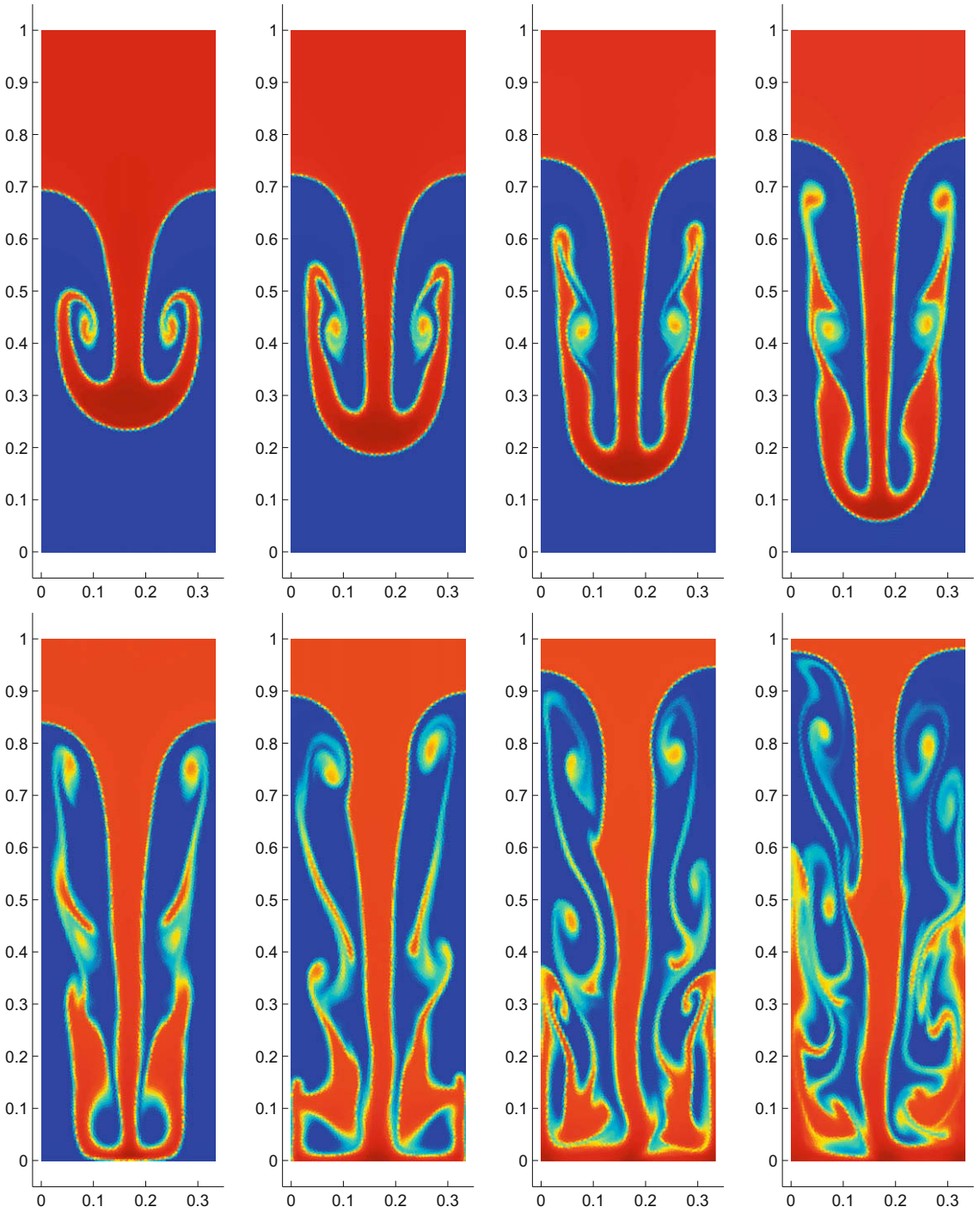

Fig. 29. Rayleigh–Taylor instability – ReALE with 66 × 200 generators at t = 8, 9, 10, 11, 12, 13, 14, 15 (from top-left to bottom-right) – Density.

3.04; RTI3D: −5.00/5.00. For time $t = 8$ – ReALE: −3.72/3.84; ALE: −3.68/3.68; RTI3D: −4.95/4.95. For time $t = 9$ – ReALE: −3.81/3.89; ALE: −4.52/4.52; RTI3D: −5.61/5.61.

