Journal of Computational Physics 322 (2016) 603–665

Contents lists available at ScienceDirect

Journal of Computational Physics

www.elsevier.com/locate/jcp

Arbitrary Lagrangian–Eulerian methods for modeling high-speed compressible multimaterial flows

Andrew J. Barlow [a], Pierre-Henri Maire [b], William J. Rider [c], Robert N. Rieben [d], Mikhail J. Shashkov [e,*]

[a] AWE, Aldermaston, UK
[b] CEA/CESTA, Le Barp, France
[c] Sandia National Laboratory, Albuquerque, USA
[d] Lawrence Livermore National Laboratory, Livermore, USA
[e] Los Alamos National Laboratory, Los Alamos, USA

ARTICLE INFO

Article history:
Received 24 May 2016
Received in revised form 30 June 2016
Accepted 1 July 2016
Available online 11 July 2016

Keywords:
Lagrangian hydrodynamics
Arbitrary-Lagrangian–Eulerian hydrodynamics
Rezone
Remap
Multimaterial cells
Interface-aware sub-scale closure models
Staggered discretization
Cell-centered discretization
High-order methods

ABSTRACT

This paper reviews recent developments in Arbitrary Lagrangian Eulerian (ALE) methods for modeling high speed compressible multimaterial flows in complex geometry on general polygonal meshes. We only consider the indirect ALE approach which consists of three key stages: a Lagrangian stage, in which the solution and the computational mesh are updated; a rezoning stage, in which the nodes of the computational mesh are moved to improve grid quality; and a remapping stage, in which the Lagrangian solution is transferred to the rezoned mesh.

1. Introduction

This paper reviews Arbitrary Lagrangian Eulerian methods for modeling high speed compressible multimaterial flows in complex geometry. The underlying numerical methods have been designed, analyzed and developed by the authors in the context of real life applications.

The page limit does not allow us to cover the history of such methods or all the topics relevant to such modeling. For example, we only describe methods for fluids and do not consider solids. However, we give relevant references for topics not covered in this paper.

Prior to the 1974 paper by Hirt et al. [96] (see also [6]) published in Journal of Computational Physics, the computational fluid dynamicist was limited to Lagrangian and Eulerian methods. In Lagrangian simulations, the mesh moves with the fluid velocity. Lagrangian methods maintain good resolution during large scale compression/expansion and are well-suited to

* Corresponding author.
E-mail addresses: Andy.Barlow@awe.co.uk (A.J. Barlow), Pierre-Henri.MAIRE@CEA.FR (P.-H. Maire), wjrider@sandia.gov (W.J. Rider), rieben1@llnl.gov (R.N. Rieben), shashkov@lanl.gov (M.J. Shashkov).

http://dx.doi.org/10.1016/j.jcp.2016.07.001
0021-9991/

maintaining material interfaces. It is also easy to specify boundary conditions on moving boundaries, such as free boundaries and pistons. A particular advantage of the Lagrangian representation is that it is the only frame of reference for which the advective terms vanish identically. On the other hand, multidimensional Lagrangian meshes tend to tangle and in general cannot represent large deformation, shear and vorticity. In Eulerian simulations, the mesh is fixed in space and the fluid moves through it. Eulerian meshes are not subject to tangling, but can produce diffusive solutions and it is difficult to maintain sharp material interfaces.

The Lagrangian and Eulerian representations are only two special types of mesh motion. The essence of the ALE idea is that the mesh motion can be chosen arbitrarily, providing additional flexibility and accuracy. The philosophy of the arbitrary Lagrangian–Eulerian methodology (ALE; cf. [96,24,141,183,142]), is to exploit this degree of freedom to improve the robustness, accuracy and efficiency of the simulation. The idea of using both frames of reference in a single calculation was perhaps pioneered by W.F. Noh in 1963 [152].

In this paper we will only consider the three stage ALE scheme,[1] which is used in practical calculations of multimaterial flows. These stages are: a Lagrangian stage, in which the solution and the computational mesh are updated; a rezoning stage, in which the nodes of the computational mesh are moved to more optimal positions; and a remapping stage, in which the Lagrangian solution is transferred to the rezoned mesh.

We would like to emphasize that the three stage approach does not have any *splitting* errors. It is very easy to understand this if one considers a thought experiment, where an infinitely accurate Lagrangian discretization is combined with an infinitely accurate remapping technique, then the overall result will also be infinitely accurate, so no additional splitting errors will be introduced.

The seminal paper by Hirt et al. [96], has been cited more than 2200 times and was reprinted in 1997 [97] as one of most cited papers in Journal of Computational Physics (the reprinted paper has been cited more than 200 times itself) with the Introduction by L. Margolin, [141], who briefly summarized progress to date and remaining issues. The last comprehensive review paper on Lagrangian and Eulerian hydrocodes, which includes ALE methods was written by D. Benson, [24] in 1992. It is 160 pages long and has been cited more than 900 times. Since these papers have been published there has been a great deal of further progress and new developments related to multimaterial ALE methods, hence, the need for this paper.

1.1. Setup

The statement of any continuum hydrodynamic problem requires the specification of the solution domain, initial and boundary conditions. For multimaterial problems one also needs to specify material subregions occupied by the materials. In this paper we only consider situations with distinct material interfaces between different materials.

The first step for any mesh-based numerical method is to create the computational mesh and specify the initial and boundary conditions on this mesh. The methods for mesh generation in complex domains is a huge area of research in itself and is beyond the scope of this paper – the interested reader can refer to [108,52,172,57]. For multimaterial problems one usually tries to construct an initial mesh which is conforms to the material interfaces, however, there are situations when this is not possible. There can be several reasons for this, for example, the geometry of the material interfaces may be very complicated or some features may be too small to resolve even with unstructured meshes aligned with interfaces, and even if it is possible the resulting conforming mesh may be prohibitively fine, or of low geometric quality. The standard way to resolve this situation is to introduce so-called *multimaterial* cells, which contain several materials as opposed to *pure* cells which contain only one material. The multimaterial cells are defined by the exact or approximate intersections of the material subregions. This allows the volume and mass of each material component within each multimaterial cell to be defined along with their initial state variables.

There are two different ways of representing materials in a multimaterial cell and both approaches are described in this paper. In the first approach, the multimaterial cell only carries information about the relative amounts of each material within each multimaterial cell, the geometry of the materials is not represented. In the second approach, the interfaces inside multimaterial cell are represented approximately by using so-called interface reconstruction algorithms.

The choice made for the representation of the materials in the multimaterial cell then dictates the approaches taken for the closure model required to evolve the material parameters in the Lagrangian stage as well as the methods used for remapping of the material locations during remap stage.

1.2. Lagrangian phase

The main ideas of Lagrangian hydrodynamics are summarized in Section 2.

Staggered-Grid Lagrangian Hydrodynamics

In Section 4 we will present the modern approach to low-order *staggered discretization* of the Lagrangian equations (SGH – Staggered Grid Hydrodynamics) on general polygonal meshes, which is a workhorse for many production codes, [41,48]. In SGH all the thermodynamic variables of the fluid, such as density, pressure, and internal energy are cell-centered, but velocity is defined at the nodes of the mesh – hence the term staggered.

[1] The three stage ALE scheme is sometimes called indirect ALE as opposed to direct ALE methods in which advective terms are explicitly included in governing equations.

The main requirements for the Lagrangian phase are the following: local mass conservation (which defines the Lagrangian representation); conservation of momentum, conservation of total energy (which is critical for the correct modeling of flows with shocks), and entropy increase for flows with shocks. In SGH the mass of each material is constant by definition and the density of each material defined from its mass and current volume. The momentum equation is discretized in conservative form, which guarantees conservation of momentum. One of the distinct features of the SGH discretization is that it uses an equation for the evolution of internal energy, which expresses conservation of entropy for smooth flows. There are two implications of such a choice. First, one needs to use the theory of mimetic discretizations, [161,159,160,166,119], to construct discrete analogs of differential operators, and/or a compatible discretization [92,100,41,48] in order to conserve total energy – Sections 4.2, 4.3, 4.4, 4.6. Second, it requires introduction of artificial viscosity which has to work only for non-smooth flows and converts kinetic energy into internal energy and therefore increase entropy for flows with shocks – Section 4.8.

We are interested in the multimaterial case, where each computational cell may be pure and contain only one material or maybe multimaterial and contain several different materials. In Lagrangian hydrodynamics the mass of each material inside each multimaterial cell does not change in time. We assume that all materials have the same velocity, but each material has its own density, internal energy and pressure. Every cell produces forces that act at its nodes, which are the result of the so-called closure model, which is also responsible for the time evolution of material properties within the multi-material cell, [167,14]. The Lagrangian stage starts by updating the nodal velocities as a result the application of the zonal forces. The new nodal velocities are then used to update the volumes of all computational cells. For the pure cells, this allows the density and internal energy of the material to be updated. For multi-material cells, the situation is more complicated. First, one needs to decide how to distribute the change in total volume of the multi-material cell between material components (which in turn allows the densities for each material component to be updated). Second, one needs to decide how to update internal energy of each material component. Finally one needs to decide how to define the common pressure, which will be used in the computation of the forces to be used to solve the momentum equation on the next time step. All these questions have to be answered by an appropriate closure model. We give examples of closure models in Section 4.11.

For the low-order SGH discretization we use a predictor–corrector scheme for the time integration, which does not require a global system of equations to be solved.

The bane of Lagrangian hydrodynamics calculations, as described in [50], is the premature breakdown of mesh quality and mesh tangling, that results in severe degradation of accuracy and in run termination, that often occurs long before the assumption of a Lagrangian zonal mass has ceased to be valid. At short spatial grid scales this is usually referred to by the term hourglass mode. This is the motion associated with under constrained grids constructed from quadrilaterals and hexahedrons in two and three dimensions, respectively. At longer spatial length scales relative to the grid spacing there is what is referred to ubiquitously as spurious vorticity, or the long-thin zone problem. In both cases the result is anomalous grid distortion and tangling that has nothing to do with the correct physically relevant solution. In Section 4.7 we show how such motions can be eliminated by the proper use of subzonal Lagrangian masses, with associated densities and pressures.

Cell-centered Grid Lagrangian Hydrodynamics

In Section 5 we describe the co-located Finite Volume (FV) discretization of the Lagrangian gas dynamics equations on moving grids – cell-centered Grid Hydrodynamics (CGH). In this approach, the conservation laws of mass, momentum and total energy are written for a co-moving control volume, *i.e.*, a zone moving with the local fluid velocity, by means of the well known Reynolds transport formula. The change in volume with respect to time of this zone is given by the Geometrical Conservation Law (GCL), which is a trivial by-product of the Reynolds transport formula. The system of conservation laws to be solved is closed by an equation of state, which allows the thermodynamic variables such as the density, pressure and specific internal energy to be related. It is well known that this set of conservation laws (mass, momentum, total energy and GCL) can have discontinuous solutions, such as contact discontinuities and shock waves. The selection of physically correct discontinuous solutions requires the second law of thermodynamics. At a shock front kinetic energy is converted into internal energy increasing entropy in a dissipative manner, provided that the equation of state satisfies a reasonable convexity assumption, refer to [148,29].

CGH is a moving mesh FV discretization of both the physical conservation laws (mass, momentum, total energy) and the GCL. It does not only rely on the numerical approximation of cell interfaces fluxes but also on the determination of the nodal velocities required to move the grid. At this point, it is worth emphasizing the two main differences between the SGH and the CGH approaches. First, the SGH method solves an internal energy equation, which is not written in conservative form, whereas the CGH method solves the total energy equation, which is written in conservative form. Second, in the SGH method velocity is defined at the nodes and it is a primary variable, which directly results from momentum equation, whereas in CGH method the nodal velocity is *not* a primary variable and thus one needs an additional procedure to compute it. This procedure must ensure the compatibility between the numerical approximation of the cell interfaces fluxes and the nodal velocity in such a way that the discrete GCL is rigorously satisfied. We point out that for the SGH approach the discrete GCL is always satisfied since the nodal velocity is a primary variable.

This issue was successfully solved at the beginning of the 2000's in the ground breaking papers [68,69]. The corresponding CGH method is called GLACE (Godunov-type LAgrangian scheme Conservative for total Energy) and is characterized by a node-based numerical flux, *i.e.*, a unique continuous nodal velocity and one nodal pressure per cell corner. A local entropy inequality is obtained expressing the pressure jump in terms of the velocity jump at each cell corner by means of a Riemann invariant-type relation. Conservation of momentum and total energy is obtained from prescribing a node-centered

instantaneous balance equation. Its unique solution provides the nodal velocity required to move the computational grid consistently with the GCL.

Subsequently, an alternative scheme named EUCCLHYD (Explicit Unstructured Cell-Centered Lagrangian HYDrodynamics) has been proposed in [2,137], which is also based on a node-centered solver, but characterized by two nodal pressures per cell corner.

We conclude this paragraph by mentioning that the co-located placement of the primary unknowns greatly facilitates the indirect ALE extension, since it relies on a unique cell-centered remapping procedure.

High-order Finite Element SGH

In Section 6 we present an alternative method to solving the Euler equations in a moving Lagrangian frame based on the use of (arbitrarily) *high-order finite elements*, as detailed in [73]. The method is built around the notion of general high-order polynomial basis function representations for the various Lagrangian state variables and can be formulated for any finite dimensional approximation of the kinematic and thermodynamic fields, including generic finite elements on 2D and 3D meshes with triangular, quadrilateral, tetrahedral or hexahedral zones. The kinematic variables of position and velocity are discretized using a continuous high-order basis function expansion of arbitrary polynomial degree which is obtained via a corresponding high-order parametric mapping from a standard reference element. This enables the use of curvilinear zone geometry (also explored in [140,54]), higher-order approximations for fields within a zone and a pointwise definition of mass conservation which we refer to as strong mass conservation. The internal energy is discretized using a piecewise discontinuous high-order basis function expansion which is also of arbitrary polynomial degree. This facilitates multi-material hydrodynamics by treating material properties, such as equations of state and constitutive models, as piecewise discontinuous functions which vary within a zone. This method can be viewed as the high-order generalization of the SGH approach.

1.3. Interface reconstruction

In the Lagrangian phase the shape of all the cells has changed and the parameters of the materials in the multimaterial cells have been updated. In particular, the volume of the materials have changed as well as the positions of the centroids. Therefore, if the ALE method uses pure material polygons inside the multimaterial cells, then these polygons have to be reconstructed to correctly represent the material interfaces after the Lagrangian stage. The representation of the materials in the multimaterial cells after Lagrangian step is required as input data for the multimaterial remapping from the Lagrangian to the rezoned mesh.

Modern interface reconstruction methods use linear representations for the material interfaces, [156,154,82,5]. The input data for these methods is the volumes of the materials in each multimaterial cell (volume of fluid methods – VOF, [156]) and sometimes additional information such as the centroids of the materials to provide a reference position (moment of fluid methods – MOF, [82,5]). These methods reconstruct material interfaces in each cell of the mesh, such that each material is represented by material polygons, which contain only one material. A brief description of the interface reconstruction methods is given in Section 7.

1.4. Rezone phase

The goal of the rezoning is to *improve* the mesh after the Lagrangian step. In general, the meaning of the *improvement* is goal and problem dependent, and it is not possible to describe all the approaches in this paper.

In a real complex ALE simulation the most basic goal of rezoning is simply to run the calculation to completion without any user intervention and still achieve reasonable accuracy. The mesh movement philosophy applied to most applications, related to high-speed multimaterial flows, [15,16,98], is to develop algorithms that will move the mesh in such a way as to maintain mesh smoothness while staying as close as possible to the Lagrangian mesh motion – see seminal papers by A. Winslow, [180] and by J. Brackbill, and J. Saltzman, [38].

Ideally, the rezoning strategy has to be based on some optimization framework with a clear objective and guarantee validity of the resulting rezoned mesh. In our opinion the target matrix paradigm described in a series of papers by P. Knupp et al. [106,105,107,109] is an ideal framework for rezoning. In Section 8 we will describe one method based on this paradigm, the so-called, reference Jacobian approach, [107], which is best suited for ALE methods.

Each of these approaches can also be applied to more general, high-order meshes obtained from a high-order parametric mapping. Techniques such as harmonic smoothing can be formulated for arbitrarily high-order curved meshes based on a simple linear iteration scheme and an appropriately defined “mesh Laplacian” operator. More advanced non-linear techniques can be formulated by treating the mesh relaxation process as an integral minimization problem which requires definition of an “energy function” and use of a Newton iteration. In this setting, the classic inverse harmonic method of Winslow can extended to arbitrarily high-order curved meshes by defining the energy function in terms of the inverse high-order parametric mapping function.

1.5. Remap phase

An important aspect of ALE methods is the techniques used for remapping variables from the Lagrangian to the rezoned mesh. The remap should be innocuous and complementary to the other steps in the overall algorithm. While it is a step in

the method, it is not a separate splitting of the physical laws, but rather a principled transfer of information from a mesh resulting from Lagrangian step to the mesh obtained from the rezone step. As such the remap step contains no physics and does not involve any time advancing of physical quantities, but still must abide by essential physical and numerical principles [143]. The physics in ALE methods occurs solely in the Lagrangian step, and distinct physical effects are to be stringently avoided in the remap.

Let us list the most important requirements for the remap. The remap has to *conserve* the same quantities and in the same form as Lagrangian step does – conservation of mass, momentum and total energy – to ensure the conservation of the overall ALE method. If we consider the multimaterial case then the volume and the mass of each material component has to be conserved as well. The remapped quantities have to satisfy *bound-preservation* property, that is, they are suppose to stay within physically justified bounds. The remap has to be as accurate as possible, however, the accuracy of the remap also has to be consistent with the accuracy of the Lagrangian step.

The choice of algorithm for the remapping depends on the relation between the Lagrangian and the rezoned meshes. If two meshes have the same connectivity and the cells of the rezoned mesh are just slightly displaced with respect to the corresponding cells of the Lagrangian mesh, then one can use flux-based methods, based on the advection analogy. However, if the cells of the meshes with the same connectivity are arbitrarily displaced with respect to each other or the Lagrangian and the rezoned meshes are completely independent of one another then one needs to use intersection-based (overlay) remap. This involves finding the intersections of the cells of two meshes.

All remapping methods start with function reconstruction on the Lagrangian mesh. In the simplest case this is a piecewise linear reconstruction from mean values over the cells obtained from the Lagrangian step. Most of these reconstruction methods require some limiting procedures for the slope to guarantee bound preservation of the remap.

After reconstruction on Lagrangian mesh is done, one performs exact or approximate integration of the reconstructed function over the cells of rezoned mesh.

In the high-order finite element approach, the remap phase is based on a high-order discontinuous Galerkin (DG) formulation of the advection equation for each conserved variable. Semi-discrete DG methods for advection equations can be formulated in terms of high-order finite element “mass” and “advection” matrices. These formulations result in high-order accuracy for sufficiently smooth fields, but produce non-monotonic results (spurious oscillations) for discontinuous fields. We therefore employ techniques from Flux Corrected Transport (FCT) methods which start with a low-order, strictly monotone method (based on a fully lumped mass matrix and a fully “up-winded” advection matrix) and add in “anti-diffusive”, conservative, high-order flux corrections in a non-linear manner to recover high-order accuracy whenever the underlying solution is sufficiently smooth. The remap stage of ALE method described in details in Section 9.

We have arranged the paper as follows. Section 2 introduces the main ideas of continuum Lagrangian hydrodynamics: continuum mechanics equations written under Lagrangian, form, kinematics, conservation laws in integral form and differential forms, thermodynamics, time evolution of mass averaged quantities; and gives a summary of the governing equations. In Section 3 we describe polygonal mesh and its dynamics. The discrete staggered grid Lagrangian Hydrodynamics is described in Section 4. The discrete cell-centered grid Lagrangian hydrodynamics described in Section 5. The high-order finite element approach discrete staggered grid Lagrangian hydrodynamics is described in Section 6. The interface reconstruction described in Section 7. The methods related to rezoning are described in Section 8. The different aspects of the remapping are covered in Section 9. It includes the remap of cell-centered and nodal quantities between different type of meshes, methods for enforcing bound preservation and multimaterial remapping, as well as high-order Discontinuous Galerkin advection-based remap for high-order SGH. Acknowledgment is given in Section 11.

2. Continuum mechanics equations in Lagrangian form

In this Section we aim to recall briefly the basic notions required to derive the conservation laws of continuum mechanics in Lagrangian form. The interested reader can find a detailed presentation of this topic in [93].

2.1. Kinematics

We consider a continuous medium deforming in time. The initial configuration of this medium is denoted by $\omega(0)$, which is a subregion of the d-dimensional space $\mathbb{R}^d$ (here d is the dimension of space – in this paper we only consider the two-dimensional case – $d = 2$) and the configuration at time $t > 0$ is denoted by $\omega(t) \subset \mathbb{R}^d$. The coordinates $\mathbf{X}$ of points in $\omega(0)$ are used to identify material particles throughout the motion. The motion of the deforming medium is described by the deformation mapping

$$\begin{aligned} \boldsymbol{\Phi}&: \omega(0) \to \omega(t) \\ &\mathbf{X} \mapsto \mathbf{x} = \boldsymbol{\Phi}(\mathbf{X}, t). \end{aligned}$$

Here, $\mathbf{x}$ is the spatial position of a material particle $\mathbf{X}$ at time t. It is worth mentioning that $\mathbf{X}$ and $\mathbf{x}$ are respectively called the Lagrangian and the Eulerian coordinates and the deformation mapping, $\boldsymbol{\Phi}$, is called the Lagrange–Euler map. We assume that $\boldsymbol{\Phi}$ is a continuous and sufficiently smooth bijective function. Thus, one might characterize the local deformation of an infinitesimal fiber by means of the deformation gradient tensor

$$\mathbb{F}(\mathbf{X},t)=\frac{\partial\boldsymbol{\Phi}}{\partial\mathbf{X}}(\mathbf{X},t). \tag{2.1}$$

The deformation mapping is assumed to be orientation preserving, that is it satisfies

$$J(\mathbf{X},t)=\det\mathbb{F}(\mathbf{X},t)>0\,.$$

Here, J denotes the Jacobian of the deformation mapping, which measures the ratio of the deformed to the undeformed volume of an infinitesimal neighborhood of material. Observing that $\boldsymbol{\Phi}(\mathbf{X},0)=\mathbf{X}$ leads to $\mathbb{F}(\mathbf{X},0)=\mathbb{I}_d$ and $J(\mathbf{X},0)=1$.

At this point we can introduce the two usual descriptions of the motion:

- The Lagrangian description, otherwise called the material description, consists of observing the medium by following the motion of the material particles from their initial location. The independent variables used for this description are $\mathbf{X}$ and t.
- The Eulerian description, otherwise called the spatial description, consists of observing the medium at fixed locations in space. The independent variables used for this description are $\mathbf{x}$ and t.

Let us consider a physical quantity attached to material particles (parcels) such as density for instance. Its Lagrangian (material) description is written as $F(\mathbf{X},t)$, whereas as its Eulerian (spatial description) is denoted by $f(\mathbf{x},t)$. Obviously, the material and the spatial descriptions of this physical quantity are related by

$$f(\mathbf{x},t)=F[\boldsymbol{\Phi}^{-1}(\mathbf{x},t),t], \tag{2.2a}$$

$$F(\mathbf{X},t)=f[\boldsymbol{\Phi}(\mathbf{X},t),t], \tag{2.2b}$$

where $\boldsymbol{\Phi}^{-1}$ denotes the inverse of the deformation mapping defined by $\mathbf{X}=\boldsymbol{\Phi}^{-1}(\mathbf{x},t)$. In what follows, when there is no danger of confusion, we will use the same symbol to denote both the Lagrangian and the Eulerian description. That is we shall use the trivial identity $F(\mathbf{X},t)=f(\mathbf{x},t)$.

Employing the foregoing notations, we define the material time derivative of F, which measures the rate of change of F following a material particle along its motion

$$\frac{dF}{dt}=\left(\frac{\partial F}{\partial t}\right)_{\mathbf{X}}(\mathbf{X},t), \tag{2.3}$$

where the subscript at the right-hand side emphasizes the fact that the time derivative is taken holding $\mathbf{X}$ fixed. Computing the material derivative of $\mathbf{x}=\boldsymbol{\Phi}(\mathbf{X},t)$ allows us to define the material velocity field

$$\mathbf{u}(\mathbf{X},t)=\left(\frac{\partial\boldsymbol{\Phi}}{\partial t}\right)_{\mathbf{X}}(\mathbf{X},t). \tag{2.4}$$

The corresponding spatial velocity field is denoted by $\mathbf{u}(\mathbf{x},t)$ and $\mathbf{x}$ satisfies

$$\frac{d\mathbf{x}}{dt}=\mathbf{u}(\mathbf{x},t),\quad \mathbf{x}(0)=\mathbf{X}. \tag{2.5}$$

This ordinary differential equation is the trajectory equation which provides the path of the material particles.

Finally, utilizing (2.2b) and the chain rule, the material derivative of the spatial description can be written

$$\frac{df}{dt}(\mathbf{x},t)=\left(\frac{\partial f}{\partial t}\right)_{\mathbf{x}}(\mathbf{x},t)+\mathbf{u}(\mathbf{x},t)\cdot\nabla f(\mathbf{x},t), \tag{2.6}$$

where ∇f denotes the gradient of f with respect to the spatial coordinate. Obviously the material derivatives of a physical quantity employing either the Lagrangian or the Eulerian description coincide, *i.e.*, $\frac{dF}{dt}(\mathbf{X},t)=\frac{df}{dt}(\mathbf{x},t)$.

2.2. Conservation laws – integral form

Consider $\omega(t)$ a fluid parcel (spatial region convecting with the deforming medium), then its mass, $m_{\omega(t)}$, momentum $\boldsymbol{\mu}_{\omega(t)}$ and total energy $E_{\omega(t)}$ are respectively defined by

$$m_{\omega(t)}=\int_{\omega(t)}\rho\,\mathrm{d}v,\quad \boldsymbol{\mu}_{\omega(t)}=\int_{\omega(t)}\rho\,\mathbf{u}\,\mathrm{d}v,\quad E_{\omega(t)}=\int_{\omega(t)}\rho\,e\,\mathrm{d}v, \tag{2.7}$$

where ρ and $e=\varepsilon+|\mathbf{u}|^2/2$ are respectively the density and the specific total energy, *i.e.*, the total energy per unit mass, here ε is the specific internal energy. In the absence of body force, heat sources and heat conduction the conservation laws of mass, momentum and total energy can be written as [93]

$$\frac{dm_{\omega(t)}}{dt} = \frac{d}{dt}\int\limits_{\omega(t)} \rho \,\mathrm{d}v = 0, \tag{2.8a}$$

$$\frac{d\boldsymbol{\mu}_{\omega(t)}}{dt} = \frac{d}{dt}\int\limits_{\omega(t)} \rho\,\mathbf{u}\,\mathrm{d}v = \int\limits_{\partial\omega(t)} \mathbb{T}\mathbf{n}\,\mathrm{d}s\,, \tag{2.8b}$$

$$\frac{dE_{\omega(t)}}{dt} = \frac{d}{dt}\int\limits_{\omega(t)} \rho\, e\,\mathrm{d}v = \int\limits_{\partial\omega(t)} \mathbb{T}\mathbf{n}\cdot\mathbf{u}\,\mathrm{d}s\,, \tag{2.8c}$$

where $\partial\omega(t)$ denotes the boundary of $\omega(t)$ and $\mathbf{n}$ its unit outward normal. In the momentum equation (2.8b), $\mathbb{T}\mathbf{n}$ represents the force per unit area exerted on the surface element and $\mathbb{T}$ is the Cauchy stress tensor, which must be symmetric to ensure angular momentum balance, refer to [93]. In the total energy equation (2.8c), $\mathbb{T}\mathbf{n}\cdot\mathbf{u}$ is the rate of doing work for this surface force. The path of a point located on the boundary of $\omega(t)$, *i.e.*, $\mathbf{x}\in\partial\omega(t)$, is determined by the trajectory equation (2.5).

The mass conservation equation (2.8a) amounts to writing that the mass of the convecting region $\omega(t)$ is constant and equal to the mass contained in $\omega(0)$, namely $m_{\omega(t)} = m_{\omega(0)}$ for all $t > 0$. *This is one of the main features of the Lagrangian representation.*

In what follows, we shall focus on gas dynamics. In this case the Cauchy stress tensor is spherical and has the form $\mathbb{T} = -p\,\mathbb{I}_d$, where p is the thermodynamic pressure.

In the case of gas dynamics the conservation of the momentum and total energy can be written as follows

$$\frac{d\boldsymbol{\mu}_{\omega(t)}}{dt} = -\int\limits_{\partial\omega(t)} p\,\mathbf{n}\,ds\,, \quad \frac{dE_{\omega(t)}}{dt} = -\int\limits_{\partial\omega(t)} p\,\mathbf{u}\cdot\mathbf{n}\,ds\,. \tag{2.9}$$

2.3. Thermodynamics

In order to achieve the thermodynamic closure of the above system of conservation laws, we can write down relationships between the thermodynamic variables ρ, p and $\varepsilon = e - \frac{1}{2}\mathbf{u}^2$. The thermodynamic properties of a material are embodied in the relation $\varepsilon = \varepsilon(\rho,\eta)$ expressing the specific internal energy in terms of the density and the specific entropy η, refer to [148]. The pressure $p = p(\rho,\eta)$ and the temperature $\theta = \theta(\rho,\eta)$ are given as first derivatives of the specific internal energy

$$p(\rho,\eta) = \rho^2\left(\frac{\partial\varepsilon}{\partial\rho}\right)_\eta \text{ and } \theta(\rho,\eta) = \left(\frac{\partial\varepsilon}{\partial\eta}\right)_\rho, \tag{2.10}$$

in accordance with the fundamental Gibbs relation

$$\mathrm{d}\varepsilon = -p\,\mathrm{d}\left(\frac{1}{\rho}\right) + \theta\mathrm{d}\eta. \tag{2.11}$$

The set of relations (2.10) is called the complete equation of state. Assuming $\theta = \frac{\partial\varepsilon}{\partial\eta} > 0$ allows us to express η in terms of ε by inverting $\varepsilon = \varepsilon(\rho,\eta)$ and hence we can write the equation of state in incomplete form $p = p(\rho,\varepsilon)$. We also assume that the equation of state is such that $(\frac{\partial p}{\partial\rho})_\eta > 0$. This enables us to define the isentropic sound speed

$$a = \sqrt{\left(\frac{\partial p}{\partial\rho}\right)_\eta}. \tag{2.12}$$

This in turn ensures the hyperbolicity of the gas dynamics equations, refer to [148].

It is important to notice that the system of conservation laws (2.8) can admit discontinuous solutions such as shock waves. The selection of physically admissible discontinuous solutions relies up on the second law of thermodynamics, which requires that the entropy does not decrease [93,148]. Thus, to ensure the thermodynamic consistency with the second law of thermodynamics, the foregoing system of conservation laws is supplemented by the entropy inequality

$$\frac{d}{dt}\int\limits_{\omega(t)} \rho\eta\,\mathrm{d}v \geq 0. \tag{2.13}$$

We observe that the above inequality boils down to an equality for smooth solutions.

2.4. Conservation laws – local form

2.4.1. The Reynolds transport formula. Evolution equations for kinetic and internal energy

The local form of the conservation laws for mass, momentum and total energy follow from the Reynolds transport formula, [93], which can be written for any spatial field

$$\frac{d}{dt}\int_{\omega(t)} f(\mathbf{x},t)\,\mathrm{d}v = \int_{\omega(t)} \frac{df}{dt} + f\nabla\cdot\mathbf{u}\,\mathrm{d}v. \tag{2.14}$$

Here, we point out that the time derivative on the right-hand side is the material derivative of f defined by (2.6). Applying the Reynolds transport formula for $f=\rho$ and using the mass conservation equation (2.8a) leads to

$$\frac{d\rho}{dt} + \rho\nabla\cdot\mathbf{u} = 0, \tag{2.15}$$

which is the local form of the mass conservation equation, which is also called the *continuity equation*. We note that this equation can also be written

$$\rho\frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla\cdot\mathbf{u} = 0. \tag{2.16}$$

Combining the Reynolds transport formula (2.14) and the local form of the mass conservation equation (2.15) yields the useful identity

$$\frac{d}{dt}\int_{\omega(t)} \rho f\,\mathrm{d}v = \int_{\omega(t)} \rho\frac{df}{dt}\,\mathrm{d}v. \tag{2.17}$$

Applying the above identity to $(\mathbf{u}, e)$ and using the momentum (2.8b) and total energy (2.8c) conservation laws leads to their local form[2]

$$\rho\frac{d\mathbf{u}}{dt} - \nabla\cdot\mathbb{T} = \mathbf{0}, \quad \rho\frac{d\mathbf{u}}{dt} + \mathbf{grad}\,p = \mathbf{0}, \tag{2.18}$$

$$\rho\frac{de}{dt} - \nabla\cdot(\mathbb{T}\mathbf{u}) = 0, \quad \rho\frac{de}{dt} + \mathbf{div}\,p\,\mathbf{u} = 0. \tag{2.19}$$

Taking the dot product of the momentum equation with the velocity field $\mathbf{u}$ yields the kinetic energy equation

$$\rho\frac{d}{dt}\left(\frac{1}{2}\mathbf{u}^2\right) - \mathbf{u}\cdot\nabla\cdot\mathbb{T} = 0, \quad \rho\frac{d}{dt}\left(\frac{1}{2}\mathbf{u}^2\right) + \mathbf{u}\cdot\mathbf{grad}\,p. \tag{2.20}$$

Subtracting the kinetic energy equation from the total energy equation (2.19) then leads to the local form for the internal energy equation

$$\rho\frac{d\varepsilon}{dt} - \mathbb{T}\colon\nabla\mathbf{u} = 0, \quad \rho\frac{d\varepsilon}{dt} + p\,\mathbf{div}\,\mathbf{u} = 0. \tag{2.21}$$

Here, $\nabla\mathbf{u}$ is the velocity gradient tensor and $:$ is the inner product between tensors defined by $\mathbb{T}\colon\mathbb{S} = \mathrm{tr}(\mathbb{T}^t\mathbb{S})$ for any second order tensor, where tr denotes the trace operator. It is worth mentioning that the above equation results from the use of the tensorial identity

$$\nabla\cdot(\mathbb{T}^t\mathbf{u}) = \mathbf{u}\cdot\nabla\cdot\mathbb{T} + \mathbb{T}\colon\nabla\mathbf{u}, \quad \mathbf{div}p\,\mathbf{u} = \mathbf{u}\cdot\mathbf{grad}\,p + p\,\mathbf{div}\,\mathbf{u}. \tag{2.22}$$

The internal energy equation (2.21) by means of the local mass conservation (2.16) can be rewritten as

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

Comparing the above equation with the fundamental Gibbs relation (2.11) shows that for smooth flows entropy is conserved, *i.e.* $\rho\theta\frac{d\eta}{dt} = 0$. On the other hand, in the presence of shock waves, entropy is increasing consistent with the second law of thermodynamics and thus satisfies the following local entropy inequality

$$\rho\frac{d\eta}{dt} \geq 0. \tag{2.23}$$

[2] In what follows we will present formulas for general stress tensor $\mathbb{T}$ and for the case where $\mathbb{T} = -p\,\mathbb{I}_d$.

2.4.2. Conservation of total energy

If we introduce the kinetic energy and the internal energy of $\omega(t)$

$$\mathcal{K}_{\omega(t)} = \int_{\omega(t)} \frac{1}{2}\rho\,\mathbf{u}^2\,\mathrm{d}v, \quad \mathcal{E}_{\omega(t)} = \int_{\omega(t)} \rho\,\varepsilon\,\mathrm{d}v,$$

then integrating (2.20) and (2.21) over $\omega(t)$ and employing (2.17) yields the integral form of the kinetic and internal energy equations

$$\frac{d\mathcal{K}_{\omega(t)}}{dt} = \frac{d}{dt}\int_{\omega(t)} \frac{1}{2}\rho\,\mathbf{u}^2\,\mathrm{d}v = \int_{\omega(t)} \mathbf{u}\cdot\nabla\cdot\mathbb{T}\,\mathrm{d}v\,, \tag{2.24}$$

$$\frac{d\mathcal{E}_{\omega(t)}}{dt} = \frac{d}{dt}\int_{\omega(t)} \rho\,\varepsilon\,\mathrm{d}v = \int_{\omega(t)} \mathbb{T}\colon\nabla\mathbf{u}\,\mathrm{d}v\,. \tag{2.25}$$

Now we can observe that the conservation of total energy, *i.e.*, $E_{\omega(t)} = \mathcal{E}_{\omega(t)} + \mathcal{K}_{\omega(t)}$ is obtained by summing (2.25) and (2.24), then comparing it to (2.8c) and utilizing the integral identity

$$\int_{\partial\omega(t)} \mathbb{T}\mathbf{n}\cdot\mathbf{u}\,\mathrm{d}s = \int_{\omega(t)} \mathbf{u}\cdot\nabla\cdot\mathbb{T}\,\mathrm{d}v + \int_{\omega(t)} \mathbb{T}\colon\nabla\mathbf{u}\,\mathrm{d}v, \tag{2.26}$$

which is nothing but the integral counterpart of the tensorial identity (2.22). The discrete analog of this identity is of great importance for constructing conservative mimetic spatial discretizations in the context of SGH, refer to [48].

In the case of gas dynamics the identity (2.26) looks as follows

$$\int_{\omega(t)} p\,\mathbf{div}\,\mathbf{u}\,\mathrm{d}v + \int_{\omega(t)} \mathbf{u}\cdot\mathbf{grad}\,p\,\mathrm{d}v = \int_{\partial\omega(t)} p\,\mathbf{u}\cdot\mathbf{n}\,\mathrm{d}s\,, \tag{2.27}$$

and the balance equations for kinetic and internal energy are

$$\frac{d\mathcal{E}_{\omega(t)}}{dt} = -\int_{\omega(t)} p\,\mathbf{div}\,\mathbf{u}\,\mathrm{d}v\,; \quad \frac{d\mathcal{K}_{\omega(t)}}{dt} = -\int_{\omega(t)} \mathbf{u}\cdot\mathbf{grad}\,p\,\mathrm{d}v\,. \tag{2.28}$$

2.4.3. Geometric conservation law

Applying (2.17) to $f = \frac{1}{\rho}$ and using the mass conservation equation (2.8a) yields

$$\frac{d}{dt}\int_{\omega(t)} \mathrm{d}v - \int_{\partial\omega(t)} \mathbf{u}\cdot\mathbf{n}\,\mathrm{d}s = 0. \tag{2.29}$$

This is the Geometrical Conservation Law which governs the time rate of change of $\omega(t)$ volume. The geometrical conservation law is strongly linked to the trajectory equation (2.5). The quality of the spatial approximation of this equation is crucial for any numerical method for discretizing Lagrangian hydrodynamics.

2.5. Divergence and evolution of parcel volume

The standard coordinate invariant definition of divergence is

$$\mathbf{div}\,\mathbf{u} = \lim_{\omega\to 0} \frac{\int_{\partial\omega}\mathbf{u}\cdot\mathbf{n}\,ds}{|\omega|}\,, \tag{2.30}$$

which defines the divergence in ω.

If we consider the Lagrangian parcel ω with constant mass in time $m = \rho(t)\,|\omega(t)|$ then differentiating in time we get

$$0 = \frac{dm}{dt} = \frac{d\rho}{dt}\,|\omega| + \rho\,\frac{d|\omega|}{dt}\,,$$

or

$$\frac{d\rho}{dt} + \rho\,\frac{1}{|\omega|}\,\frac{d|\omega|}{dt} = 0\,.$$

Comparing this equation with the continuity equation (2.15), we conclude that

$$\mathbf{div}\,\mathbf{u} = \lim_{\omega\to 0} \frac{1}{|\omega|}\frac{d|\omega|}{dt}\,. \tag{2.31}$$

Therefore, in the continuum case we have two equivalent definitions for the divergence – (2.30) and (2.31). Both of these forms will be used in the discrete case and we would ideally like to obtain the same discrete divergence operator for both.