We think that because RTI3D is incompressible code its vorticity color map is more sharp. ReALE clearly gives results closer to RTI3D and much sharper than ALE results. The ReALE results seem to better match spatial distribution of the vorticity, which matches interface shape. We wish to point that with this low resolution the ALE simulation code produces acceptable results. However, if a finer resolution is to be used, the mesh becomes very pinched and stretched so that numerical oscillations are generated (see Fig. 1 from the motivation section as example). As a consequence vorticity and, in general, most physical variables are contaminated leading to stability issues. In ReALE due to reconnection mesh follows the fluid, which allows to obtain more meaningful results (see Section 9.4.2 for high-resolution ReALE results).

Finally in Fig. 28 we show the mean vorticity, as a function of time for all three methods ReALE, ALE and RTI3D. Discrete mean vorticity is defined as follows:

$$\sqrt{\frac{\sum_c \left\{ |\mathbf{curl}_c^h \mathbf{u}|^2 |\Omega_c| \right\}}{|\Omega|}} \sim \sqrt{\frac{\int_\Omega |\mathbf{curlu}|^2 dV}{|\Omega|}}.$$

We observe that the ReALE suppresses vorticity less than ALE does. Qualitatively time evolution of mean vorticity is very similar for all three methods.

9.4.2. ReALE simulation of the later stages of the Rayleigh–Taylor instability

In this section we present high-resolution (66×200 generators are used leading to 13,200 cells) results for later stages of Rayleigh–Taylor instability. We perform our calculations up to time $t = 15$ when the heavy fluid has reached the bottom of the vessel and lighter fluid almost reached top of the vessel. In Fig. 29 we present color maps of the density at time moments $t = 8, 9, 10, 11, 12, 13, 14, 15$. In the last three figures one can clearly see violation of symmetry with respect to central vertical line. This is because this flow is very unstable and also because cleaning procedure in our rezone strategy is non-symmetric and depends on cell ordering.

Finally, in Fig. 30, we present density, vorticity and cells colored by initial domain at time $t = 15$. Right panel in Fig. 30 also shows the mesh.

Results presented in this section shows that ReALE method can be used to run problems with strong shear deformation without any special tuning of the parameters of the rezone strategy during the calculations. This is impossible for standard ALE methods without mesh reconnection, where user intervention is usually required.

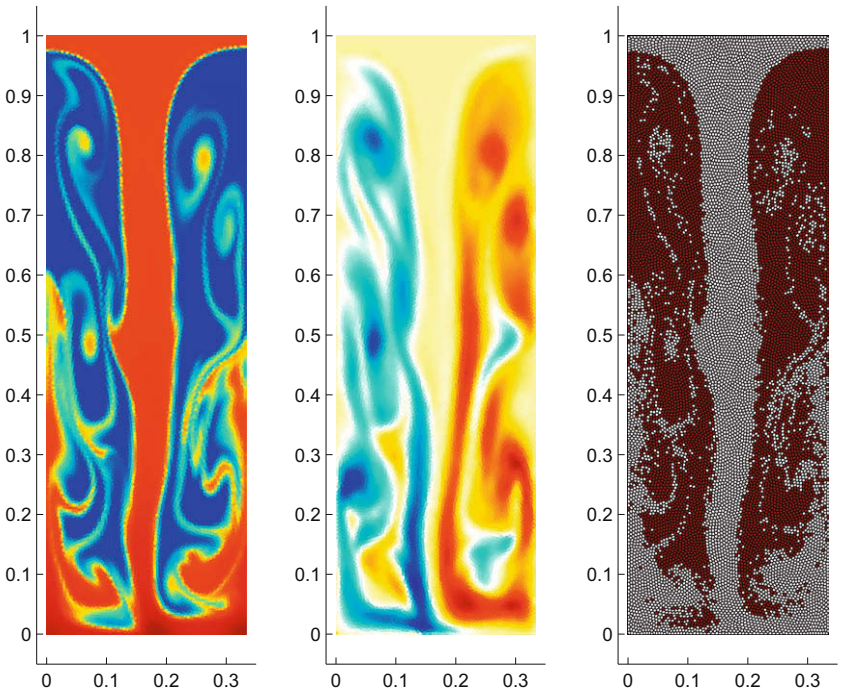

Fig. 30. Rayleigh–Taylor instability – ReALE with 66×200 generators at $t = 15$ – Density, vorticity and, mesh and cells colored by their initial domain; Vorticity scale is from −11.78 to *9.63* (blue for the minimal value to red for the maximal one). (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.).

10. Conclusion and perspectives

We have presented a new reconnection-based ALE method. It includes three main elements.