2.6. Time evolution of mass averaged quantities

Identity (2.17) can be rewritten as follows

$$\frac{d}{dt}\int\limits_{\omega(t)} \rho\, f\, \mathrm{d}v = \int\limits_{\omega(t)} \rho\,\frac{df}{dt}\,\mathrm{d}v = m_{\omega(t)}\left(\frac{df}{dt}\right)_{m_{\omega(t)}}, \tag{2.32}$$

where $\left(\frac{df}{dt}\right)_{m_{\omega(t)}} = \left(\int_{\omega(t)} \rho\,\frac{df}{dt}\,dv\right)/m_{\omega(t)}$ is the mass average of the df/dt over $\omega(t)$.

This means that if we take $f = \mathbf{x}$ then the time evolution of the center of mass of $\omega(t)$ – $\mathbf{x}_{\omega(t)} = \left(\int_{\omega(t)} \rho\,\mathbf{x}\,\mathrm{d}v\right)/m_{\omega(t)}$ is

$$\frac{d\mathbf{x}_{\omega(t)}}{dt} = \left(\frac{d\mathbf{x}}{dt}\right)_{m_{\omega(t)}}.$$

Applying (2.32) to the velocity and the internal energy we can obtain a useful form for the conservation of momentum equation and the balance equation for the internal energy (for gas dynamics)

$$m_{\omega(t)}\frac{d\mathbf{u}_{m_{\omega(t)}}}{dt} = -\int\limits_{\partial\omega(t)} p\,\mathbf{n}\,\mathrm{d}s\,,\quad m_{\omega(t)}\frac{d\varepsilon_{m_{\omega(t)}}}{dt} = -\int\limits_{\omega(t)} p\,\mathbf{div}\,\mathbf{u}\,\mathrm{d}v\,. \tag{2.33}$$

The first equation in (2.33) means that the total force exerted on $\omega(t)$ is equal to its mass times the acceleration of its center of mass. This is nothing more than Newton's law for classical particle mechanics.

2.7. Summary of the governing equations for Lagrangian hydrodynamics

Let us present the partial differential equations for Lagrangian hydrodynamics[3]

$$\frac{d\mathbf{x}}{dt} = \mathbf{u}\,,\quad \text{Trajectory equation} \tag{2.34}$$

$$\rho\frac{d}{dt}\left(\frac{1}{\rho}\right) - \nabla\cdot\mathbf{u} = 0\,,\quad \frac{d\rho}{dt} + \rho\,\mathbf{div}\mathbf{u} = 0\,,\quad \text{mass conser.} \tag{2.35}$$

$$\rho\frac{d\mathbf{u}}{dt} - \nabla\cdot\mathbb{T} = \mathbf{0}\,,\ \rho\frac{d\mathbf{u}}{dt} + \mathbf{grad}\,p = \mathbf{0}\,,\ \text{momentum conservation} \tag{2.36}$$

$$\rho\frac{de}{dt} - \nabla\cdot(\mathbb{T}\mathbf{u}) = 0\,,\ \rho\frac{de}{dt} + \mathbf{div}p\mathbf{u} = 0\,,\ \text{total energy conservation}\,. \tag{2.37}$$

We remark that for smooth flows the total energy equation can be replaced by the internal energy equation

$$\rho\frac{d\varepsilon}{dt} - \mathbb{T}\colon\nabla\mathbf{u} = 0\,,\quad \rho\frac{d\varepsilon}{dt} + p\,\mathbf{div}\mathbf{u} = 0\,. \tag{2.38}$$

3. Mesh and its dynamics

The two-dimensional Euclidean space is equipped with the orthonormal basis $(\mathbf{e}_x, \mathbf{e}_y)$, which is supplemented by $\mathbf{e}_z = \mathbf{e}_x \times \mathbf{e}_y$. Let $D(t)$ be the convecting computational domain at time $t \geq 0$. It is paved using a collection of non-overlapping polygonal cells denoted by $\omega_c(t)$, *i.e.* $D(t) = \cup_c \omega_c(t)$. The label c identifies specific $\omega_c(t)$.

In Sections 4 and 5 mesh cell, $\omega_c(t)$, is a general polygon with sides which are the segments of straight lines. In Section 6 we consider cells with curvilinear boundaries. In this Section we will consider only polygonal cells and all the required information about curvilinear cells will be presented in Section 6.

[3] We present equations for general stress tensor as well as for gas dynamics.

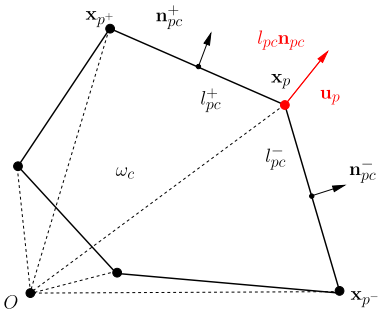

Fig. 1. Polygonal cell triangulation and notations.

3.1. Discrete fluid particle and trajectory equation

When taking the discrete Lagrangian approach the first decision to be made is how to represent a fluid parcel. Because in this paper we are only considering mesh based methods the fluid parcel will be represented by a cell of the mesh, $\omega_c(t)$.

In the case of the polygonal cells, which we are considering in this Section, the shape of the cell is completely defined by the positions of the vertices, $p \in \mathcal{P}(c)$, here $\mathcal{P}(c)$ is a set of all the vertices of the cell c.

Therefore, the notion of the fluid parcel moving with the fluid in this case means that the vertices of the cell are moving with the local fluid velocity and therefore their coordinates, $\mathbf{x}_p$ have to be found from some approximation of the trajectory equation

$$\frac{d\mathbf{x}_p}{dt} = \mathbf{u}_p \,, \mathbf{x}_p(0) = \mathbf{X}_p \,, \tag{3.1}$$

where

$$\mathbf{u}_p = \frac{d\mathbf{x}_p}{dt} = \left(\frac{dx_p}{dt} \,, \frac{dy_p}{dt}\right)^T = \left(u_p \,, v_p\right)^T$$

is the velocity of the vertex. This means that in any discrete Lagrangian hydrodynamics scheme one needs to only have a discrete representation of the velocity at the nodes, $\mathbf{u}_p$. It also means that in any discrete Lagrangian hydrodynamics scheme one needs some approximation of trajectory equation (3.1).

Let us note that even if at the initial time the shape of the parcel is a polygon with straight sides, these sides will not stay straight for later time moments. Therefore, our assumption that computational cell always has straight sides is a source of error in the definition of the shape and volume of the cell and therefore in the density. Clearly there are other sources of time and space errors too, for example, a time error is coming from approximation of trajectory equation, space and time errors are coming from momentum equation from which the velocity of the vertex is defined.

3.2. Volume of the cell and its evolution

Employing a triangulation of the polygonal cell, refer to Fig. 1, its volume is expressed in terms of the positions of the vertices as follows

$$|\omega_c(t)| = \sum_{p \in \mathcal{P}(c)} \frac{1}{2}(\mathbf{x}_p \times \mathbf{x}_{p^+}) \cdot \mathbf{e}_z. \tag{3.2}$$

In the counter-clockwise ordered list of vertices of cell c, p^+ is the next point with respect to p. The coordinate form of the volume is

$$|\omega_c(t)| = \sum_{p \in \mathcal{P}(c)} x_p \frac{y_{p^+} - y_{p^-}}{2} = -\sum_{p \in \mathcal{P}(c)} y_p \frac{x_{p^+} - x_{p^-}}{2} \,, \tag{3.3}$$

where vertex p^- precedes point p.

Remarking that the cell volume is an explicit function of the coordinates of the vertices, we apply the chain rule of the composed derivative to compute the time rate of change of the cell volume as

$$\frac{d}{dt}(|\omega_c(t)|) = \sum_{p \in \mathcal{P}(c)} \frac{\partial |\omega_c(t)|}{\partial \mathbf{x}_p} \cdot \frac{d\mathbf{x}_p}{dt} = \sum_{p \in \mathcal{P}(c)} \left(\frac{\partial |\omega_c(t)|}{\partial x_p} u_p + \frac{\partial |\omega_c(t)|}{\partial y_p} v_p \right).$$

Taking the gradient of (3.2) with respect to the vertex coordinates $\mathbf{x}_p$, we readily obtain

$$\frac{\partial|\omega_c(t)|}{\partial \mathbf{x}_p} = \frac{1}{2}[(\mathbf{x}_{p^+} - \mathbf{x}_p) \times \mathbf{e}_z + (\mathbf{x}_p - \mathbf{x}_{p^-}) \times \mathbf{e}_z] = \left(\frac{y_{p^+} - y_{p^-}}{2}, -\frac{x_{p^+} - x_{p^-}}{2} \right)^T.$$

We note that the right-hand side of the above equation is nothing more than half of the sum of the two normals for the edges impinging on vertex p, referring to Fig. 1, for which it will be useful to introduce separate notations

$$\begin{aligned} l_{pc}^- \mathbf{n}_{pc}^- &= \frac{1}{2}(\mathbf{x}_p - \mathbf{x}_{p^-}) \times \mathbf{e}_z = \left(\frac{y_p - y_{p^-}}{2}, -\frac{x_p - x_{p^-}}{2} \right)^T, \\ l_{pc}^+ \mathbf{n}_{pc}^+ &= \frac{1}{2}(\mathbf{x}_{p^+} - \mathbf{x}_p) \times \mathbf{e}_z = \left(\frac{y_{p^+} - y_p}{2}, -\frac{x_{p^+} - x_p}{2} \right)^T, \end{aligned} \tag{3.4}$$

where $\mathbf{n}_{pc}^+$ is the unit normal to the edge connecting points p and p^+, and l_{pc}^+ which is half of the length of this edge, definitions for $\mathbf{n}_{pc}^-$, l_{pc}^- are similar but for the edge connecting points p^- and p. Bearing this in mind, the volume gradient can be written as

$$\frac{\partial|\omega_c(t)|}{\partial \mathbf{x}_p} = l_{pc}^- \mathbf{n}_{pc}^- + l_{pc}^+ \mathbf{n}_{pc}^+ = l_{pc}\mathbf{n}_{pc}, \tag{3.5}$$

where $l_{pc}\mathbf{n}_{pc}$ is called the *corner* vector. In the definition of this vector the $\mathbf{n}_{pc}$ is a unit corner normal, and l_{pc} is the length of the corner vector. Let us note the corner vector depends on both the cell and the point, that is, it will be different for the same point when we consider a different cell. This geometrical object is the gradient of the volume with respect to the vertex coordinates. Its importance has already been identified in many papers, such as [145]. We note in passing that the corner vector satisfies the fundamental geometrical identity

$$\sum_{p \in \mathcal{P}(c)} l_{pc}\mathbf{n}_{pc} = \sum_{p \in \mathcal{P}(c)} \frac{\partial|\omega_c(t)|}{\partial \mathbf{x}_p} = \mathbf{0}. \tag{3.6}$$

Gathering the foregoing results and recalling the trajectory equation (3.1), the time rate of change of the cell volume is given by

$$\frac{d}{dt}(|\omega_c(t)|) = \sum_{p \in \mathcal{P}(c)} l_{pc}\mathbf{n}_{pc} \cdot \mathbf{u}_p. \tag{3.7}$$

The right-hand side of the above equation represents a consistent discretization of the volume flux

$$\sum_{p \in \mathcal{P}(c)} l_{pc}\mathbf{n}_{pc} \cdot \mathbf{u}_p \approx \int_{\partial\omega_c(t)} \mathbf{u} \cdot \mathbf{n}\, \mathrm{d}s, \tag{3.8}$$

and therefore equation (3.7) represent the discrete form of the GCL, (2.29).

Comment 1. Regardless the cell shape, the geometrical identity (3.6) means that the cell volume does not change when the velocity field, $\mathbf{u}$, is constant over the computational grid.

Comment 2. We have shown that

$$\frac{1}{|\omega_c(t)|}\frac{d}{dt}(|\omega_c(t)|) = \frac{1}{|\omega_c(t)|} \sum_{p \in \mathcal{P}(c)} l_{pc}\mathbf{n}_{pc} \cdot \mathbf{u}_p\,.$$

The LHS of this expression is an approximation of the divergence in the form of (2.31) and because of (3.8) the expression on RHS is an approximation of the divergence in the form (2.30) so both of these approximations are the same. Therefore, it is natural to define the discrete divergence operator over the polygonal cell ω_c as

$$\mathcal{DIV}_c(\mathbf{u}) = \frac{1}{|\omega_c(t)|}\frac{d}{dt}(|\omega_c(t)|) = \frac{1}{|\omega_c(t)|} \sum_{p \in \mathcal{P}(c)} l_{pc}\mathbf{n}_{pc} \cdot \mathbf{u}_p\,. \tag{3.9}$$

One can show that the above formula holds exactly for affine velocity fields. If $\mathbf{u}(\mathbf{x}) = \mathbb{A}\mathbf{x} + \mathbf{B}$, where $\mathbb{A}$ and $\mathbf{B}$ are respectively a constant second-order tensor and a constant vector, then $\mathcal{DIV}_c(\mathbf{u}) = \mathrm{tr}(\mathbb{A})$. This result is the consequence of the geometrical identities (3.6) and

$$\sum_{p \in \mathcal{P}(c)} l_{pc}\mathbf{n}_{pc} \otimes \mathbf{x}_p = |\omega_c|\mathbb{I}_d.$$

The interested reader should refer to [135] for the derivation of the above identity. Moreover, in [64] this identity is the cornerstone of the proof of the weak consistency of the GLACE scheme.

4. Discrete Staggered Grid Lagrangian Hydrodynamics – SGH

4.1. Continuity equation and discrete divergence

By definition the mass of a cell, m_c, is constant in time, and therefore average density in the cell is

$$\rho_c = m_c / |\omega_c| \,. \tag{4.1}$$

We have defined the discrete velocity at the points and discrete density (and later will define discrete internal energy and pressure) in the cells, that is, discrete primary variables are defined in different locations. This fact justifies the name *staggered* for such discretizations.

Repeating the derivation of the discrete divergence in Section 3.2, we can write the continuity equation for cell c as follows

$$\frac{d\rho_c}{dt} + \rho_c \, (\mathcal{D}IV \, \mathbf{u})_c = 0 \,, \tag{4.2}$$

where similar to (3.9) the discrete divergence, $\mathcal{D}IV$ is

$$(\mathcal{D}IV \, \mathbf{u})_c = \frac{1}{|\omega_c|} \frac{d|\omega_c|}{dt} \,. \tag{4.3}$$

4.2. Equation for the internal energy

The simplest form of equation of state is $p = p(\rho, \varepsilon)$, for example, for an ideal gas we have $p = (\gamma - 1)\,\rho\,\varepsilon$. Let us remember that discrete density is naturally defined at the cells. Therefore, to avoid any additional interpolation in defining the pressure it is natural to define the internal energy and pressure resulting from equation of state also in a cell – ε_c, p_c.

The staggered location of variables makes it natural to use (2.38) to evolve the internal energy for the SGH discretization. One can see that in this case there is no interpolation needed: ρ and ε in LHS are defined at the cell; p and discrete divergence of the velocity, (4.3), in RHS is also defined at the cell.

Taking into account the definition of the discrete divergence, (4.3), the discrete analog of Eq. (2.38) can be written as follows

$$\rho_c \frac{d\varepsilon_c}{dt} = -p_c \, (\mathcal{D}IV \, \mathbf{u})_c \,, \tag{4.4}$$

or

$$m_c \frac{d\varepsilon_c}{dt} = -p_c \, |\omega_c| \, (\mathcal{D}IV \, \mathbf{u})_c \tag{4.5}$$

Let us also note that Eq. (4.5) is the discrete analog of the internal energy equation in (2.33), because we consider a discretization for which the internal energy, pressure and divergence are all constant over the computational cell.

Taking into account the definition of the discrete divergence, (4.3), a discrete analog of Eq. (2.38) can be written as follows

$$m_c \frac{d\varepsilon_c}{dt} = -p_c \frac{d|\omega_c|}{dt} \,. \tag{4.6}$$

The Eq. (4.6) is sometimes called the *entropy* form of the equation for the internal energy because it expresses conservation of entropy of the fluid parcel for smooth flows – refer to Eq. (2.11).

4.3. Discrete gradient, momentum equation, conservation of momentum

Momentum conservation for the fluid parcel in the form of Eq. (2.33) follows from the differential momentum equation (2.36) and from the following property of the **grad**

$$\int\limits_{\omega} \mathbf{grad}\, p = \int\limits_{\partial\omega} p \, \mathbf{n} \, ds \,. \tag{4.7}$$

Let us write the discrete equation for momentum conservation in the following form

$$\rho_p \frac{d\mathbf{u}_p}{dt} = -\,(\mathcal{G}RAD \, p)_p \,,$$

where ρ_p and $\mathcal{G}RAD$ are still undefined.

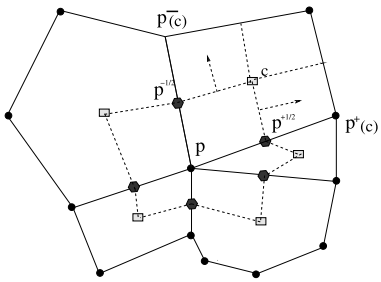

Fig. 2. Cell ω_p of dual mesh around point p.

To transform this equation into a discrete analog of momentum equation in (2.33) we first need to define some analog of the discrete fluid parcel for a point. This will be done later using the notion of a *dual* mesh. For now we just assume that computational domain can be represented as a union of dual mesh cells, ω_p, related to the points, which covers the computational domain without any gaps or overlaps. Then analog of the momentum equation in (2.33) can be written in the following form

$$m_p \frac{d\mathbf{u}_p}{dt} = -|\omega_p| \, (\mathcal{GRAD}\, p)_p \,, \tag{4.8}$$

where $m_p = \rho_p |\omega_p|$. We shall assume that nodal mass m_p does not depend on time and will return to the exact definition of ω_p and m_p later. Comparison of Eq. (4.8) with Eq. (4.7) suggests that in order to satisfy conservation of momentum in the discrete case the expression $|\omega_p| \, (\mathcal{GRAD}\, p)_p$ has to represent the discrete analog of the surface integral on the RHS of (4.7), that is,

$$|\omega_p| \, (\mathcal{GRAD}\, p)_p \approx \int\limits_{\partial\omega_p} p\,\mathbf{n}\, ds \,. \tag{4.9}$$

Because we use equation (2.38) for advancing the internal energy, the conservation of total energy in general will not be automatically satisfied if Eqs. (4.7) and (2.38) are approximated in a non-compatible way. This means that we cannot use an arbitrary discretization of $\mathcal{GRAD}$, which just satisfies (4.9).

As was discussed in Section 2, conservation of total energy follows from the identity (2.27). In the discrete case, according to the theory of mimetic finite difference discretization, [119], the discrete analog of this identity can be used to construct a discrete gradient from discrete divergence. In the case of SGH the discrete analog of (2.27) can be written as follows

$$\sum_c p_c \, (\mathcal{DIV}\, \mathbf{u})_c \, |\omega_c| + \sum_p \mathbf{u}_p \cdot (\mathcal{GRAD}\, p)_p \, \omega_p = \text{Boundary Terms} \,. \tag{4.10}$$

In Eq. (4.10) the first sum is over all the cells in the mesh, the second sum is over all points in the mesh; the boundary terms refers to the discrete analog of the surface integral of $p\,\mathbf{u}\,\mathbf{n}$ over the boundary of computational domain.

Eq. (4.10), according to the mimetic finite difference methodology is considered an identity with respect to $\mathbf{u}_p$ and allows us to define the discrete operator $\mathcal{GRAD}$ – this means that the discrete gradient is the negative adjoint of the discrete divergence.

Using the explicit definition of the discrete divergence we can obtain

$$(\mathcal{GRAD}\, p)_p = \frac{1}{|\omega_p|} \cdot \left(\sum_{c \in \mathcal{C}(p)} \frac{\partial |\omega_c|}{\partial \mathbf{x}_p} \, p_c \right) = \frac{1}{|\omega_p|} \cdot \begin{pmatrix} \sum\limits_{c \in \mathcal{C}(p)} p_c \, \dfrac{y_{p^-(c)} - y_{p^+(c)}}{2} \\ - \sum\limits_{c \in \mathcal{C}(p)} p_c \, \dfrac{x_{p^-(c)} - x_{p^+(c)}}{2} \end{pmatrix},$$

where $\mathcal{C}(p)$ is set of cells which share the point p.

The discrete momentum equation, (4.8), now can be written as follows

$$m_p \frac{d\mathbf{u}_p}{dt} = \sum_{c \in \mathcal{C}(p)} \mathbf{f}_{pc} \,, \quad \mathbf{f}_{pc} = -p_c \frac{\partial |\omega_c|}{\partial \mathbf{x}_p} = \begin{pmatrix} -p_c \, \frac{y_{p^-(c)} - y_{p^+(c)}}{2} \\ p_c \, \frac{x_{p^-(c)} - x_{p^+(c)}}{2} \end{pmatrix}. \tag{4.11}$$

Now let us prove that this discrete gradient satisfies the discrete analog of (4.9). We define ω_p using notion of the median dual mesh. ω_p is formed by connecting the mid sides of the edges with middle points of the cells as shown in Fig. 2.

Because the pressure is constant in the cell it is simple to show that

$$f_{pc} = -\int_{\partial\omega_p \cap \omega_c} p\,\mathbf{n}\,ds\,,$$

and therefore

$$\sum_{c\in\mathcal{C}(p)} f_{pc} = -\int_{\partial\omega_p} p\,\mathbf{n}\,ds\,,$$

that is the discrete momentum equation can be written in locally conservative form.

The total momentum for entire computational domain is defined as follows

$$\boldsymbol{\mu}_{D(t)} = \sum_p m_p\,\mathbf{u}_p\,, \tag{4.12}$$

and therefore

$$\frac{d\boldsymbol{\mu}_{D(t)}}{dt} = \sum_p m_p\,\frac{d\mathbf{u}_p}{dt}\,, \tag{4.13}$$

and it is clear that local conservation of momentum guarantees global conservation (if the surface forces on the boundary of computational domain are zero).

4.4. Conservation of total energy

If we define the total energy as

$$E_{D(t)} = \mathcal{E}_{D(t)} + \mathcal{K}_{D(t)}\,;\quad \mathcal{E}_{D(t)} = \sum_c m_c\,\varepsilon_c\,,\quad \mathcal{K}_{D(t)} = \sum_p m_p\,\frac{|\mathbf{u}_p|^2}{2}\,, \tag{4.14}$$

then the conservation of discrete total energy for the entire computational domain follows from the compatible definition of the discrete divergence and gradient and the forms of the discrete momentum and internal energy equations.

However, it will be educational and useful for the derivation in later sections to derive conservation of total energy in a different more general way. From the momentum equation in the form (4.11) one can derive the evolution equation for the kinetic energy by multiplying both sides of this equation by $\mathbf{u}_p$

$$m_p\,\frac{d(|\mathbf{u}_p|^2/2)}{dt} = \mathbf{u}_p \cdot \sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc}\,,$$

and, therefore, the evolution of kinetic energy of entire domain is

$$\frac{d\mathcal{K}_{D(t)}}{dt} = \sum_p \left(m_p\,\frac{d(|\mathbf{u}_p|^2/2)}{dt} \right) = \sum_p \left(\mathbf{u}_p \cdot \sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc} \right)\,. \tag{4.15}$$

Let us for simplicity assume that the normal component of the velocity is zero on the boundary, then by rearranging terms in the RHS of (4.15) we get

$$\frac{d\mathcal{K}_{D(t)}}{dt} = \sum_c \left(\sum_{p\in\mathcal{P}(c)} \mathbf{f}_{pc}\cdot\mathbf{u}_p \right)\,. \tag{4.16}$$

The internal energy equation, (4.6), can be rewritten using the definition of $\mathbf{f}_{pc}$ as follows

$$m_c\,\frac{d\varepsilon_c}{dt} = -\sum_{p\in\mathcal{P}(c)} \mathbf{u}_p\cdot\mathbf{f}_{pc}$$

and, therefore, the evolution of the internal energy of entire domain is

$$\frac{d\mathcal{E}_{D(t)}}{dt} = \sum_c \left(m_c\,\frac{d\varepsilon_c}{dt} \right) = -\sum_c \left(\sum_{p\in\mathcal{P}(c)} \mathbf{u}_p\cdot\mathbf{f}_{pc} \right) \tag{4.17}$$

Now one can recognize that RHSs of the Eqs. (4.16) and (4.17) have the same magnitude and opposite sign and therefore

$$\frac{dE_{D(t)}}{dt} = \frac{d\mathcal{E}_{D(t)}}{dt} + \frac{d\mathcal{K}_{D(t)}}{dt} = 0\,.$$

4.5. Cell, nodal and subzonal masses

To complete description of the SGH discretization we need to define cell, m_c, and nodal, m_p masses.
Clearly total mass counted by cells or points has to be the same

$$\sum_c m_c = \sum_p m_p\,. \tag{4.18}$$

In modern SGH methods on general polygonal meshes in order to define consistent cell and nodal masses one employs the notion of the Lagrangian corner mass m_{pc} from which both cell and nodal mass are defined as follows

$$m_c = \sum_{p\in\mathcal{P}(c)} m_{pc}\,, \quad m_p = \sum_{c\in\mathcal{C}(p)} m_{pc}\,,$$

this definition guarantees (4.18). The are many ways to define the corner mass, for example, if at an initial time moment we have an analytic distribution of density then we can subdivide a cell into corners or subcell in some way and then compute masses corresponding to corner subcells.

4.6. The general form of the compatible SGH discretization

The generic compatible form of the discrete momentum and internal energy equations which guarantee conservation of total energy are (see [41,48] for details)

$$m_p \frac{d\mathbf{u}_p}{dt} = \sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc}\,, \quad m_c \frac{d\varepsilon_c}{dt} = -\sum_{p\in\mathcal{P}(c)} \mathbf{f}_{pc}\cdot\mathbf{u}_p\,, \tag{4.19}$$

where $\mathbf{f}_{pc}$ denotes generic *so-called* sub-zonal forces.

In previous Sections we have shown that in the simplest case of pressure forces SGH can be presented in the form of the Eqs. (4.19). In the next two sections we will give other examples of forces, which are not physical – artificial viscosity forces and anti-hourglass forces.

Let us note that the general requirement for local conservation of momentum is that the expression $\sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc}$ approximates some surface integral.

The condition for conservation of momentum over the entire domain, which is weaker than the local conservation of momentum can be derived as follows. From Eqs. (4.13), (4.19) we get

$$\frac{d\boldsymbol{\mu}_{D(t)}}{dt} = \sum_p m_p \frac{d\mathbf{u}_p}{dt} = \sum_p \left(\sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc} \right) = \sum_c \left(\sum_{p\in\mathcal{P}(c)} \mathbf{f}_{pc} \right)\,.$$

Therefore a sufficient condition for global conservation of momentum is

$$\sum_{p\in\mathcal{P}(c)} \mathbf{f}_{pc} = 0\,.$$

4.7. Elimination of artificial grid motion by means of subcell masses and pressures

We can illustrate the basic problem of anomalous grid distortion on an example of a so called *hourglass* grid motion. Consider the six patterns of motion illustrated in Fig. 3 that are depicted in terms of a single quadrilateral zone where the arrows indicate the velocity vectors at the grid points. It is possible to repeat any of these six patterns globally across the entire grid. They define the six global physical motions of the grid in two dimensions: two each of translation, extension (contraction), and shear.[4] A single quadrilateral has eight degrees of freedom corresponding to the two independent directions of each of its four defining points. However, only the six global patterns, three of them shown in Fig. 3 are physical; an additional two patterns are the non-physical hourglass motions in each independent direction that are also global across the grid.

A pure hourglass motion in two dimensions consists of a checkerboard pattern with a velocity field in a given direction that alternates in sign at every grid point, as shown in Fig. 4. There is one such pattern in each direction for a two-dimensional quadrilateral grid.

[4] Note that a pure rotation can be obtained by subtracting two patterns of shear in orthogonal directions.

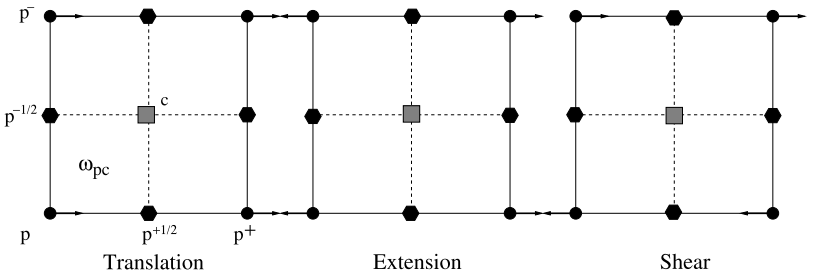

Fig. 3. Physical modes for cell ω_c. Subcell ω_{pc} formed by connecting point p with midside point $p^{+1/2}$, geometric cell center of cell – c, and midside point $p^{-1/2}$ – see left bottom panel.

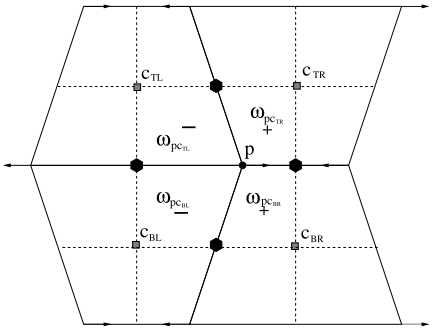

Fig. 4. Reaction of subcell pressures on hourglass point movement. Subcell pressure in subcells $\omega_{pc_{BR}}$ and $\omega_{pc_{TR}}$ is growing, and subcell pressure in subcells $\omega_{pc_{BL}}$ and $\omega_{pc_{TL}}$ is decreasing.

Let us next consider how the dynamical equations respond to the eight global patterns of motion on a staggered spatial grid composed of quadrilateral zones with one density and one pressure per cell.

Let us consider the situation when the initial pressure is constant throughout the domain and the velocity field corresponds to a pure hourglass pattern. The pure hourglass motion does not change volume of the cells and therefore according to Eqs. (4.1), (4.6), density and internal energy in the cell does not change. Therefore pressure, which is computed from equation of state also does not change. The acceleration of the points is computed from (4.11), and according to (3.6) for constant pressure the acceleration is zero, therefore the hourglass velocity field is not changing and eventually the mesh will tangle. Clearly, in real calculation hourglass patterns develop in the background of the main fluid motion and start from perturbations in the fields due to finite accuracy of computer calculations and discretization errors.

The SGH discretization with one density and one pressure per cell does not respond to hourglass motion, because the density and internal energy in the cell, and therefore the pressure, do not change under such motion.

Let us note that of the eight global motions that only the ones corresponding to extension (contraction) elicit a direct response from the dynamical equations independently of the boundary conditions; the others leave the volume of the quadrilateral zone unchanged. However, both the shear and translation patterns are physical and cause no difficulties. The difficulty is with the non-physical hourglass patterns. They are generally found to grow with time at a rate that is not easy to predict.

A possible solution to this problem that has long been recognized is to allow for subzonal volumes, Lagrangian masses, and thus, subzonal densities and pressures that produce subzonal forces, since the root of the difficulty is that the zone volume does not change in response to hourglass motion. Focus on point p as shown in Fig. 4. If we consider the four subcells, $\omega_{pc_{TR}}$, $\omega_{pc_{TL}}$, $\omega_{pc_{BL}}$, $\omega_{pc_{BR}}$ that all contain point p as a vertex, then these subcells do change as it moves, as can be seen from this figure. If we consider the masses inside these subzonal volumes to be constant, that is consider subzones as Lagrangian objects, in addition to the total mass of the zone, then from these changing subvolumes the separate densities, and thus pressures, can be computed. The direction of the change in the pressure in these subcells are indicated by plus and minus signs on the Fig. 4. These pressures produce forces that obviously oppose motion of point p and also that of the hourglass pattern everywhere across the grid. At the same time it is important to note that they provide no resistance to the physical patterns of either translation or shear. For extension (contraction) both the zone and subzone volumes change. In this case the values of the subcell pressures correspond to refining the grid by a factor of two in each direction. Thus, these subzonal pressures do not oppose this physical motions.

Let us describe how the subcell pressures resulting from subcell masses and subcell volumes can be included in SGH discretization methodology. First of all, as we have mentioned before we assume that at an initial time moment we can introduce a dual median mesh such that in each cell we have set of quadrilateral subcells, ω_{pc}, whose vertices are vertices of the cell, midpoints of the edges and the geometric cell center – top left panel in Fig. 3. Then taking into account initial density distribution we assign some mass m_{pc} to the subcells and will consider these constant in time. The shape of the

subcell at each time moment is defined from current coordinates of the vertices of the cell in the same way as it was done at the initial time. Therefore, volume of subcell has changed and we can compute subcell density as $\rho_{pc} = m_{pc}/|\omega_{pc}|$. In the discretization we have describing we still have only one internal energy per cell. Therefore, subcell pressure is defined as $p_{pc} = p(\rho_{pc}, \varepsilon_c)$. There are simplified approach to compute p_{pc}, which does not involve an EOS call. It is based on adiabatic assumption, which leads to the following approximate expression $p_{pc} = p_c + a_c^2 (\rho_{pc} - \rho_c)$.

Now that we have pressures we need to describe how to produce subcell forces, $\mathbf{f}_{pc}$. There is no unique way of doing this. Let us describe two possible approaches. First approach based on discretization of momentum equation and direct discretization of the surface integral $\int_{\partial\omega_p} p\,\mathbf{n}\,ds$. Referring to Fig. 2 we define subcell force as follows

$$\mathbf{f}_{pc} = -\left(\frac{p_{pc} + p_{p^+c}}{2} \cdot \mathbf{N}_{p^{+1/2}c} + \frac{p_{pc} + p_{p^-c}}{2} \cdot \mathbf{N}_{p^{-1/2}c} \right)$$

which corresponds to part of the surface $\partial\omega_p$ going through the cell c, and vectors $\mathbf{N}$ is the product of the length and the unit normal to the corresponding edge of the dual cell. That is the pressure on the piece of the surface between two subcells is defined as average of the subcell pressures. Let us note that for such an approach only the subcell pressures in the subcells, which share interfaces with the subcell ω_{pc} are involved in the definition of the force $\mathbf{f}_{pc}$. By construction such discretization conserves momentum. After the force $\mathbf{f}_{pc}$ is defined according to Eqs. (4.19) our discretization also conserves total energy. There is another possible approach based on mimetic discretization theory. In this approach one starts with the definition of the discrete divergence $(\mathcal{D}IV\,\mathbf{u})_{pc}$ which is based on the connection of the divergence with volume change

$$(\mathcal{D}IV\,\mathbf{u})_{pc} = \frac{1}{|\omega_{pc}|} \frac{d|\omega_{pc}|}{dt}\,. \tag{4.20}$$

Here differentiation with respect to time has to be performed taking into account that only the coordinates of one vertex of the subcell is an independent variable and all other vertices are obtained by averages. The details of the derivation is presented in Section 6.5 of [119]. The resulting expression gives the divergence in the subcell as a function of the velocities of the parent cell. Next one derives a compatible discretization for the gradient using a discrete analog of identity (2.27). In this case the resulting subcell force $\mathbf{f}_{pc}$ involves subcell pressures from all subcells of the parent cell. Let us note that a more general approach based on distribution of forces from so-called *non-dynamical* points, whose movement is slaved to *dynamical* points is described in [50].

4.8. Artificial viscosity

In SGH discretization described in previous sections we use an internal energy evolution equation in the form (4.6), which expresses conservation of entropy for a Lagrangian parcel for smooth flows. For non-smooth flows, which contain shocks entropy has to increase. Therefore we need to introduce a mechanism which will allow the transformation of kinetic energy into internal energy and increase the entropy. This is done by introducing an artificial viscosity term into momentum equation and corresponding work term into the internal energy equation such that internal energy is increased and total energy is conserved.

Let us enumerate the desired properties that a proper artificial viscosity should possess, [165,51]:

- Artificial viscosity must always act to decrease kinetic energy, that is, it must be dissipative (Dissipativity).
- Artificial viscosity should vanish uniformly (smoothly) as the velocity field becomes constant (Galilean invariance).
- Artificial viscosity should vanish for a uniform contraction and/or a rigid rotation (Self-similar motion invariance).
- Artificial viscosity should have no effect along a wave front of constant phase. This is because the velocity component tangential to a shock front is continuous in the limit of arbitrary grid refinement in this direction (Wave front invariance).
- The artificial viscous force should go to zero continuously as compression vanishes and expansion develops, and remain zero for the latter. Compression and expansion must be defined in some relevant context (Viscous force continuity).
- The asymptotic independence of mesh (Mesh invariance).

There are many approaches for designing artificial viscosity. Here we will present a short description of the so-called *tensor* artificial viscosity [46,120]. The main design principle for tensor viscosity is to construct it on the basis of the discretization of the gradient of the velocity. This is important as it allows us to reduce mesh imprinting. Therefore, the viscosity tensor will be based up on a discrete analog of $\kappa\,\mathbf{grad}\,\mathbf{u}$, where κ is some coefficient; in this paper $\kappa > 0$ is a positive scalar, in general it can be a positive definite tensor. If we only consider the viscosity terms then in the continuum case the momentum and the internal energy equations can be written as follows

$$\rho \frac{d\mathbf{u}}{dt} = \mathbf{div}\,(\kappa\,\mathbf{grad}\,\mathbf{u})\,, \quad \rho \frac{d\varepsilon}{dt} = \kappa\,\mathbf{grad}\,\mathbf{u} : \mathbf{grad}\,\mathbf{u}\,. \tag{4.21}$$

It is clear that in the differential case this form of the viscosity is dissipative, that is the term on the RHS of second equation in (4.21) is non-negative.

To discretize the operators of the gradient of the vector and the divergence of the tensor we will use a mimetic finite difference method, [119,46]. The discrete gradient is described by its tangential components with respect to the edges of cell

$$((\mathcal{GRAD}\,\mathbf{u})\cdot\mathbf{t_e})_e = \left(\mathbf{u}_{e(e)} - \mathbf{u}_{b(e)}\right)\big/\,|e|\,, \tag{4.22}$$

where e is the edge of the cell, which in 2D coincides with the face, but in 3D will be an edge; $\mathbf{t}_e$ is the unit vector tangential to the edge, and $e(e)$, $b(e)$ are the end points of the edge. The Eq. (4.22) is just an extension of the formula for the directional derivatives to tensors. Therefore this operator acts from the space of vectors defined by its Cartesian components at the vertices to the space of tangential components of a tensor defined on its edges. The derivation of the discrete divergence which acts on the tensor is based on the following identity

$$\int_D (\mathcal{GRAD}\,\mathbf{u}) : \mathbb{T}\,dv + \int_D \mathbf{u}\cdot\mathbf{div}\,\mathbb{T}\,dv = \int_{\partial D} \mathbf{u}\cdot(\mathbb{T}\cdot\mathbf{n})\,ds\,, \tag{4.23}$$

which means that $\mathcal{DIV} = -\mathcal{GRAD}^*$. The resulting discrete divergence acts from the space of tensors given by its tangential components on the edges to the space of vectors defined at the vertices. The discrete equations are

$$m_p\,\frac{d\mathbf{u}_p}{dt} = v_p\,\left(\mathcal{DIV}\,(\kappa\,\mathcal{GRAD}\,\mathbf{u})\right)_p\,, \quad m_c\,\frac{d\varepsilon_c}{dt} = (\kappa\,\mathcal{GRAD}\,\mathbf{u}\colon\mathcal{GRAD}\,\mathbf{u})_c\,, \tag{4.24}$$

and therefore the tensor viscosity is dissipative in the discrete case. In [46] it is shown that the discrete divergence of the gradient can be represented in terms of subcell forces and the contribution of the tensor artificial viscosity into momentum and internal energy equations can be represented in a compatible form – (4.19). Details of the derivations can be found in [119,46].

The general form of the scalar coefficient κ_c for the ideal gas is given by

$$\kappa_c = (1-\psi_c)\,\rho_c\left\{c_2\,\frac{(\gamma+1}{4}|\Delta\mathbf{u}_c| + \sqrt{c_2^2\left(\frac{\gamma+1}{4}\right)^2|\Delta\mathbf{u}_c|^2 + c_1^2 a_c^2}\right\}\,,$$

where $\Delta\mathbf{u}_c$ is some measure of the change of the velocity over the cell, c_1 and c_2 are dimensionless coefficients, a_c is adiabatic speed of sound; ψ_c is a limiter which is equal to unity for adiabatic compression and therefore turns the viscosity off when this occurs. Finally, if the cell is under expansion, then artificial viscosity is also turned off. The detailed analysis of the tensor artificial viscosity with respect to the list of the desired properties and in particular different options for the construction of κ_c are described in [46]. The form for the scalar coefficient κ_c was originally introduced by Wilkins, [178] based on the work of Kurapatenko.