- An explicit Lagrangian phase in which the solution on polygonal mesh is updated (without changing mesh connectivity).
- A rezoning phase in which a new grid is defined using specific movement of generators and formalism of Voronoi diagrams. It allows the change of mesh connectivity. In this work we keep number of cells unchanged, but number of vertices of each cell can change due to connectivity evolution. Generator movement is chosen in such a way that cell movement is close to Lagrangian and cell shape is close to regular hexagon.
- A remapping phase in which the Lagrangian solution is transferred (conservatively interpolated) from one polygonal mesh to another.

On numerical examples we have demonstrated that our new method is more accurate and robust in comparison with standard ALE methods with fixed connectivity.

We recognize that our new method requires more testing, which we are planning to do in the future. We also recognize that the question of efficiency is very important. The particular implementation of our method used to obtain numerical results in this paper was not intended to be optimal and uses pieces which originally were not intended to work together. For this reason we do not present any comparison of efficiency of ALE and ReALE. We will do it in the future paper.

Also we are planning to explore different mechanisms for mesh adaptation. In the framework of current ReALE method at rezone stage we can develop new strategies for the choice of ω parameter to reflect features of the flow. We also could move vertices (as in standard ALE) after reconnection is done. We are planning to use mechanism of weighted Voronoi diagrams [44] to introduce adaptivity by choosing weight proportional to some monitor function, which may be an error indicator or may just reflect some physics which requires mesh refinement. The adaptation can be also achieved by adding or deleting cells. In principle, ReALE-like methods are perfectly suited for this strategy (cf. [37]).

In next paper we are planning to present 2D results in $r-z$- axisymmetric geometry, which will allow to demonstrate performance of our method for interesting and more realistic problems.

Also in future we will incorporate interface reconstruction methods such as volume of fluid (VOF) [105,106] and moment of fluid (MOF) [52,2,75]. Moreover, we plan to incorporate more advanced closure models for mixed multi-material cells [111]. This will increase accuracy of multi-material calculations.

There is no conceptual difficulties in extending ReALE methodology to 3D, however, first, we plan to improve efficiency of our new method in 2D by making code parallel.

Finally, we are planning to incorporate material strength into ReALE code. We are considering approach described in the recent paper [73].

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 authors also are very grateful to M.J. Andrews of the Los Alamos National Laboratory for the results of his RTI3D code on the Rayleigh–Taylor instability test problem. The authors also thank J. Grove from the Los Alamos National Laboratory for providing the Frontier results and allowing their use.

References

[1] F.L. Adessio, D.E. Carroll, K.K. Dukowicz, J.N. Johnson, B.A. Kashiwa, M.E. Maltrud, H.M. Ruppel, CAVEAT: a computer code for fluid dynamics problems with large distortion and internal slip, Technical Report LA-10613-MS, Los Alamos National Laboratory, 1986.
[2] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096–2132.
[3] R.W. Anderson, N.S. Elliott, R.B. Pember, An arbitrary-Lagrangian–Eulerian method with adaptive mesh refinement for the solution of the Euler equations, J. Comput. Phys. 199 (2004) 598–617.
[4] M.J. Andrews, Accurate computation of convective transport in transient two-phase flow, Int. J. Numer. Meth. Fluid 21 (3) (1995) 205–222.
[5] N.V. Ardelyan, V.I. Bychkov, K.V. Kosmachevskii, M.N. Sablin, Numerical simulating 2D plasma dynamics problems on the base of implicit free-Lagrange method, in: Proceedings of WEHSFF 2007 (ECCOMAS Thematic Conference) – WEST-EAST HIGH SPEED FLOW FIELD CONFERENCE, November 19–22, 2007, Moscow, Russia, 2007. Available on web: <http://wehsff.imamod.ru/pages/s7.htm>.
[6] N.V. Ardelyan, K.V. Kosmatchevskii, Implicit free-Lagrange method for computing two-dimensional magnetogasdynamic flows, Comput. Math. Model. 6 (4) (1995) 209–224.
[7] F. Aurenhammer, Voronoi diagrams – a survey of fundamental geometric data structures, ACM Comput. Surveys 23 (3) (1991) 345–405.
[8] F. Aurenhammer, R. Klein, Voronoi diagrams, Technical Report 198, Fern Universität Hagen, Department of Computer Science, Germany, 1996. Available on web: <http://www.pi6.fernuni-hagen.de/publ/tr198.pdf>.
[9] M.J. Baines, Moving Finite Elements, Oxford Science Publications, Clarendon Press, Oxford, 1994.
[10] G.J. Ball, A free-Lagrange method for unsteady compressible flow: simulation of a confined cylindrical blast wave, Shock Waves 5 (1996) 311–325.

[11] G.J. Ball, Sliding interfaces in a 2D free-Lagrange Godunov scheme on a Voronoi mesh, in: Workshop on Numerical Methods for Multi-material Flows, Institut Henri Poincaré, Paris, September 23–25, 2002. Available on web: <http://www.ann.jussieu.fr/~despres/ABSTRACT/gjball_abs.pdf>.
[12] G.J. Ball, B.P. Howell, T.G. Leighton, M.J. Schofield, Shock-induced collapse of a cylindrical air cavity in water: a free-Lagrange simulation, Shock Waves 10 (2000) 265–276.
[13] R.A. Barabanov, O.I. Butnev, S.G. Volkov, B.M. Jogov, V.A. Pronin, Method Meduza-3D for computation of three-dimensional problems of gas dynamics on irregular meshes (In Russian), Quest. Atomic Sci. Tech. Ser.: Math. Mod. Phys. Process. 2 (2005) 15–26.
[14] A.J. Barlow, ALE and AMR mesh refinement techniques for multi-material hydrodynamics problems, ICFD Workshop on Mesh Refinement Techniques 7th December 2005, <http://www.icfd.rdg.ac.uk/Workshops/AMR/meshreftechf.pdf>.
[15] A.J. Barlow, Challenges and recent progress in developing numerical methods for multi-material ALE hydrocodes, ICFD 25 Year Anniversary Conference 15–16th September 2008, Oxford University, <http://www.icfd.rdg.ac.uk/ICFD25/Talks/ABarlow.pdf>.
[16] A.J. Barlow, A compatible finite element multi-material ALE hydrodynamics algorithm, Int. J. Numer. Meth. Fluid 56 (2008) 953–964.
[17] T.J. Barth, Numerical methods for gasdynamic systems on unstructured meshes, in: D. Kroner, M. Ohlberger, C. Rohde (Eds.), 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, Springer, Berlin, 1997, pp. 195–284.
[18] A.L. Bauer, D.E. Burton, E.J. Caramana, R. LoubFre, M.J. Shashkov, P.P. Whalen, The internal consistency, stability, and accuracy of the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys. 218 (2) (2006) 572–593.
[19] D.J. Benson, An efficient, accurate, simple ALE method for nonlinear finite element programs, Comput. Meth. Appl. Mech. Eng. 72 (1989) 305–350.
[20] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Meth. Appl. Mech. Eng. 99 (1992) 235–394.
[21] D.J. Benson, Momentum advection on a staggered mesh, J. Comput. Phys. 100 (1992) 143–162.
[22] A.F. Bower, Applied Mechanics of Solids, CRC Press, 2009.
[23] J.U. Brackbill, J.J. Monaghan (Eds.), Proceedings of the workshop on particle methods in fluid dynamics and plasma physics, Comput. Phys. Commun. 8(1), 1987.
[24] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (1982) 342–368.
[25] D.E. Burton, Exact conservation of energy and momentum in staggered-grid hydrodynamics with arbitrary connectivity, Advances in the Free-Lagrange Method, Springer-Verlag, New York, 1990.
[26] D.E. Burton, Consistent finite-volume discretization of hydrodynamics conservation laws for unstructured grids, Report UCRL-JC-118788, Lawrence Livermore National Laboratory, 1994.
[27] D.E. Burton, Multidimensional discretization of conservation laws for unstructured polyhedral grids, Report UCRL-JC-118306, Lawrence Livermore National Laboratory, in: Proceedings of Second International Workshop on Analytical Methods and Process Optimization in Fluid and Gas Mechanics (SAMGOP), Arzamas-16, Russia, 1994.
[28] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (4) (2001) 739–765.
[29] J.C. Campbell, M.J. Shashkov, A compatible Lagrangian hydrodynamics algorithm for unstructured grids, Seltuk J. Appl. Math. 4 (2) (2003) 53–57.
[30] E.J. Caramana, D.E. Burton, M.J. Shashkov, P.P. Whalen, The construction of compatible hydrodynamics algorithms utilizing conservation of total energy, J. Comput. Phys. 146 (1) (1998) 227–262.
[31] E.J. Caramana, R. Loubère, "Curl-q": a vorticity damping artificial viscosity for Lagrangian hydrodynamics calculations, J. Comput. Phys. 215 (2) (2006) 385–391.
[32] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distorsion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (1998) 521–561.
[33] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multidimensional shock wave computations, J. Comput. Phys. 144 (1998) 70–97.