4.9. Fully discrete equation – time integration

In this Section, we describe the complete generic predictor–corrector time integration method and the sequence of computations for the compatible spatial discretization. We will use superscript n for quantities at time level t^n and $n+1$ for $t^{n+1} = t^n + \Delta t^n$. The time step, Δt^n can change with time throughout a simulation, we will briefly describe how it is chosen at the end of this section.

4.9.1. Predictor–corrector

For time integration we use predictor–corrector method. This method can be considered as one iteration for solving implicit discretization of governing equations [130]. This method does not require solving any global system of equations, it is conditionally stable and both predictor and corrector steps are conservative.

Predictor stage

The momentum equation is discretized as follows:

$$m_p\,\frac{\mathbf{u}_p^{n+1,pr} - \mathbf{u}_p^n}{\Delta t} = \sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc}^n\,, \tag{4.25}$$

where superscript $n+1, pr$ identifies the *predicted* value of the velocity at a new time level. Note that the forces are taken from the previous time step and therefore Eq. (4.25) is explicit.

The new predicted velocity, $\mathbf{u}_p^{n+1,pr}$, allows the computation of new predicted coordinates, volumes and densities

$$\mathbf{x}_p^{n+1,pr} = \mathbf{x}_p^n + \mathbf{u}_p^{n+1/2,pr}\,\Delta t\,,\ \text{where}\ \mathbf{u}_p^{n+1/2,pr} = \frac{1}{2}\left(\mathbf{u}_p^{n+1,pr} + \mathbf{u}_p^n\right)$$
$$|\omega_c^{n+1,pr}| = \mathcal{V}(\mathbf{x}_p^{n+1,pr}\,;\,p\in\mathcal{P}(c))\,,\quad \rho_c^{n+1,pr} = m_c/|\omega_c^{n+1,pr}|\,,$$

where $\mathcal{V}$ is the expression for volume in terms of the coordinates.

Next the internal energy equation is computed as

$$m_c \frac{\varepsilon_c^{n+1,pr} - \varepsilon_c^n}{\Delta t} = - \sum_{p \in \mathcal{P}(c)} \left(\mathbf{f}_{pc}^n \cdot \mathbf{u}_p^{n+1/2,pr} \right) .$$

As we will show later in this section for conservation of total energy it is important that in this equation we also use the same force (that is force from n-th time level) as in momentum the equation.

Finally, the material pressures are updated as

$$p_c^{n+1,pr} = EOS(\rho_c^{n+1,pr}, \varepsilon_c^{n+1,pr}) .$$

Let us note one more time that all computations are explicit, that is we are not required to solve any systems of equations.

Corrector stage

The corrector stage is similar to the predictor stage. The main difference is that for the computation of the subcell forces, information obtained from the predictor stage is used. If we denote average forces computed from p_c^n and $p_c^{n+1,pr}$ as $\mathbf{f}_{pc}^{n+1/2}$ then the formulas for the corrector stage can be written as follows:

$$m_p \frac{\mathbf{u}_p^{n+1} - \mathbf{u}_p^n}{\Delta t} = \sum_{c \in \mathcal{C}(p)} \mathbf{f}_{pc}^{n+1/2} ,$$

$$\mathbf{x}_p^{n+1} = \mathbf{x}_p^n + \mathbf{u}_p^{n+1/2} \Delta t , \quad \mathbf{u}_p^{n+1/2} = \frac{1}{2} \left(\mathbf{u}_p^{n+1} + \mathbf{u}_p^n \right) ,$$

$$|\omega_c^{n+1}| = \mathcal{V}(\mathbf{x}_p^{n+1} ; p \in \mathcal{P}(c)) , \quad \rho_c^{n+1} = m_c / |\omega_c^{n+1}| ,$$

$$m_c \frac{\varepsilon_c^{n+1} - \varepsilon_c^n}{\Delta t} = - \sum_{p \in \mathcal{P}(c)} \left(\mathbf{f}_{pc}^{n+1/2} \cdot \mathbf{u}_p^{n+1/2} \right) ,$$

$$p_c^{n+1} = EOS(\rho_c^{n+1}, \varepsilon_c^{n+1}) .$$

4.9.2. Conservation and geometric conservation law in fully discrete case

Conservation

In the fully discrete case the definition of momentum, kinetic, internal and total energy for the entire computational domain at time t^n are given by Eqs. (4.12), (4.14)

$$\boldsymbol{\mu}^n = \sum_p m_p \mathbf{u}_p^n ; \quad E^n = \mathcal{E}^n + \mathcal{K}^n , \ \mathcal{E}^n = \sum_c m_c \varepsilon_c^n , \ \mathcal{K}^n = \sum_p m_p \frac{|\mathbf{u}_p^n|^2}{2} .$$

Let us note that the momentum and the total energy is conserved both at the predictor and the corrector stages. The conservation of momentum follows directly from form of the discrete momentum equation. Now let us consider how the conservation of total energy follows from the discrete momentum and internal energy equations. Consider the predictor stage. If we multiply momentum equation by $\mathbf{u}_p^{n+1/2,pr} = 0.5 (\mathbf{u}_p^n + \mathbf{u}_p^{n+1,pr})$ we get

$$m_p \frac{\frac{|\mathbf{u}_p^{n+1,pr}|^2}{2} - \frac{|\mathbf{u}_p^n|^2}{2}}{\Delta t} = \sum_{c \in \mathcal{C}(p)} \left(\mathbf{f}_{pc}^n \cdot \mathbf{u}_p^{n+1/2,pr} \right) .$$

Now if we use the discrete analog of the internal energy equation and follow the derivation in Section 4.4 we arrive at the conservation of total energy in discrete form. Let us note again that to have the cancellation of terms we have to use the forces from the same time moment in the momentum and the internal energy equations.

4.10. Time step

To complete description of time discretization we need to describe how the time step is chosen. It has to be done before we start the predictor–corrector procedure.

For single material Lagrangian hydrodynamics there are two main constraints on the time step. The first constraint is related to the Courant stability condition, usually written as

$$\Delta t \leq \Delta t_{courant} = \min_c \frac{L_c}{\tilde{a}_c} ,$$

where L_c is some characteristic length associated with cell c and $\tilde{a}_c$ is some effective zonal speed of sound, which also includes information about the artificial viscosity – see, for example, Section 4 in [48].

The second constraint relates to how quickly the volume of the zone may change. Let us explain the origin of this constraint. The internal energy equation for cell c at the predictor stage can be written approximately as

$$m_c \left(\varepsilon_c^{n+1} - \varepsilon_c^n\right) = -p_c^n \,\Delta|\omega_c^{n+1}| + \Delta t\, Q_c^n \quad \Delta|\omega_c^{n+1}| = |\omega_c^{n+1}| - |\omega_c^n|\,,$$

where the heating, Q_c^n, from artificial viscosity is positive by construction. Therefore, ε_c^{n+1} is defined from this equation to be positive, so it is sufficient to choose Δt such that

$$m_c\, \varepsilon_c^n - p_c^n\, \Delta|\omega_c^{n+1}| > 0\,. \tag{4.26}$$

Clearly, it is also assumed that the internal energy on the previous time step is positive, $\varepsilon_c^n > 0$.

If $\Delta|\omega_c^{n+1}| \le 0$, that is, the zone is under compression, then there is no additional requirement on Δt to satisfy (4.26) because the expression represents the summation of two positive numbers.

If $\Delta|\omega_c^{n+1}| > 0$, then the inequality (4.26) means that the time step must be chosen such that

$$\Delta|\omega_c^{n+1}| < m_c\, \varepsilon_c^n \big/ p_c^n\,. \tag{4.27}$$

Because $m_c = \rho_c^n\, |\omega_c^n|$ and[5] $p_c^n = (\gamma - 1)\, \rho_c^n\, \varepsilon_c^n$, inequality (4.27) leads to the following inequality

$$\Delta|\omega_c^{n+1}| \Big/ |\omega_c^n| < 1/(\gamma - 1)\,, \tag{4.28}$$

which represents a constraint on Δt. Using $\mathbf{DIVu} \approx \frac{1}{|\omega|}\,\frac{d|\omega_c|}{dt}$, the constraint can be rewritten as

$$\Delta t < 1/\left((\gamma - 1)\,\mathbf{DIVu}^n\right)\,. \tag{4.29}$$

The chosen value of Δt with respect to this constraint is then determined by the minimal value of the expression over all cells which are under expansion.

The final Δt is the one which satisfies all constraints. An additional constraint may also be applied which does not allow the time step to grow more than 10%–20%.

4.11. Closure models for multimaterial cells

Multimaterial cells are used in ALE codes to represent material interfaces that undergo high deformation and cannot be modeled robustly using meshes with pure cells, that is when mesh faces are aligned with material interfaces. A separate set of material properties is normally maintained for all the materials in each multimaterial cell along with volume fractions that define the proportion of the cells volume occupied by each material. A closure model is then required to close the governing equations, which are otherwise under-determined, that is, to define how the volume fractions and states of the individual material components evolve during the Lagrangian step. The challenge in developing closure models is how to accurately update the thermodynamic states of the individual material components in the multimaterial cell, and determine the nodal forces that such a zone generates – despite the lack of information about the velocity distribution within multimaterial cells. The closure model must also not break conservation.

In what follows the index i will be used to identify specific materials within a multimaterial cell. The set of materials in zone c is denoted by $M(c)$. The volume of material i in zone c at time t^n is denoted by $V_{c,i}^n$. Each material has its own mass, $m_{c,i}$; density $\rho_{c,i}^n$; pressure, $p_{c,i}^n$; and internal energy, $\varepsilon_{c,i}^n$. It is also useful to introduce the volume fraction for each material, $0 < \alpha_{c,i}^n < 1$.

4.11.1. Equal compressibility closure model

The simplest closure model available is the equal compressibility model, or constant volume fraction model. The equal compressibility model assumes that the volume fraction does not change in time during the Lagrangian step, *i.e.* $\alpha_{c,i}^{n+1} = \alpha_{c,i}^n$, which is clearly non-physical.

To discretize the momentum equation, a single pressure per zone is required – even in the case of multimaterial zones. This pressure is referred to as the *average* pressure. The computation of this pressure is performed by the closure model. The new nodal velocities enable the new cell volumes V_c^{n+1} at t^{n+1} to be defined. Therefore, the total change in the volume of any zone, $\Delta V_c^{n+1} = V_c^{n+1} - V_c^n$ is also known.

If a zone contains multiple materials, the equal compressibility closure model distributes ΔV_c^{n+1} between materials as follows

$$\Delta V_{c,i}^{n+1} = \alpha_{c,i}^n\, \Delta V_c^{n+1}\,, \tag{4.30}$$

[5] Here, the simplest case of ideal gas is considered for simplicity.

The update of internal energies for each material is done using individual $p\,dV$ equations,

$$m_{c,i}\,(\varepsilon_{c,i}^{n+1} - \varepsilon_{c,i}^{n}) = -p_{c,i}^{n}\,\Delta V_{c,i}^{n+1} = -p_{c,i}^{n}\,\alpha_{c,i}^{n}\,\Delta V_{c}^{n+1}\,. \tag{4.31}$$

Summing equations (4.31) in zone c over all materials, the total change of internal energy in the multimaterial cell is obtained as

$$\begin{aligned} m_c\,(\varepsilon_c^{n+1} - \varepsilon_c^{n}) &= \sum_{i\in M(c)} \left[m_{c,i}\,(\varepsilon_{c,i}^{n+1} - \varepsilon_{c,i}^{n}) \right] = \\ -\sum_{i\in M(c)} \left[p_{c,i}^{n}\,\Delta V_{c,i}^{n+1} \right] &= -\sum_{i\in M(c)} \left[p_{c,i}^{n}\,\alpha_{c,i}^{n}\,\Delta V_{c}^{n+1} \right] \\ &= -\left\{ \sum_{i\in M(c)} \left[p_{c,i}^{n}\,\alpha_{c,i}^{n} \right] \right\} \Delta V_{c}^{n+1}\,. \end{aligned} \tag{4.32}$$

If the *average* zonal pressure $\tilde{p}_c^n$ is defined as

$$\tilde{p}_c^n = \sum_{i\in M(c)} \left[\alpha_{c,i}^{n}\,p_{c,i}^{n} \right], \tag{4.33}$$

then equation (4.32) may be written in a form resembling the internal energy equation for the single material case if p_c^n is replaced by $\tilde{p}_c^n$

$$m_c\,(\varepsilon_c^{n+1} - \varepsilon_c^{n}) = -\tilde{p}_c^{n}\,\Delta V_c^{n+1}\,. \tag{4.34}$$

Equation (4.34) is compatible with the momentum equation and guarantees overall conservation.

4.11.2. Pressure relaxation versus sub-cell dynamics

There are two main types of sub-scale closure model; pressure relaxation (PR) and sub-cell dynamics. PR models explicitly enforce pressure relaxation towards equilibrium every time step for all materials and do not require any information about the interface geometry. One of the PR closure models, which is widely used in production multimaterial arbitrary Lagrangian–Eulerian codes, is the so-called Tipton's model, [13]. The goal of Tipton's closure model is to find volume changes for each material such that they sum to the total volume change of the multimaterial cell (which then allows material densities to be updated) and to find a common pressure which will be used in the momentum and material internal energy equations. This has recently been extended into a closure model for high order finite element Lagrangian hydrodynamics – Section 6.12 of current paper. Sub-cell dynamics closure models such as the interface aware sub-scale dynamics closure model [14] attempt to emulate the behavior of separate Lagrangian sub-cells and require the interface geometry to be known.

4.11.3. Interface aware sub-scale dynamics (IA-SSD) closure model

The IA-SSD closure model consists of two stages. During the first, bulk stage, the well known equal compressibility model described in section 4.11.1 is used. During the second stage, sub-scale interactions of the materials inside the multimaterial cell are taken into account. Each material interacts in a pair-wise fashion with the materials with which it has a common interface inside multimaterial cell. The interactions are based on the solution of the acoustic Riemann problem between each pair of materials and are limited using physically justifiable constraints: positivity of volume, positivity of internal energy and controlled rate of pressure relaxation.

In what follows and where it does not lead to misunderstanding, the zonal index c is dropped because only a single zone is considered.

It is assumed that the total change of volume for the multimaterial zone, ΔV, is given. It is also assumed that the topology of the materials inside the multimaterial zone is known from the interface reconstruction algorithm. Each material is represented by a pure sub-polygon, and for each material i the set of the materials, $k \in M(i)$, which have a common edge with material i is known. The area of the common edge is denoted by $S_{i,k}$, and the unit normal to the common edge pointing from material i to material k is denoted by $\mathbf{n}_{i,k}$.

Volume change model

The volume change of material i described in section 4.11.1 is now denoted $\Delta V_i^{bulk} = \alpha_i\,\Delta V$, with the corresponding value of the material volume denoted as $V_i^{bulk} = \alpha_i\,V^{n+1}$. This change of volume is referred to as *bulk* because it is the result of the distribution of the change of volume of the entire multimaterial cell, which does not take into account sub-scale dynamics.

To take into account the difference in pressure of the materials inside the multimaterial zone, the volume exchange between neighboring materials in flux form is introduced as

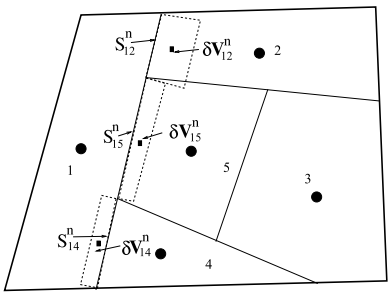

Fig. 5. Multimaterial cell: definition of the relative volume exchange between materials.

$$\Delta V_i = \Delta V_i^{bulk} + \sum_{k \in M(i)} \delta V_{i,k}\,, \quad \delta V_{i,k} = -\delta V_{k,i}\,, \tag{4.35}$$

where the explicit form of $\delta V_{i,k}$ will be described later. This form conserves the total change of the zonal volume $\sum_{i \in M(z)} \Delta V_i = \Delta V$ because $\sum_{i \in M(z)} \Delta V_i^{bulk} = \Delta V$ and because the sub-scale exchange term is in "flux" form.

The volume exchange term is constructed in the form

$$\delta V_{i,k} = \Psi_{i,k}\, \delta V_{i,k}^{max}\,, \tag{4.36}$$

where $\delta V_{i,k}^{max}$ is defined from some physical considerations taking into account material interaction, and $0 \le \Psi_{i,k} \le 1$ are limiters which are chosen in such a way that they are as close as possible to 1 and the overall model does not violate physically justified constraints – for example, the positivity of material volumes.

The value of $\delta V_{i,k}^{max}$ is estimated using the acoustic Riemann solver

$$\delta V_{i,k}^{max} = \frac{p_i - p_k}{\rho_i\, a_i + \rho_k\, a_k}\, S_{i,k}\, \Delta t\,, \tag{4.37}$$

where c_i is the adiabatic speed of sound of material i, and $S_{i,k}$ is the area of interface between materials i and k – Fig. 5. In Fig. 5 a multimaterial cell with five materials is presented with a graphical representation of $\delta V_{i,k}^{max}$ (the same notation is used for its volume and the corresponding rectangle). For material #1, $(i = 1)$; there are three neighbors: $\{2; 4; 5\}$. In the situation presented in this figure p_1 is larger than p_2 and p_5, and smaller than p_4.

Bounds for material volume

It is necessary that the volume of each material is larger than zero and less than the zonal volume

$$V^{n+1} > V_i^{n+1} > 0\,. \tag{4.38}$$

Clearly, volume may not be reduced exactly to zero in the Lagrangian step.

Therefore, the following constraints are imposed:

$$V_i^{n+1} \ge \kappa_{bot}\, V_i^{n+1,bulk}\,, 1 \ge \kappa_{bot} > 0\,. \tag{4.39}$$

Here, the parameter κ_{bot} controls how close to zero the volume of the material may reach. The constraint on volume is intended to ensure the volume stays positive, finite and the density of the material does not exceed the maximum density that the material can be compressed to.

The constraint (4.39) represents a system of linear inequalities with respect to the limiters $\Psi_{i,k}$

$$V_i^{n+1,bulk} + \sum_{k \in M(i)} \Psi_{i,k}\, \delta V_{i,k}^{max} \ge \kappa_{bot}\, V_i^{n+1,bulk}\,. \tag{4.40}$$

Positivity of internal energy

In the IA-SSD closure model, each material has a separate approximate $p\,dV$ equation. The most natural way to write the $p\,dV$ equation for material i is

$$m_i \left(\varepsilon_i^{n+1} - \varepsilon_i^n\right) \approx -p_i^n\, \Delta V_i^{n+1}\,. \tag{4.41}$$

Taking into account the above model for the change of the material volume, equation (4.41) can be rewritten as

$$m_i \left(\varepsilon_i^{n+1} - \varepsilon_i^n\right) \approx -p_i^n\, \alpha_i^n\, \Delta V^{n+1} - p_i^n \sum_{k \in M(i)} \Psi_{i,k}\, \delta V_{i,k}^{max}\,.$$

The first term in the RHS of this equation, $p_i^n\, \alpha_i^n\, \Delta V^{n+1,pr}$, corresponds to the equal compressibility model as described in section 4.11.1. The second term on the RHS, $p_i^n \sum_i \Psi_{i,k}\, \delta V_{i,k}^{max}$, is not in flux form, and therefore will lead to a violation of

the conservation of total energy. We replace p_i^n by the interfacial pressure $p_{i,k}^*$ and it is moved under the summation to form a flux exchange of internal energy between materials i and k.

$$m_i \left(\varepsilon_i^{n+1} - \varepsilon_i^n\right) = -p_i^n \alpha_i^n \Delta V^{n+1} - \sum_{k \in M(i)} p_{i,k}^* \Psi_{i,k} \delta V_{i,k}^{max}, \tag{4.42}$$

where the pressure $p_{i,k}^*$ is obtained from the one-dimensional acoustic Riemann problem between materials i and k:

$$p_{i,k}^* = \frac{(\rho_k a_k)\, p_i + (\rho_i a_i)\, p_k - (\rho_k a_k)\,(\rho_i a_i)\,(\mathbf{u}_k - \mathbf{u}_i)\cdot \mathbf{n}_{i,k}}{\rho_k a_k + \rho_i a_i}. \tag{4.43}$$

It is assumed that Δt is chosen in such a way that if all $\Psi_{i,k}$ are zero, then the internal energies of all materials at the new time step are positive, *i.e.* the time step guarantees positivity of internal energy for each material obtained by the equal compressibility model.