[34] G. Carré, S. Del Pino, B. DesprTs, E. Labourasse, A cell-centered Lagrangian hydrodynamics scheme in arbitrary dimension, J. Comput. Phys. 228 (14) (2009) 5160–5183.
[35] R.A. Clark, Compressible Lagrangian hydrodynamics without Lagrangian cells, Report LAUR-85-1882, Los Alamos National Laboratory, 1985.
[36] R.A. Clark, The evolution of HOBO, Comput. Phys. Commun. 48 (1988) 61–64.
[37] M.C. Cline, J.K. Dukowicz, F.L. Addessio, CAVEAT-GT: a general topology version of the caveat code, Technical Report LA-11812-MS, Los Alamos National Laboratory, 1990.
[38] P. Colella, A direct Eulerian MUSCL scheme for gas dynamics, SIAM J. Stat. Comput. 6 (1) (1985) 104–117.
[39] W.P. Crowley, FLAG: a free-Lagrange method for numerically simulating hydrodynamic flows in two dimensions, in: M. Holt (Ed.), Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, vol. 8, September 15–19, 1970, University of California, Berkeley, Lecture Notes in Physics, Springer, Berlin, 1971, pp. 37–43.
[40] B. DesprTs, C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Rational Mech. Anal. 178 (2005) 327–372.
[41] J. Du, B. Fix, J. Glimm, X. Li, Y. Li, L. Wu, A simple package for front tracking, J. Comput. Phys. 213 (2) (2006) 613–628.
[42] Q. Du, M. Emelianenko, M. Gunzburger, Convergence of the Lloyd algorithm for computing centroidal Voronoi tesselations, SAIM J. Numer. Anal. 44 (1) (2006) 102–119.
[43] Q. Du, V. Faber, M. Gunzburger, Centroidal Voronoi tesselations: applications and algorithms, SIAM Rev. 41 (1999) 637–676.
[44] Q. Du, M. Gunzburger, Grid generation and optimization based on centroidal Voronoi tesselations, Appl. Math. Comput. 133 (2002) 591–607.
[45] J.K. Dukowicz, M.C. Cline, F.S. Addessio, A general topology Godunov method, J. Comput. Phys. 82 (1989) 29–63.
[46] J.K. Dukowicz, B. Meltz, Vorticity errors in multidimensional Lagrangian codes, J. Comput. Phys. 99 (1992) 115–134.
[47] J.K. Dukowicz, A simplified adaptive mesh technique derived from the moving finite element method, J. Comput. Phys. 56 (1984) 324–342.
[48] J.K. Dukowicz, J. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (2000) 318–335.
[49] J.K. Dukowicz, J.W. Kodis, Accurate conservative remapping (rezoning) for arbitrary-Lagrangian–Eulerian computations, SIAM J. Stat. Comput. 8 (1987) 305–321.
[50] V.F. Dyachenko, The free point method for problems of continuous media, USSR Comp. Math. Math. Phys. 5 (4) (1965) 680–688.
[51] V.F. Dyachenko, Some new method for numerically solving nonstationary gas dynamics problems, Comput. Meth. Appl. Mech. Eng. 2 (3) (1973) 265–277.
[52] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361–5384.
[53] C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of the flow problems on the moving grids, J. Comput. Phys. 174 (2001) 669–694.
[54] A.P. Favorskij, M. Yu. Shashkov, A.V. Solovjov, V.F. Tishkin, The Dirichlet particle method for the simulation of 2-D gas dynamic flows with strong deformations, in: A.A. Samarskii, M.P.Sapagovas (Eds.), Mathematical Modelling and Applied Mathematics, Proceedings of IMACS International Conference, Moscow/USSR June 18–23, 1990, pp. 153–162 (Elsevier Science Publishers North Holland, Amsterdam, 1992).
[55] M.J. Fritts, W.P. Crowley, H. Trease (Eds.), The Free-Lagrange Method: Proceedings of the First International Conference of Free-Lagrange Methods, Lecture Notes in Physics, vol. 238, Springer-Verlag, New York, 1985.
[56] R.R. Giddings, Mesh movement via optimal transportation, in: Conference on Numerical Methods for Multi-material Flows Czech Technical University in Prague on September 10–14, 2007. Available on web: <http://www-troja.fjfi.cvut.cz/~multimat07/prezentace.html>.
[57] S. Giuliani, An algorithm for continuous rezoning of the hydrodynamic grid in arbitrary-Lagrangian–Eulerian computer codes, Nucl. Eng. Des. 72 (1982) 205–212.
[58] J. Glimm, J. Grove, X.-L. Li, K.-M. Shyue nad, Q. Zhang, Y. Zeng, Three-dimensional front tracking, SIAM J. Sci. Comput. 19 (1998) 703–727.
[59] J. Glimm, J.W. Grove, X.L. Li, W. Oh, D.H. Sharp, A critical analysis of Rayleigh–Taylor growth rates, J. Comput. Phys. 169 (2001) 652–677.

[60] S.K. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko, G. Prokopov, RTsolution numTrique des problFmes multidimensionnels de la dynamique des gaz, Mir, 1979.
[61] J. Grandy, Conservative remapping and region overlays by intersecting arbitrary polyhedra, J. Comput. Phys. 148 (2) (1999) 133–466.
[62] J.-F. Haas, B. Sturtevant, Interaction of weak-shock waves, J. Fluid Mech. 181 (1987) 41–76.
[63] V.M. Hazins, V.V. Svetsov, A conservative stable smoothness-enhancing free-Lagrangian method, J. Comput. Phys. 105 (2) (1993) 187–198.
[64] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary-Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (1974) (reprinted in J. Comput. Phys. 135 (1997) 203–216).
[65] P. Hoch, Mesh quality and conservative projection in Lagrangian compressible hydrodynamic, in: Conference on Numerical Methods for Multi-material Fluid Flows, Czech Technical University in Prague on September 10–14, 2007, <http://www-troja.fjfi.cvut.cz/multimat07/presentations/tuesday/Rebourcet_H>.
[66] W.G. Hoover, Smooth Particle Applied Mechanics: The State of the Art, World Scientific Publishing Co. Pvt. Ltd., 2006.
[67] B.P. Howell, G.J. Ball, Damping of mesh-induced errors in free-Lagrange simulations or Richmayer–Meshkov instability, Shock Waves 10 (2000) 253–264.
[68] B.P. Howell, G.J. Ball, A free-Lagrange augmented Godunov method for simulation of elastic–plastic solids, J. Comput. Phys. 175 (2002) 128–167.
[69] Y. Huang, H. Quin, D. Wang, Centroidal Voronoi tesselation-based finite element superconvergence, Int. J. Numer. Meth. Eng. 76 (2008) 1819–1839.
[70] D.S. Kershaw, M.K. Prasad, M.J. Shaw, J.L. Milovich, 3D unstructured mesh ALE hydrodynamics with the upwind discontinuous finite element method, Comput. Meth. Appl. Mech. Eng. 158 (1998) 81–116.
[71] P. Kjellgren, J. Hyvarien, An arbitrary-Lagrangian–Eulerian finite element method, Comput. Mech. 21 (1998) 81–90.
[72] R. Klein, A. Lingas, A linear-time randomized algorithm for the bounded Voronoi diagram of a simple polygon, Int. J. Comput. Geomet. Appl. 6 (3) (1996) 263–278.
[73] G. Kluth, B. Desprès, Discretization of the hyperelasticity on unstructured mesh with cell-centered scheme, Technical Report R09063, Laboratoire Jacques-Louis Lions, Universitè Pierre et Marie Curie, 2009. Available on web: <http://www.ann.jussieu.fr/publications/2009/R09063.html>.
[74] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 176 (2002) 93–128.
[75] M. Kuchařik, R. Garimella, S. Schofield, M. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, J. Comput. Phys. 229 (7) (2010) 2432–2452.
[76] M. Kuchařik, M. Shashkov, Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity, Int. J. Numer. Meth. Fluid 56 (8) (2007) 1359–1365.
[77] M. Kuchařik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping methods, J. Comput. Phys. 188 (2003) 462–471.
[78] H.J. Kull, Theory of the Rayleigh–Taylor instability, Phys. Rep. 206 (1991) 197–325.
[79] K. Lipnikov, M. Shashkov, The error-minimization-based strategy for moving mesh methods, Commun. Comput. Phys. 1 (1) (2006) 53–81.
[80] G.R. Liu, Mesh Free Methods, Moving Beyond the Finite Element Method, CRC Press, 2003.
[81] R. Lohner, C. Yang, Improved ALE mesh velocities for moving bodies, Commun. Numer. Meth. Eng. 12 (1996) 599–608.
[82] R. Loubère, First steps into ale inc(ubator) – version 2.0.0, Technical Report, Los Alamos National Laboratory Report LAUR-04-8840, 2004.
[83] R. Loubère, E.J. Caramana, The force/work differencing of exceptional points in the discrete, compatible formulation of Lagrangian hydrodynamics, J. Comput. Phys. 216 (1) (2006) 1–18.