Then,

$$m_i\, \varepsilon_i^{n+1} = \Delta \mathcal{E}_i^{bulk} - \sum_{k \in M(i)} \Psi_{i,k}\, p_{i,k}^*\, \delta V_{i,k}^{max},$$

and the inequality required for the positivity of internal energy on the new time step can be written as

$$\Delta \mathcal{E}_i^{bulk} \geq \sum_{k \in M(i)} \Psi_{i,k}\, p_{i,k}^*\, \delta V_{i,k}^{max}.$$

Controlling pressure equilibration

Another important property of closure models is the manner in which pressure equilibration is achieved in time. Pressure equilibration should be achieved without oscillations, and in such a way that the pressures of different materials approach equilibrium in a smooth fashion.

The following approximation for the material pressure evolution is used

$$\tilde{p}_i^{n+1}(\Psi_{i,k}) = \tilde{p}_i^{bulk,n+1} - \frac{\rho_i^n (a_i^n)^2}{V_i^n} \sum_{k \in M(i)} \Psi_{i,k}\, \delta V_{i,k}^{max} \tag{4.44}$$

where

$$\tilde{p}_i^{bulk,n+1} = p_i^n - \frac{\rho_i^n (a_i^n)^2}{V_i^n} \Delta V_i^{bulk,n+1}$$

and the second term on the RHS of (4.44) is again a linear function of $\Psi_{i,k}$.

To achieve smooth equilibration, an estimate of the target equilibrated pressure, $\bar{p}$, towards which the material pressures are intended to relax toward at this time moment is computed. The simplest choice is to set $\bar{p} = \sum_i \alpha_i^n\, \tilde{p}_i^{bulk,n+1}$. Material pressures $\tilde{p}_i^{bulk,n+1}$ may be higher or lower than $\bar{p}$, however, $\max_i \tilde{p}_i^{bulk,n+1} \geq \bar{p} \geq \min_i \tilde{p}_i^{bulk,n+1}$. The limiters are chosen in such a way that $\tilde{p}_i^{n+1}$ defined by equation (4.44) will relax toward (if possible, or at least not diverge from) $\bar{p}$.

Consider the case when $\tilde{p}_i^{bulk,n+1} \geq \bar{p}$. In this case, it is required that

$$\kappa_i\, \bar{p} + (1 - \kappa_i)\, \tilde{p}_i^{bulk,n+1} \leq \tilde{p}_i^{n+1} \leq \tilde{p}_i^{bulk,n+1}. \tag{4.45}$$

Here $0 < \kappa_i < 1$ is a parameter which controls the rate of relaxation. In numerical results presented in this paper $\kappa_i = 0.5$.

Using (4.44), inequality (4.45) transfers to following requirement for $\Psi_{i,k}$

$$\frac{\kappa_i\, V_i^n}{\rho_i^n (a_i^n)^2} (\tilde{p}_i^{bulk,n+1} - \bar{p}) \geq \sum_{k \in M(i)} \Psi_{i,k}\, \delta V_{i,k}^{max} \geq 0, \tag{4.46}$$

The case when $\tilde{p}_i^{bulk,n+1} \leq \bar{p}$; leads to a similar inequality. It remains to solve an quadratic optimization problem with respect to limiters

$$\min_{\Psi_{i,k}} \sum_{i,k} \left(1 - \Psi_{i,k}\right)^2$$

with described linear constraints to determine the limiter values for each interface within each multimaterial cell. The limiter values and maximum fluxes for each interface are then used to update the material component volumes, volume fractions, densities and internal energies. Further details of the IA-SSD method can be found in [14].

5. Discrete cell-centered grid Lagrangian hydrodynamics – CGH

In this section we will construct a cell-centered Finite Volume discretization for the conservation laws of mass, momentum and total energy from the Lagrangian integral formulation (2.8) of the continuum mechanics equations.

5.1. Notation and assumptions

If we apply the mass conservation equation (2.8a) to a polygonal cell $\omega_c(t)$ its mass $m_c = \int_{\omega_c(t)} \rho \, \mathrm{d}v$ will remain constant in time. For any physical variable, φ, we can define its mass averaged value over the cell c

$$\varphi_c = \left(\int_{\omega_c(t)} \rho\varphi \, \mathrm{d}v \right) / m_c \, .$$

As in SGH, the cell density, is related to the cell mass and the cell volume, $|\omega_c(t)|$, by means of $m_c = \rho_c(t)|\omega_c(t)|$. Utilizing the foregoing notations and applying (2.8b), (2.8c) and the GCL (2.29) to the polygonal cell ω_c leads to the semi-discrete system of conservation laws

$$m_c \frac{d}{dt}\left(\frac{1}{\rho_c}\right) - \int_{\partial\omega_c(t)} \mathbf{u}\cdot\mathbf{n} \, \mathrm{d}s = 0, \tag{5.1a}$$

$$m_c \frac{d\mathbf{u}_c}{dt} - \int_{\partial\omega_c(t)} \mathbb{T}\mathbf{n} \, \mathrm{d}s = \mathbf{0}, \tag{5.1b}$$

$$m_c \frac{de_c}{dt} - \int_{\partial\omega_c(t)} \mathbb{T}\mathbf{n}\cdot\mathbf{u} \, \mathrm{d}s = 0. \tag{5.1c}$$

Here, $\mathbf{u}_c$ and e_c denote respectively the mass averaged values of the velocity and the total energy over $\omega_c(t)$. Let $\mathbb{T}_c$ be a constant value of the Cauchy stress tensor within the cell c, defined by $\mathbb{T}_c = -p_c\,\mathbb{I}_d$, where the thermodynamic pressure of the cell c is given by the equation of state $p_c = p(\rho_c, \varepsilon_c)$. The specific internal energy of cell c is then computed from $\varepsilon_c = e_c - \frac{1}{2}\mathbf{u}_c^2$. The main assumption in our discretization is that the physical variables are cell-centered, hence the method is termed Cell-Centered Grid Hydrodynamics (CGH). The motion of each vertex p of the polygonal grid is given by the trajectory equation (3.1).

To complete the discretization of the system of equations (5.1) it remains to determine

(1) A relevant approximation of the numerical fluxes for the discrete conservation laws (5.1a), (5.1b) and (5.1c);
(2) A relevant approximation of the vertex velocity to move the computational grid.

These two tasks shall be addressed in the next sections by invoking the following design principles

- GCL compatibility: The time rate of change of the zone volume computed from the volume flux approximation must rigorously coincide with the zone volume change resulting from the vertices motion;
- Thermodynamic consistency: The numerical fluxes for the set of conservation laws are constructed to ensure that a semi-discrete entropy inequality holds within each cell and is consistent with the second law of thermodynamics;
- Momentum and total energy conservation: The numerical fluxes are constructed in order to ensure the conservation of momentum and total energy over the whole computational domain.

Let us point out that this latter point is particularly important from a theoretical point of view since Lax–Wendroff's theorem suggests that we can hope to correctly approximate discontinuous solutions to the conservation laws.

5.2. Compatible discretization of the GCL

The compatible discretization of CGL was described in Section 3.2. For the sake of completeness, we recall its expression here

$$m_c \frac{d}{dt}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathcal{P}(c)} l_{pc}\mathbf{n}_{pc}\cdot\mathbf{u}_p = 0.$$

Let us point out that the above equation is fully compatible with the volume variation resulting from the movement of the vertices, *i.e.*, obtained from the trajectory equation $d\mathbf{x}_p/dt = \mathbf{u}_p$.

5.3. Subcell force-based discretization

Let $p^{-\frac{1}{2}}$ and $p^{+\frac{1}{2}}$ be respectively the midpoints of the edges $[p^-, p]$ and $[p, p^+]$. We recall that the subcell ω_{pc} is the quadrangle obtained by joining the points p, $p^{-\frac{1}{2}}$, c and $p^{+\frac{1}{2}}$, where c denotes the cell centroid, as shown in Fig. 2 and Fig. 3. It is clear that the set of sub cells is a partition of the cell ω_c, that is

$$\omega_c = \bigcup_{p\in\mathcal{P}(c)} \omega_{pc}.$$

The sub cell force exerted on the outer boundary of the sub cell ω_{pc} is defined by

$$\mathbf{F}_{pc} = \int_{\partial\omega_{pc}\cap\partial\omega_c} \mathbb{T}\mathbf{n}\,\mathrm{d}s. \tag{5.2}$$

This is the numerical flux for the momentum equation. Let us note that the CGH sub cell force is clearly different from the sub cell force, $\mathbf{f}_{pc}$, defined previously in the context of the SGH discretization, refer to (4.11). Now, employing the above notation, the momentum flux can also be written

$$\int_{\partial\omega_c(t)} \mathbb{T}\mathbf{n}\,\mathrm{d}s = \sum_{p\in\mathcal{P}(c)} \mathbf{F}_{pc},$$

whereas the total energy flux is approximated by means of

$$\int_{\partial\omega_c(t)} \mathbb{T}\mathbf{n}\cdot\mathbf{u}\,\mathrm{d}s = \sum_{p\in\mathcal{P}(c)} \mathbf{F}_{pc}\cdot\mathbf{u}_p.$$

We remark that the sub cell contribution to the total energy flux is the work rate of the sub cell force, that is the dot product of the sub cell force, $\mathbf{F}_{pc}$, with the vertex velocity $\mathbf{u}_p$.

Finally, we recast the system of conservation laws (5.1) into the semi-discrete form

$$m_c\frac{d}{dt}\left(\frac{1}{\rho_c}\right) - \sum_{p\in\mathcal{P}(c)} l_{pc}\mathbf{n}_{pc}\cdot\mathbf{u}_p = 0, \tag{5.3a}$$

$$m_c\frac{d\mathbf{u}_c}{dt} - \sum_{p\in\mathcal{P}(c)} \mathbf{F}_{pc} = \mathbf{0}, \tag{5.3b}$$

$$m_c\frac{de_c}{dt} - \sum_{p\in\mathcal{P}(c)} \mathbf{F}_{pc}\cdot\mathbf{u}_p = 0. \tag{5.3c}$$

To close the above semi-discrete system, it remains to determine expressions for the sub cell force, $\mathbf{F}_{pc}$, and the vertex velocity, $\mathbf{u}_p$ in terms of the cell-centered physical variables. Let us recall that the vertex velocity is also needed to move the grid by solving the trajectory equation (3.1). In the next section, we shall express the sub cell forces in terms of the cell-centered stress $\mathbb{T}_c$ and the velocity jump $\mathbf{u}_p - \mathbf{u}_c$ so that the resulting semi-discrete scheme satisfies a local entropy inequality.

5.4. Local entropy inequality

From the Gibbs relation (2.11), the differential of specific entropy can be written as

$$\mathrm{d}\eta = \frac{p}{\theta}\mathrm{d}\left(\frac{1}{\rho}\right) - \frac{\mathbf{u}}{\theta}\mathrm{d}\mathbf{u} + \frac{1}{\theta}\mathrm{d}e.$$

Considering the specific entropy as a function of the vector of physical variables, *i.e.*, $\eta = \eta(\mathbf{U})$, where $\mathbf{U} = (\frac{1}{\rho}, \mathbf{u}, e)^t$, the Gibbs relation rewritten in the compact form

$$\mathrm{d}\eta = \mathbf{V}\cdot\mathrm{d}\mathbf{U},$$

where $\mathbf{V} = \frac{1}{\theta}(p, -\mathbf{u}, 1)^t$ is the vector of entropic variables, which is simply the gradient of entropy with respect to the vector of physical variables, *i.e.*, $\mathbf{V} = \frac{\partial\eta}{\partial\mathbf{U}}$. This implies that the time rate of change of entropy is given by

$$\rho\frac{d\eta}{dt} = \mathbf{V}\cdot\rho\frac{d}{dt}\mathbf{U}.$$

Thus, the discrete analog of the time rate of change of entropy within the polygonal cell ω_c is given by

$$m_c \frac{d\eta_c}{dt} = \mathbf{V}_c \cdot m_c \frac{d}{dt}\mathbf{U}_c = \frac{1}{\theta_c}[p_c m_c \frac{d}{dt}\left(\frac{1}{\rho_c}\right) - \mathbf{u_c} \cdot m_c \frac{d\mathbf{u}_c}{dt} + m_c \frac{de_c}{dt}],$$

where $\theta_c > 0$ is the averaged temperature of the cell ω_c. Substituting the semi-discrete conservation laws (5.3a), (5.3b) and (5.3c) into the above equation leads to

$$m_c \frac{d\eta_c}{dt} = \frac{1}{\theta_c} \sum_{p \in \mathcal{P}(c)} (\mathbf{F}_{pc} - l_{pc}\mathbb{T}_c \mathbf{n}_{pc}) \cdot (\mathbf{u}_p - \mathbf{u}_c). \tag{5.4}$$

The above factorization results from the use of the trivial identity $-p_c \mathbf{n}_{pc} = \mathbb{T}_c \mathbf{n}_{pc}$ and the geometric identity (3.6).

To satisfy the entropy imbalance (2.13) the right hand-side of the above equation has to be positive. Thus, the thermodynamic consistency of the CGH scheme with the second law of thermodynamic is ensured provided that there exists a positive definite 2×2 matrix, $\mathbb{M}_{pc}$, such that the sub cell force can be written as

$$\mathbf{F}_{pc} = l_{pc}\mathbb{T}_c \mathbf{n}_{pc} + \mathbb{M}_{pc}(\mathbf{u}_p - \mathbf{u}_c). \tag{5.5}$$

This is sufficient to ensure the entropy production of the CGH scheme is positive. Dimensional analysis shows that the dimensions of the corner matrix $\mathbb{M}_{pc}$ to be mass per unit time. Moreover, this should satisfy the principle of material frame indifference. Namely, if $\mathbb{Q}$ denotes a rigid rotation, then the rigid rotation will transform the corner matrix into $\mathbb{Q}^t \mathbb{M}_{pc} \mathbb{Q}$.

Substituting (5.5) into (5.4) allows us to express the entropy production term as a quadratic function of the velocity jump $\mathbf{u}_p - \mathbf{u}_c$

$$m_c \frac{d\eta_c}{dt} = \frac{1}{\theta_c} \sum_{p \in \mathcal{P}(c)} \mathbb{M}_{pc}(\mathbf{u}_p - \mathbf{u}_c) \cdot (\mathbf{u}_p - \mathbf{u}_c) \geq 0.$$

We observe that the entropy production term is directly governed by the corner matrix $\mathbb{M}_{pc}$. Different kinds of corner matrices will be investigated in Section 5.6.

5.5. Conservation of total energy and momentum

Having determined the expression of the sub cell force in terms of the cell centered stress tensor and the velocity jump $\mathbf{u}_p - \mathbf{u}_c$, we realize that we have constructed a special Finite Volume discretization characterized by node-based numerical fluxes, *i.e.*, the sub cell forces $\mathbf{F}_{pc}$. Contrary to the usual face-based FV discretization the conservation of total energy and momentum is not trivial. We shall now derive a condition to ensure that the total energy and the momentum are globally conserved.

The conservation of total energy over the computational domain can be written

$$\sum_c m_c \frac{de_c}{dt} = \int_{\partial D(t)} \mathbb{T}\mathbf{n} \cdot \mathbf{u}\, ds,$$

where $\partial D(t)$ denotes the boundary of the moving computational domain. For the sake of conciseness, we do not derive the boundary condition treatment, the interested reader might refer for instance to [135]. Considering, for instance, the case of a free boundary condition, *i.e.*, $\mathbb{T}\mathbf{n} = \mathbf{0}$ on $\partial D(t)$, the balance of total energy turns into

$$\sum_c m_c \frac{de_c}{dt} = 0. \tag{5.6}$$

Replacing the semi-discrete equation of total energy (5.3c) in the left hand-side of the above equation yields

$$\sum_c \sum_{p \in \mathcal{P}(c)} \mathbf{F}_{pc} \cdot \mathbf{u}_p = 0.$$

Exchanging the order of summation at the left hand-side, that is summing over the nodes and summing over the cells surrounding each node, we recast the above equation as follows

$$\sum_p (\sum_{c \in \mathcal{C}(p)} \mathbf{F}_{pc}) \cdot \mathbf{u}_p = 0,$$

where $\mathcal{C}(p)$ is the set of cells surrounding node p. The foregoing condition shall be satisfied for all nodal velocity field $\mathbf{u}_p$ to ensure the conservation of total energy over the computational grid at the semi-discrete level.

Therefore, the semi-discrete CGH scheme conserves total energy if and only if the following condition is fulfilled

$$\sum_{c\in\mathcal{C}(p)} \mathbf{F}_{pc} = \mathbf{0}, \quad \text{for all } p. \tag{5.7}$$

It is interesting to observe that this condition also implies the conservation of momentum over the computational domain.

5.6. Nodal solver

Combining the expression of the sub cell force that satisfies the entropy inequality

$$\mathbf{F}_{pc} = l_{pc}\mathbb{T}_c\mathbf{n}_{pc} + \mathbb{M}_{pc}(\mathbf{u}_p - \mathbf{u}_c),$$

and the condition (5.7) allows us to write the following vectorial equation that uniquely determines the nodal velocity

$$\mathbb{M}_p\mathbf{u}_p = \sum_{c\in\mathcal{C}(p)} -l_{pc}\mathbb{T}_c\mathbf{n}_{pc} + \mathbb{M}_{pc}\mathbf{u}_c, \quad \text{for all } p, \tag{5.8}$$

where $\mathbb{M}_p$ is the sum of the corner matrices surrounding node p, *i.e.*, $\mathbb{M}_p = \sum_{c\in\mathcal{C}(p)} \mathbb{M}_{pc}$. Recalling that $\mathbb{M}_{pc}$ is symmetric positive definite, then $\mathbb{M}_p$ is also symmetric positive definite and thus invertible. This shows that the system (5.8) will always admit a unique solution. Equation (5.8) is a system of two equations for the components of $\mathbf{u}_p$. Given some choice of the $\mathbb{M}_{pc}$ matrix, all the terms in the right hand-side of (5.8) are known, because $\mathbb{T}_c$ and $\mathbf{u}_c$ are primary unknowns. Moreover, once (5.8) is solved, the nodal velocity is expressed in terms of the primary unknowns. Thus, replacing $\mathbf{u}_p$ in the sub cell force expression (5.5), shows that $\mathbf{F}_{pc}$ is expressed in terms of $\mathbb{T}_c$ and $\mathbf{u}_c$, so the spatial discretization is complete.

Recalling that $\mathbb{M}_{pc}$ has the dimension of a mass per unit time, it is worth mentioning that (5.8) can be interpreted as an instantaneous balance of momentum at node p. More precisely, introducing the SGH sub cell force, $\mathbf{f}_{pc}$, this equation can be rewritten as

$$\mathbb{M}_p\mathbf{u}_p = \sum_{c\in\mathcal{C}(p)} \mathbf{f}_{pc} + \mathbf{f}_{pc}^{vis}, \tag{5.9}$$

where $\mathbf{f}_{pc} = -l_{pc}\mathbb{T}_c\mathbf{n}_{pc} = l_{pc}p_c\mathbf{n}_{pc}$ is the pressure force and $\mathbf{f}_{pc}^{vis} = \mathbb{M}_{pc}\mathbf{u}_c$ is the viscous force induced by the nodal solver and characterized by the viscous matrix $\mathbb{M}_{pc}$. Here, the main difference with the SGH balance of momentum (4.19) is due to the presence of the nodal mass matrix, $\mathbb{M}_p$, instead of the scalar nodal mass, m_p.

5.7. The EUCCLHYD nodal solver

The nodal solver was introduced in [2,137] and the corresponding CGH scheme is called EUCCLHYD, which is shorthand for Explicit Unstructured Cell Centered Lagrangian HYDrodynamics. The EUCCLHYD solver is characterized by the following two-point quadrature rule for approximating the momentum flux

$$\int_{\partial\omega_c} \mathbb{T}\mathbf{n}\,\mathrm{d}s = -\int_{\partial\omega_c} p\mathbf{n}\,\mathrm{d}s = -\sum_{p\in\mathcal{P}(c)} (l_{pc}^- p_{pc}^- \mathbf{n}_{pc}^- + l_{pc}^+ p_{pc}^+ \mathbf{n}_{pc}^+),$$

where p_{pc}^- and p_{pc}^+ denote the two nodal pressures defined at each cell corner, as shown in Fig. 6. Here, we have one pressure per unit normal at each cell corner. These degrees of freedom have been introduced to ensure that the nodal solver coincides with the classical Godunov acoustic solver [90,91] for the case of one-dimensional planar (radial) flows aligned with Cartesian (polar) grids, refer to [137]. The EUCCLHYD sub cell force can be written as

$$\mathbf{F}_{pc}^{EU} = -(l_{pc}^- p_{pc}^- \mathbf{n}_{pc}^- + l_{pc}^+ p_{pc}^+ \mathbf{n}_{pc}^+). \tag{5.10}$$

The nodal pressures, p_{pc}^- and p_{pc}^+, are computed by approximating the Riemann invariant $\mathrm{d}p + \rho a\mathrm{d}u$ written in the direction of the unit corner normals $\mathbf{n}_{pc}^-$ and $\mathbf{n}_{pc}^+$

$$p_{pc}^- - p_c = -z_c(\mathbf{u}_p - \mathbf{u}_c)\cdot\mathbf{n}_{pc}^-, \quad \text{and} \quad p_{pc}^+ - p_c = -z_c(\mathbf{u}_p - \mathbf{u}_c)\cdot\mathbf{n}_{pc}^+. \tag{5.11}$$

Substituting the above formulas into the EUCCLHYD sub cell force yields

$$\mathbf{F}_{pc}^{EU} = -l_{pc}p_c\mathbf{n}_{pc} + z_c\left\{l_{pc}^-[(\mathbf{u}_p - \mathbf{u}_c)\cdot\mathbf{n}_{pc}^-]\mathbf{n}_{pc}^- + l_{pc}^+[(\mathbf{u}_p - \mathbf{u}_c)\cdot\mathbf{n}_{pc}^+]\mathbf{n}_{pc}^+\right\}, \tag{5.12}$$

where $z_c = \rho_c a_c$ denotes the acoustic impedance of the cell c. Utilizing once more the identity $(\mathbf{a}\otimes\mathbf{b})\mathbf{c} = (\mathbf{c}\cdot\mathbf{b})\mathbf{a}$, the expression of the sub cell force turns into

$$\mathbf{F}_{pc}^{EU} = -l_{pc}p_c\mathbf{n}_{pc} + \mathbb{M}_{pc}^{EU}(\mathbf{u}_p - \mathbf{u}_c),$$

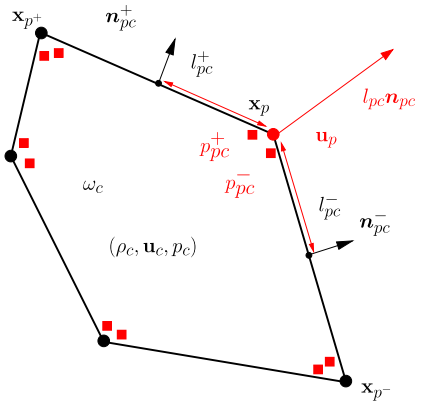

Fig. 6. Notations related to EUCCLHYD nodal solver.

where the EUCCLHYD viscous matrix is written as

$$\mathbb{M}^{EU}_{pc} = z_c \left[l^-_{pc} (\mathbf{n}^-_{pc} \otimes \mathbf{n}^-_{pc}) + l^+_{pc} (\mathbf{n}^+_{pc} \otimes \mathbf{n}^+_{pc}) \right]. \tag{5.13}$$

The corner matrix of the EUCCLHYD solver is symmetric positive definite and thus ensures non-negative entropy production.

Comment 3. The first CGH scheme which satisfied these three design principles enumerated in Section 5.1 was the GLACE scheme (Godunov-type LAgrangian scheme Conservative for total Energy) introduced in the seminal paper [69]. The spatial discretization is characterized by a one-point quadrature of the momentum and the total energy fluxes. More precisely, these numerical fluxes are computed utilizing one nodal pressure per cell corner. This nodal pressure is evaluated by approximating the Riemann invariant written in the direction of the unit corner normal $\mathbf{n}_{pc}$. It is worth mentioning the important theoretical result that the weak consistency of the GLACE scheme was demonstrated in [64].

Comment 4. Non-linear variants of the CGH formulation have been proposed. They rely on the use of a generalized impedance instead of the acoustic impedance. The generalized impedance is constructed employing the Dukowicz approximate Riemann solver [76,134].

5.8. First-order time discretization

The first-order explicit time integration of the system (5.3) of semi-discrete conservation laws over the time interval $[t^n, t^{n+1}]$ leads to

$$m_c \left[\left(\frac{1}{\rho_c} \right)^{n+1} - \left(\frac{1}{\rho_c} \right)^{n} \right] - \Delta t^n \sum_{p \in \mathcal{P}(c)} (l_{pc} \mathbf{n}_{pc})^{n+\frac{1}{2}} \cdot \mathbf{u}^n_p = 0, \tag{5.14a}$$

$$m_c (\mathbf{u}^{n+1}_c - \mathbf{u}^n_c) - \Delta t^n \sum_{p \in \mathcal{P}(c)} \mathbf{F}^n_{pc} = \mathbf{0}, \tag{5.14b}$$

$$m_c (e^{n+1}_c - e^n_c) - \Delta t^n \sum_{p \in \mathcal{P}(c)} \mathbf{F}^n_{pc} \cdot \mathbf{u}^n_p = 0. \tag{5.14c}$$

Here, the superscript n denotes the value of the physical variables at time t^n and $\Delta t^n = t^{n+1} - t^n$ is the time step. The superscript $n+\frac{1}{2}$ utilized in (5.14a) correspond to a time centered integration at $t^n + \frac{\Delta t^n}{2}$. This particular choice is motivated by the need to satisfy rigorously the GCL at the discrete level. We point out that it does not require any implicit time integration as we shall explain in Section 5.8.2. It remains to detail how the numerical fluxes are computed.

5.8.1. Nodal fluxes computation

The nodal solver is required to compute both the nodal velocity, $\mathbf{u}^n_p$, and the sub cell force, $\mathbf{F}^n_{pc}$. Once the viscous matrix is chosen, $\mathbb{M}^n_{pc}$, the nodal velocity is obtained by solving the following system

$$\mathbb{M}^n_p \mathbf{u}^n_p = \sum_{c \in \mathcal{C}(p)} l^n_{pc} p\mathbb{T}^n_c \mathbf{n}^n_{pc} + \mathbb{M}^n_{pc} \mathbf{u}^n_c, \tag{5.15}$$

where the nodal mass matrix is written as $\mathbb{M}^n_p = \sum_{c \in \mathcal{C}(p)} \mathbb{M}^n_{pc}$. Then, the sub cell force can be computed by means of

$$\mathbf{F}_{pc}^{n} = l_{pc}^{n}\,\mathbb{T}_{c}^{n}\mathbf{n}_{pc}^{n} + \mathbb{M}_{pc}^{n}(\mathbf{u}_{p}^{n} - \mathbf{u}_{c}^{n}), \tag{5.16}$$

using the expression for the nodal velocity obtained from (5.15). At this point all the numerical fluxes have been determined from the primary unknowns and the geometric quantities evaluated at time level t^n.

5.8.2. Discrete GCL and trajectory equation

The explicit time integration of the trajectory equation which is required to move the grid can be written as

$$\mathbf{x}_{p}^{n+1} = \mathbf{x}_{p}^{n} + \Delta t^{n}\mathbf{u}_{p}^{n}. \tag{5.17}$$

This means that the updated geometry is known once the nodal velocities have been computed. The cell volume, $|\omega_c^{n+1}|$, may also then be determined directly from the geometry at time t^{n+1} and hence the density can then be obtained as follows via mass conservation

$$\rho_{c}^{n+1} = m_c/|\omega_{c}^{n+1}|\,. \tag{5.18}$$

The updated density may also be computed from the discrete GCL (5.14a) by integrating the volume flux exactly with respect to time. Given the corner normal, $l_{pc}\mathbf{n}_{pc}$, is a linear function with respect to time, the volume flux will be exactly computed provided that at least a midpoint or trapezoidal quadrature rule is employed, hence the $n+\frac{1}{2}$ superscript in (5.14a). Finally, computing the time centered corner normal as

$$(l_{pc}\mathbf{n}_{pc})^{n+\frac{1}{2}} = \frac{1}{2}[(l_{pc}\mathbf{n}_{pc})^{n} + (l_{pc}\mathbf{n}_{pc})^{n+1}], \tag{5.19}$$

allows us to write the discrete GCL equation as

$$|\omega_{c}^{n+1}| - |\omega_{c}^{n}| = \Delta t^{n} \sum_{p\in\mathcal{P}(c)} (l_{pc}\mathbf{n}_{pc})^{n+\frac{1}{2}} \cdot \mathbf{u}_{p}.$$

Thus, we conclude that the density computed from the above discrete GCL is consistent with the density computed from the updated geometry (5.18).

5.8.3. Thermodynamic closure

The updated cell-centered stress tensor can be written $\mathbb{T}_{c}^{n+1} = -p_{c}^{n+1}\,\mathbb{I}_{d}$, where the updated pressure, p_c^{n+1}, is obtained from the equation of state

$$p_{c}^{n+1} = p(\rho_{c}^{n+1}, \varepsilon_{c}^{n+1}).$$

The updated specific internal energy is then given by subtracting the kinetic energy from the total energy, *i.e.*, $\varepsilon_{c}^{n+1} = e_{c}^{n+1} - \frac{1}{2}(\mathbf{u}_{c}^{n+1})^2$.

5.8.4. Time step monitoring

As with the SGH discrete scheme described in Section 4.10, the CGH discrete scheme requires a time step control. It also relies on two criteria: the first one limits the cell volume variation and ensures the positivity of the cell volume, while the second one is a kind of CFL condition, which ensures the positivity of the updated specific internal energy.

5.8.4.1. Volume variation limiting We are looking for a time step constraint, $\Delta t^n \leq \Delta t_v$, such that for all cells

$$\frac{||\omega_{c}^{n+1}| - |\omega_{c}^{n}||}{|\omega_{c}^{n}|} \leq C_v, \tag{5.20}$$

where $C_v \in]0,1[$ is a user-defined parameter. Employing a first-order time discretization for the GCL, (5.20) amounts to imposing $\Delta t^n \leq C_v \Delta t_v$ where the upper bound can be written as

$$\Delta t_v = \min_c \frac{|\omega_{c}^{n}|}{|\sum_{p\in\mathcal{P}(c)}(l_{pc}\mathbf{n}_{pc})^{n}\cdot\mathbf{u}_{p}^{n}|} = \min_c \left[\frac{1}{\mathcal{DIV}_{c}^{n}(\mathbf{u})}\right]. \tag{5.21}$$

Here, $\mathcal{DIV}_{c}^{n}(\mathbf{u})$ is the discrete divergence operator defined over the cell c at time t^n, as discussed in Comment 2.

5.8.4.2. CFL-like criterion We briefly discuss the construction of a constraint on the time step written under the form $\Delta t^n \leq \Delta t_e$ required to ensure specific internal energy resulting from the CGH discrete scheme remains positive, refer to [177]. Combining the discrete conservation laws leads to a discrete internal energy equation of the form

$$\varepsilon_c^{n+1} = \varepsilon_c^n - \Delta t^n \frac{p_c^n}{\rho_c^n} \mathcal{DIV}_c^n(\mathbf{u}) + \frac{\Delta t^n}{m_c} \mathcal{Q}(\mathbf{u}_p^n - \mathbf{u}_c^n).$$

The last term on the right hand-side is a quadratic form expressed in terms of a velocity jump $\mathbf{u}_p^n - \mathbf{u}_c^n$, which also depends on the viscous matrix, $\mathbb{M}_{pc}$, the time step and the cell mass. A sufficient condition to ensure the positivity of the above quadratic form for the EUCCLHYD solver can be written as

$$\Delta t^n \leq \frac{2m_c}{z_c^n \sum_{p\in\mathcal{P}(c)} l_{pc}^{-,n} + l_{pc}^{+,n}},$$

where $z_c^n = \rho_c^n a_c^n$ is the acoustic impedance of the cell c and $l_{pc}^{\pm,n}$ denotes the length of the half-edges that impinges on node p. Similar conditions can be easily derived for the other nodal solvers, refer to [177]. Finally, there exists a time step constraint of the form $\Delta t^n \leq \Delta t_e$ to ensure the positiveness of the foregoing quadratic form. The upper bound on the time step can be written as

$$\Delta t_e = \min_c \left(\frac{l_c^n}{a_c^n} \right), \tag{5.22}$$

where l_c^n is some characteristic length related to the cell c. In the case of the EUCCLHYD solver this characteristic length takes the explicit form

$$l_c^{n,EU} = 2 \frac{|\omega_c^n|}{\sum_{\in\mathcal{P}(c)} l_{pc}^{-,n} + l_{pc}^{+,n}}.$$

This characteristic length is equal to two times the cell area divided by its perimeter. In the case of a Cartesian rectangular cell $l_c^n = \frac{\Delta x^n \Delta y^n}{\Delta x^n + \Delta y^n}$.

Finally, granted that $\Delta t^n \leq \Delta t_e$, where Δt_e is defined by (5.22), the updated specific internal energy satisfies

$$\varepsilon_c^{n+1} \geq \varepsilon_c^n - \Delta t^n \frac{p_c^n}{\rho_c^n} \mathcal{DIV}_c^n(\mathbf{u}).$$

Assuming that the time step obeys the constraint on the volume variation, *i.e.*, $\Delta t^n \leq C_v \Delta t_v$, implies that $\Delta t^n |\mathcal{DIV}_c^n(\mathbf{u})| \leq C_v$. Thus, the updated specific internal energy will remain positive provided that the user-defined coefficient, C_v, satisfies the constraint

$$C_v \leq \rho_c^n \varepsilon_c^n / p_c^n. \tag{5.23}$$

For a gamma gas law, $p = (\gamma - 1)\rho\varepsilon$, and the above constraint boils down to $C_v \leq \frac{1}{\gamma - 1}$ which does not represent a too severe a constraint.

We conclude this section by giving the formula that is used in practice to control the time step evolution

$$\Delta t^{n+1} = \min(C_m \Delta t^n, C_v \Delta t_v, C_e \Delta t_e), \tag{5.24}$$

where C_m, C_v and C_e are user-defined parameters. C_m is a multiplicative coefficient usually set to $C_m = 1.05$, whereas C_e is a safety coefficient fixed to $C_e = 0.3$ and C_v might be set to $C_v = 0.1$. We note that the time step control ensures the positivity of the updated density and the specific internal energy provided that $C_v \in]0, 1[$ satisfies the constraint (5.23) and Δt_e is computed according to (5.22).

5.9. Second-order extension – design principles

For the sake of conciseness we shall not detail the second-order extension of the CGH scheme knowing that many approaches are available. This section aims to summarize the main ingredients required to develop such an extension. The simplest approach relies on a two-stage time integrator such as a Predictor–Corrector procedure. Regarding the second-order space discretization, it consists in computing the numerical fluxes, nodal velocity and sub cell force, by means of the nodal solver wherein the piece-wise constant values of the stress tensor and the velocity within cell c are replaced by their nodal extrapolated values resulting from their piece-wise linear reconstructions. The monotonicity of the piece-wise linear reconstruction is ensured by means of a limiting procedure, which ensures that the nodal extrapolated values of the physical variables remain inside the bounds determined by their piece-wise constant values taken in the local neighborhood of the target cell.

It is worth mentioning that a one-stage time integrator might also be employed to construct the second-order extension. In this alternative approach, the numerical fluxes are evaluated at time $t^{n+\frac{1}{2}} = t^n + \frac{\Delta t}{2}$, where Δt is the current time step, by means of a Taylor expansion. The numerical fluxes are then computed by means of the nodal solver using nodal extrapolated values of the cell-centered stress and velocity. In addition, the time derivatives of the numerical fluxes are computed by means of the nodal solver in which the time derivative of the primary unknowns are expressed in terms of their gradients by means of a Lax–Wendroff elimination procedure. This approach, which is nothing but the extension of the GRP (Generalized Riemann Problem) methodology, introduced by Ben Artzi and Falcovitz [23], has been successfully applied to two-dimensional Lagrangian hydrodynamics [134].

5.9.1. Second-order nodal solver

We assume that all physical variables and all the geometric quantities are known at time t^n. The second-order extension of the nodal solver still relies on the sub cell force balance equation at node p

$$\sum_{c\in\mathcal{C}(p)} \widetilde{\mathbf{F}}_{pc}^n = \mathbf{0}. \tag{5.25}$$

Here, the tilde symbol denotes the second-order extension of the sub cell force which can be written

$$\widetilde{\mathbf{F}}_{pc}^n = l_{pc}^n \widetilde{\mathbb{T}}_c^n(\mathbf{x}_p^n)\mathbf{n}_{pc}^n + \mathbb{M}_{pc}^n\left[\mathbf{u}_p^n - \widetilde{\mathbf{u}}_c^n(\mathbf{x}_p^n)\right]. \tag{5.26}$$

The first-order sub cell force utilizes the cell-centered piece-wise constant values of the stress tensor and the velocity, whereas its second-order extension employs the nodal extrapolated values of the stress and the velocity. These nodal extrapolated values are obtained thanks to the piece-wise linear representation of the stress tensor and the velocity within cell c, which are respectively denoted by $\widetilde{\mathbb{T}}_c^n(\mathbf{x})$ and $\widetilde{\mathbf{u}}_c^n(\mathbf{x})$. Substituting the second-order expression of the sub cell force (5.26) into the balance equation (5.25) provides us with the vectorial equation which is satisfied by the nodal velocity

$$\mathbb{M}_p^n \mathbf{u}_p^n = \sum_{c\in\mathcal{C}(p)} -l_{pc}^n \widetilde{\mathbb{T}}_c^n(\mathbf{x}_p^n)\mathbf{n}_{pc}^n + \mathbb{M}_{pc}^n \widetilde{\mathbf{u}}_c^n(\mathbf{x}_p^n), \tag{5.27}$$

where $\mathbb{M}_{pc}^n$ is the viscous corner matrix and $\mathbb{M}_p^n = \sum_{c\in\mathcal{C}(p)} \mathbb{M}_{pc}^n$ is the nodal mass matrix. We observe that once the viscous corner matrix has been chosen the nodal velocity and the sub cell force are completely defined by (5.27) and (5.26).

The piece-wise linear reconstruction used in the second-order extension of the nodal solver requires consistent approximations for the gradient of the stress tensor and the velocity within each cell. Being given the set of the piece-wise constant values of a generic physical variable (stress tensor and velocity components) over the computational grid, we might compute its gradient utilizing the classical least squares approach, as described in [135]. We note that many alternative techniques are available to approximate the gradient. One might employ the nodal values of the numerical fluxes (pressure and velocity) to construct cell-centered approximations of the corresponding gradients by means of the Green formula, refer for instance to [53]. It is also possible to increase the accuracy of the reconstruction by employing node-centered gradient approximations, refer to [44].

Finally, the aforementioned piece-wise linear reconstruction must be completed by a limiting procedure to prevent the occurrence of new extrema with respect to the neighboring bounds of the physical variables. For the scalar variables the usual limiting procedure relies on well known limiter, [63,78,19], which often referred as Barth–Jespersen limiter. It may be straightforwardly extended to vector and tensor fields applying it separately to each component. However, such a procedure is frame dependent and thus leads to rotational symmetry distortion. Namely, component limiters do not preserve symmetry since a rotation of the coordinate axis produces different results. This flaw is particularly crucial in the framework of Lagrangian hydrodynamics since we are dealing with moving mesh discretizations that are particularly sensitive to symmetry breaking. To correct this flaw, the limiting procedure for vector and tensor fields must be frame indifferent.

One possible choice consists of defining within each cell a local reference orthonormal basis attached to the fluid flow. For instance, one may choose the eigenvectors of the discrete deformation tensor of the cell. This orthonormal basis is naturally frame invariant. Projecting the vector (tensor) onto this basis, we apply the foregoing limiter to the projected components and then we map them back to the Cartesian basis. The interested reader will find a detailed presentation of this approach in [135].