[84] R. Loubère, M. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 204 (23) (2004) 155–160.
[85] R. Loubère, M. Staley, B. Wendroff, The repair paradigm: new algorithms and applications to compressible flow, J. Comput. Phys. (2005) 385–404.
[86] P.-H. Maire, A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry, J. Comput. Phys. 228 (2009) 6882–6915.
[87] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (2009) 2391–2425.
[88] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for compressible flow problems, SIAM J. Sci. Comput. 29 (4) (2007) 1781–1824.
[89] P.-H. Maire, J. Breil, A second-order cell-centered Lagrangian scheme for two-dimensional compressible flow problems, Int. J. Numer. Meth. Fluid 56 (2008) 1417–1423.
[90] P.-H. Maire, J. Breil, S. Galera, A cell-centered arbitrary-Lagrangian–Eulerian (ALE) method, Int. J. Numer. Meth. Fluid 56 (2008) 1161–1166.
[91] P.-H. Maire, B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, J. Comput. Phys. 228 (2009) 799–821.
[92] L.G. Margolin, M. Shashkov, P. Smolarkiewicz, A discrete operator calculus for finite difference approximations, Comp. Meth. Appl. Mech. Eng. 187 (2000) 365–383.
[93] L.G. Margolin, Introduction to "An arbitrary-Lagrangian–Eulerian computing method for all flow speeds", J. Comput. Phys. 135 (1997) 198–202.
[94] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[95] N.V. Mikhailova, V.F. Tishkin, N.N. Tyurina, A.P. Favorskii, M.Y. Shashkovm, Numerical modelling of two-dimensional gas-dynamic flows on a variable-structure mesh, USSR Comput. Math. Math. Phys. 26 (5) (1988) 74–84.
[96] A. Okabe, B. Boots, K. Sugihara, S.N. Chiu, Spatial Tesselations, Concepts and Applications of Voronoi Diagrams, Wiley, 2000.
[97] T.I. Orlova, V.V. Svetsov, Deep penetration of projectile material into the planet during the impact, Earth Moon Planets 71 (3) (1995) 255–263.
[98] P.J. O'Rourke, M.S. Sahota, A variable explicit/implicit numerical method for calculating advection on unstructured meshes, J. Comput. Phys. 143 (1998) 312–345.
[99] J.R. Pasta, S. Ulam, Heuristic numerical work in some problems of hydrodynamics, Math. Tables Aids Comput. XIII (65) (1959) 1–12.
[100] J.S. Peery, D.E. Carroll, Multi-material ALE methods in unstructured grids, Comput. Meth. Appl. Mech. Eng. 187 (2000) 591–619.
[101] R. Pember, R. Anderson, Comparison of direct Eulerian Godunov and Lagrange plus remap artificial viscosity schemes for compressible flow, Technical Report 2001-2644, AIAA, 2001.
[102] I.G. Pushkina, V.F. Tishkin, Adaptive grids from Dirichlet cells for mathematical physics problems: a methodology for grid generation, examples, Math. Model. 12 (3) (2000) 97–109.
[103] J. Quirk, S. Karni, On the dynamics of a shock–bubble interaction, J. Fluid Mech. 318 (1996) 129–163.
[104] B. Rebourcet, Comments on the filtering of numerical instabilities in Lagrangian hydrocodes, in: Conference on Numerical Methods for Multi-material Fluid fLows, Czech Technical University in Prague on September 10–14, 2007, <http://www-troja.fjfi.cvut.cz/multimat07/presentations/tuesday/Rebourcet_f>.
[105] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 121 (1998) 112–152.
[106] M. Rudman, Volume tracking methods for interfacial flow calculations, Int. J. Numer. Meth. Fluid 24 (1997) 671–691.
[107] M.S. Sahota, H.E. Trease, A three-dimensional free-Lagrange code for multimaterial flow simulations, Report LAUR-90-3383, Los Alamos National Laboratory, 1990.
[108] G. Scovazzi, Stabilized shock hydrodynamics: II. Design and physical interpretation of the SUPG operator for Lagrangian computations, Comput. Meth. Appl. Mech. Eng. 196 (2007) 966–978.

[109] G. Scovazzi, M.A. Christon, T.J.R. Hughes, J.N. Shadid, Stabilized shock hydrodynamics: I. A Lagrangian method, Comput. Meth. Appl. Mech. Eng. 196 (2007) 923–966.
[110] G. Scovazzi, E. Love, M.J. Shashkov, Multi-scale Lagrangian shock hydrodynamics on Q1/P0 finite elements: theoretical framework and two-dimensional computations, Comput. Meth. Appl. Mech. Eng. 197 (2008) 1056–1079.
[111] M. Shashkov, Closure models for multimaterial cells in arbitrary-Lagrangian–Eulerian hydrocodes, Int. J. Numer. Meth. Fluid 56 (2007) 1497–1504.
[112] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, J. Comput. Phys. 198 (1) (2004) 265–277.
[113] M.Yu. Shashkov, A.V. Solovjov, A generalization of the notion of Dirichlet cell for non-convex domain, Preprint No. 32 of Keldysh Institute of Applied Mathematics, USSR Academy of Sciences, Moscow, Russia, 1990 (in Russian).
[114] M.Yu. Shashkov, A.V. Solovjov, Numerical simulation of two-dimensional flows by the free-Lagrangian flow simulations, Report TUM-M9105, Mathematisches Institut, Technische Universität, Munchen, 1991. Available at <http://cnls.lanl.gov/~shashkov>.
[115] I.D. Sofronov, V.V. Rasskazova, L.V. Nesterenko, The use of nonregular nets for solving two-dimensional non-stationary problems in gas dynamics, in: N.N. Yanenko, Yu.I. Shokin (Eds.), Numerical Methods in Fluid Dynamics, MIR, Moscow, 1984, pp. 82–121.
[116] S.S. Sokolov, A.I. Panov, A.A. Voropinov, I.G. Novikov, I.V. Sobolev, A.V. Yalozo, Method TIM for computation of three-dimensional problems of mechanics of continuum media using unstructured polyhedral Lagrangian meshes (in Russian), Quest. Atomic Sci. Tech. Ser.: Math. Mod. Phys. Process. 3 (2005) 37–52.
[117] S.S. Sokolov, A.A. Voropinov, I.G. Novikov, A.I. Panov, I.V. Sobolev, A.A. Pushkarev, Method TIM-2D for computation of problems of mechanics of continuum media using unstructured polygonal with arbitrary connections at the nodes (in Russian), Quest. Atomic Sci. Tech. Ser.: Math. Mod. Phys. Process. 4 (2006) 29–44.
[118] A. Solovjov, E. Solovjova, M. Shashkov, V. Tishkin, A. Favorskii, Approximation of finite difference operators on mesh of Dirichlet cells, Differ. Equat. 22 (1986) 863–872.
[119] V. Springel, E pur si muove: Galiliean-invariant cosmological hydrodynamical simulations on a moving mesh, Mon. Not. R. Astron. Soc. (in press), arXiv:0901.4107, <http://arxiv.org/pdf/0901.4107>.
[120] C. Stoker, C. Gay, F. Bay, J.-L. Chenot, A velocity approach for the ALE-method applied for 2D and 3D problems, in: Simulation of Material Processing: Theory, Methods and Applications, Balkema, Rotterdam, 1998.
[121] H. Trease, M.J. Fritts, W.P. Crowley (Eds.), Advances in the Free-Lagrange Method: Including Contributions on Adaptive Gridding and the Smooth Particle Hydrodynamics Method: Proceedings of the Next Free-Lagrange Conference, Lecture Notes in Physics, vol. 395, Springer-Verlag, New York, 1990.
[122] H.E. Trease, Three-dimensional free-Lagrangian method hydrodynamics, Comput. Phys. Commun. 48 (1988) 39–50.
[123] B. van Leer, Towards the ultimate conservative difference scheme, J. Comput. Phys. 32 (1979) 101–136.
[124] W.B. VanderHeyden, B.A. Kashiwa, Compatible fluxes from van Leer advection, J. Comput. Phys. 146 (1998) 1–28.
[125] J. von Neumann, R.D. Richtmyer, A method for the numerical calculations of hydrodynamical shocks, J. Appl. Phys. 21 (1950) 232–238.
[126] R. Whitehurst, A free-Lagrange method for gas dynamics, Mon. Not. R. Astron. Soc. 277 (1995) 655–680.
[127] M.L. Wilkins, Calculation of elastic plastic flow, Math. Comput. Phys. 3 (1964).
[128] A.M. Winslow, Equipotential zoning of two-dimensional meshes, Technical Report UCRL-7312, Lawrence Livermore National Laboratory, 1963.
[129] A.M. Winslow, Numerical solution of the quasilinear Poisson equations in a nonuniform triangle mesh, J. Comput. Phys. 1 (1966) 149–172.