Another very promising approach is the Vector Image Polygon (VIP) methodology introduced by Luttwak and Falcovitz in [132]. This method provides a rigorous mathematical framework to extend the notion of discrete maximum principle to vector and tensor fields. It consists in constructing the convex hull of the vector (tensor)-space points corresponding to the neighboring cell-centered vectors (tensors). If the nodal extrapolated vector (tensor) lies inside the convex hull, then the piece-wise linear representation is monotonicity preserving, otherwise, a limiting procedure is required. This limiting procedure consists in modifying the slope by projecting the nodal extrapolated value on the boundary of the convex hull. This methodology has been successfully applied in many works, such as [99].

5.9.2. Predictor–corrector algorithm for the CGH Lagrangian scheme

- Predictor

Step 1 Compute the nodal velocity, $\mathbf{u}_p^n$, using the second-order extension of the nodal solver and the sub cell force, $\widetilde{\mathbf{F}}_p^n$, following the methodology exposed in Section 5.9.1;
Step 2 Solve the trajectory equation and update all the geometric quantities;
Step 3 Update the primary unknowns (mass density, velocity and specific total energy) solving

$$m_c\left[\left(\frac{1}{\rho_c}\right)^{n+1,\mathrm{pr}}-\left(\frac{1}{\rho_c}\right)^{n}\right]-\Delta t^n\sum_{p\in\mathcal{P}(c)}(l_{pc}\mathbf{n}_{pc})^{n+\frac{1}{2},\mathrm{pr}}\cdot\mathbf{u}_p^n=0,$$
$$m_c(\mathbf{u}_c^{n+1,\mathrm{pr}}-\mathbf{u}_c^n)-\Delta t^n\sum_{p\in\mathcal{P}(c)}\widetilde{\mathbf{F}}_{pc}^n=\mathbf{0},$$
$$m_c(e_c^{n+1,\mathrm{pr}}-e_c^n)-\Delta t^n\sum_{p\in\mathcal{P}(c)}\widetilde{\mathbf{F}}_{pc}^n\cdot\mathbf{u}_p^n=0;$$

Step 4 Update the thermodynamic quantities using the equation of state

$$\varepsilon_c^{n+1,\mathrm{pr}}=e_c^{n+1,\mathrm{pr}}-\frac{1}{2}(\mathbf{u}_c^{n+1,\mathrm{pr}})^2,\quad \mathbb{T}_c^{n+1,\mathrm{pr}}=-p_c^{n+1,\mathrm{pr}}\mathbb{I}_d,$$
$$p_c^{n+1,\mathrm{pr}}=p(\rho_c^{n+1,\mathrm{pr}},\varepsilon_c^{n+1,\mathrm{pr}}).$$

In the mass density equation, the time centered corner normal is defined by

$$(l_{pc}\mathbf{n}_{pc})^{n+\frac{1}{2},\mathrm{pr}}=\frac{1}{2}[(l_{pc}\mathbf{n}_{pc})^n+(l_{pc}\mathbf{n}_{pc})^{n+1,\mathrm{pr}}],$$

to satisfy the compatibility with the discrete GCL, as discussed in Section 5.8.2.

- Corrector

Step 1 Compute the nodal velocity, $\mathbf{u}_p^{n+1,\mathrm{pr}}$, using the second-order extension of the nodal solver and the sub cell force, $\widetilde{\mathbf{F}}_p^{n+1,\mathrm{pr}}$, following the methodology discussed in Section 5.9.1;
Step 2 Solve the trajectory equation and update all the geometric quantities;
Step 3 Update the primary unknowns (density, velocity and specific total energy) solving

$$m_c\left[\left(\frac{1}{\rho_c}\right)^{n+1}-\left(\frac{1}{\rho_c}\right)^{n}\right]-\Delta t^n\sum_{p\in\mathcal{P}(c)}(l_{pc}\mathbf{n}_{pc})^{n+\frac{1}{2}}\cdot\mathbf{u}_p^{n+\frac{1}{2}}=0,$$
$$m_c(\mathbf{u}_c^{n+1}-\mathbf{u}_c^n)-\Delta t^n\sum_{p\in\mathcal{P}(c)}\widetilde{\mathbf{F}}_{pc}^{n+\frac{1}{2}}=\mathbf{0},$$
$$m_c(e_c^{n+1}-e_c^n)-\Delta t^n\sum_{p\in\mathcal{P}(c)}\widetilde{\mathbf{F}}_{pc}^{n+\frac{1}{2}}\cdot\mathbf{u}_p^{n+\frac{1}{2}}=0;$$

Step 4 Update the thermodynamic quantities using the equation of state

$$\varepsilon_c^{n+1}=e_c^{n+1}-\frac{1}{2}(\mathbf{u}_c^{n+1})^2,\quad \mathbb{T}_c^{n+1}=-p_c^{n+1}\mathbb{I}_d,\quad p_c^{n+1}=p(\rho_c^{n+1},\varepsilon_c^{n+1}).$$

The time centered nodal velocity and sub cell force are computed as the arithmetic averages of their counterparts at time t^n and at the end of predictor stage.

5.10. Mixed cells treatment

In extending the CCH schemes into a full multi-material ALE scheme, as for example described in [84], required the ability to handle multi-material cells within the computational domain. This requires a method to determine the time evolution of the material components within these multi-material cells during the Lagrange phase of the three stage ALE algorithm. Namely, knowing the global change of volume, momentum and total energy, we need to be able to distribute it over the material components of the multi-material cells. To this end, we shall define the average zonal pressure, the average zonal sound speed and the distribution of internal energy over the materials. In what follows, for the sake of conciseness, we shall address the three foregoing issues for the equal compressibility closure model, already described in Section 4.11.1 in the context of SGH. The interested reader might refer to [84] for a detailed presentation of this topic.

The equal compressibility closure model assumes that the volume fraction of each material remains constant during the Lagrangian phase, *i.e.*, $\alpha_{c,i}^{n+1}=\alpha_{c,i}^n$, refer to Section 4.11.1 for the definition of the notations. Thus, as in SGH, the average zonal pressure, $\tilde{p}_c^n$ is given by $\tilde{p}_c^n=\sum_{i\in M(c)}\alpha_{c,i}^n p_{c,i}^n$. The average zonal sound speed is computed by means of

$$(a_{c,i}^n)^2 = \frac{1}{\rho_c^n}\left[\sum_{i\in M(c)} \alpha_{c,i}^n \rho_{c,i}^n (a_{c,i}^n)^2\right]. \tag{5.28}$$

Finally, we explain how to update the internal energy of each material within the framework of the first-order discretization. Knowing the updated values of the primary unknowns (density, cell-centered velocity, total energy) over the multi-material cell, we recast the internal energy variation over the time step as follows

$$m_c(\varepsilon_c^{n+1} - \varepsilon_c^n) + p_c^n(V_c^{n+1} - V_c^n) = \Delta t^n Q_c^n. \tag{5.29}$$

Here, the updated internal energy results from subtracting the kinetic energy to the total energy, that is $\varepsilon_c^{n+1} = e_c^{n+1} - \frac{1}{2}(\mathbf{u}_c^{n+1})^2$. The right hand-side of the above equation corresponds to some numerical entropy production which should remain non-negative provided the time step control described in Section 5.8.4 is applied. We observe that (5.29) corresponds to some first-order in time approximation of the Gibbs relation. Let us point out that (5.29) is not a new equation but simply a consequence of the CGH discretization. Bearing this in mind, we propose to distribute internal energy over the materials of the mixed cells according to the following discrete equation

$$m_{c,i}(\varepsilon_{c,i}^{n+1} - \varepsilon_{c,i}^n) + p_{c,i}^n(V_{c,i}^{n+1} - V_{c,i}^n) = \Delta t^n Q_{c,i}^n, \tag{5.30}$$

where the source term, $Q_{c,i}^n$, has to be defined. To this end, we check the consistency of this discrete equation with respect to (5.29). Recalling that $\alpha_{c,i}^n = \alpha_{c,i}^{n+1}$, $p_c^n = \sum_i \alpha_{c,i}^n p_{c,i}^n$ and summing (5.30) over all the mixed cell materials yields

$$m_c(\varepsilon_c^{n+1} - \varepsilon_c^n) + p_c^n(V_c^{n+1} - V_c^n) = \Delta t^n \sum_{i\in M(c)} Q_{c,i}^n.$$

Comparing this equation to (5.29) leads to the following definition of the source term

$$Q_{c,i}^n = \alpha_{c,i}^n Q_c^n. \tag{5.31}$$

The source term applies equal rate of volumetric heating to each material in the multi-material cell. This also ensures the correct total internal energy is maintained for each multi-material zone.

6. Discrete staggered grid Lagrangian hydrodynamics – high-order finite element approach for SGH

In this section we review the approach originally described in [73] which is based on a generic, finite dimensional, weak variational formulation of the Euler equations in a Lagrangian frame. The approach is valid for a wide variety of discrete functions spaces; in this paper we consider (arbitrarily) high-order finite elements defined on 2D quadrilateral meshes. This approach can be viewed as the high-order generalization of the SGH discretization described in Section 4; the details of this connection are discussed in [73].

This approach can be considered as high-order extension of SGH discretization described in Section 4.

6.1. Kinematic and thermodynamic basis functions

A semi-discrete Lagrangian discretization is determined by two finite dimensional spaces on the initial domain $D(0)$:

- A continuous, vector valued, *kinematic* space $\mathcal{U} \subset [H^1(D(0))]^d$, with a basis $\{W_i\}_{i=1}^{N_\mathcal{U}}$,
- A discontinuous, scalar valued, *thermodynamic* space $\mathcal{E} \subset L_2(D(0))$, with a basis $\{\phi_i\}_{j=1}^{N_\mathcal{E}}$.

For the specific case of 2D finite elements defined on quadrilateral meshes, the basis functions for the kinematic and thermodynamic spaces can be constructed by tensor products of 1D interpolating polynomials defined over the unit interval

$$\eta_{ij}(x, y) = \eta_i(x)\eta_j(y),$$

which is illustrated in Fig. 7. The interpolation points for the 1D basis functions $\eta_i(x)$ are arbitrary, but in practice we use both Gauss–Lobatto and Gauss–Legendre points.

6.2. Position and deformation

Finite element "cells" are defined through a parametric mapping defined with respect to a reference space configuration as

$$\omega_c(t) \equiv \{\mathbf{x} = \hat{\mathbf{\Phi}}_c(\hat{\mathbf{X}}, t) \ : \ \hat{\mathbf{X}} \in \hat{\omega}_c\} \tag{6.1}$$

where $\hat{\omega}_c$ is the reference cell (or element), e.g. the unit square for quadrilateral meshes. Throughout the remainder of this section, all quantities defined with respect to this reference space configuration will be accented with a "hat" symbol.

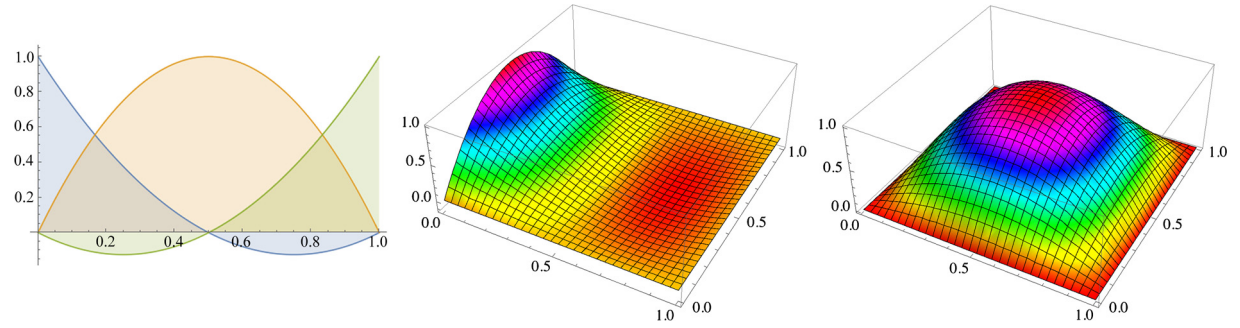

Fig. 7. 1D Q2 Gauss–Lobatto basis functions, $\eta_i(x)$, defined over the unit interval (*left*) and 2D bi-quadratic versions, $\eta_i(x, y)$ (*right*).

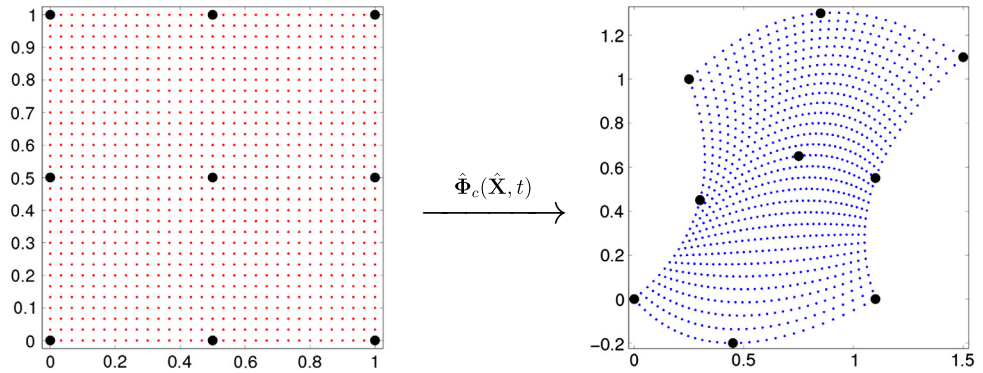

Fig. 8. Example of reference space mapping for the case of a bi-quadratic basis. The black points represent the nodal position values, $\mathbf{x}_c(t)$. It is important to note that the mapping defines a function which varies over the entire cell.

We discretize the Lagrangian position variable, $\mathbf{x}$, using Gauss–Lobatto (nodal) basis functions defined on the reference element $\hat{\eta}_i$ as in Fig. 7. The reference space mapping for a given cell $\omega_c(t)$ is then given by

$$\hat{\mathbf{\Phi}}_c(\hat{\mathbf{X}}, t) = \sum_i \mathbf{x}_{c,i}(t)\hat{\eta}_i(\hat{\mathbf{X}}) \tag{6.2}$$

where $\mathbf{x}_c(t)$ is vector of nodal position values associated with cell c, a subset of the vector $\mathbf{x}(t)$ of size $N_{\mathcal{V}}$ which contains all of the discrete nodal position values. An illustration of this mapping is shown in Fig. 8.

The kinematic space $\mathcal{U}(t)$ is used to discretize the Lagrangian position variable $\mathbf{x}$ as well as kinematic variables like velocity $\mathbf{u}$. With any discrete Lagrangian position $\mathbf{x}_p$ we associate d kinematic vector basis functions of the form

$$W_{p_1} = (\eta_p, 0), \qquad W_{p_2} = (0, \eta_p), \qquad \text{in 2D}\,.$$

The scalar function η_p is non-zero only in cells ω_c that contain the point p. It is by definition continuous, which means it is single valued for any Lagrangian position value, including points that share the same physical location at shared cell boundaries. In contrast, the thermodynamic space is discontinuous to account for material interfaces and is therefore not necessarily single valued for any given Lagrangian position value. This is illustrated in Fig. 10.

Since the kinematic and thermodynamic basis functions on the reference element are independent of time, the global basis functions are defined to move with the mesh and are therefore constant along all particle trajectories implying that

$$\frac{dW_i}{dt} = 0 \qquad \text{and} \qquad \frac{d\phi_j}{dt} = 0\,. \tag{6.3}$$

This means that the basis functions maintain there interpolation properties independent of the mesh configuration. This is illustrated in Fig. 9 as well as Fig. 10.

6.3. Strong mass conservation principle

The Jacobian matrix corresponding to the reference space mapping for a given cell c is

$$\hat{\mathbb{F}}_c = \frac{\partial \hat{\mathbf{\Phi}}_c}{\partial \hat{\mathbf{X}}}(\hat{\mathbf{X}}, t). \tag{6.4}$$

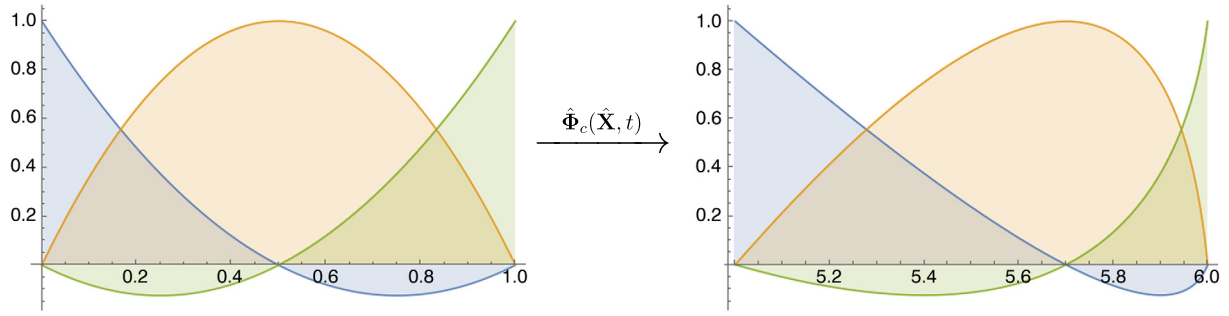

Fig. 9. Illustration of the Lagrangian nature of the basis functions. A quadratic basis $\{\hat{W}\}$ defined on the 1D reference element $\hat{\mathbf{x}} = \{0, 0.5, 1\}$ (*left*) and the transformed basis $\{W\}$ on a deformed 1D element given by $\mathbf{x} = \{5, 5.7, 6\}$.

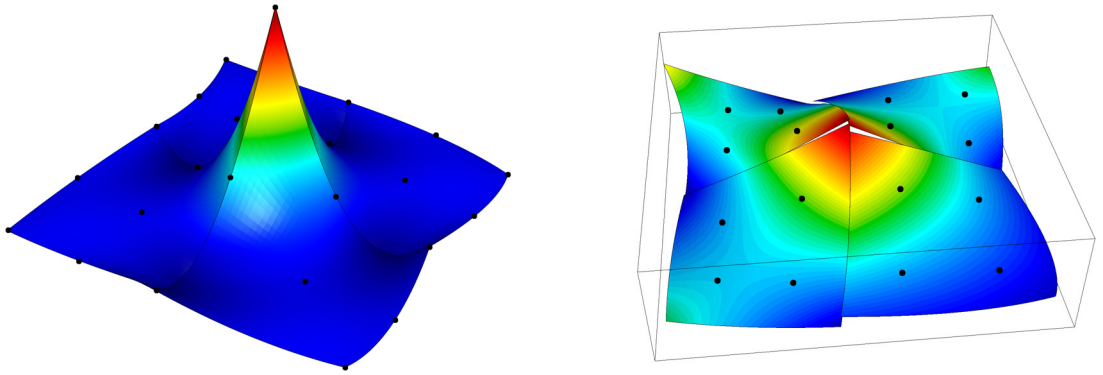

Fig. 10. Examples of bi-quadratic (Q_2) continuous and bi-linear (Q_1) discontinuous finite element basis functions defined on a 4 element bi-quadratic (Q_2) quadrilateral mesh. Note that the continuous Q_2 function is single valued at all points, including cell boundaries while the discontinuous function is multi-valued at cell boundaries. In both cases, the basis functions are obtained by mapping the reference space values via a Q_2 element transformation.

For bijective mappings (*i.e.* non-twisted elements), it is a non-singular function defined over each zone and its determinant, J_c, can be viewed as a local (or point-wise) volume function since

$$|\omega_c(t)| = \int_{\omega_c(t)} J_c \, .$$

Furthermore, the total mass in a cell is

$$m_c \equiv \int_{\omega_c(t)} \rho \, J_c \, .$$

The Lagrangian perspective requires that

$$\frac{dm_c}{dt} = 0 \, .$$

However, when dealing with high-order finite elements, both density and volume are now discrete *functions* which vary inside of a cell; so this is not a strong enough statement. We therefore utilize the so called strong (or point-wise) mass conservation principle which is derived by considering the limiting case of

$$\int_{\omega'(t)} \rho(t) = \int_{\omega'(0)} \rho(0) \quad \longrightarrow \quad \int_{\hat{\omega}'} \hat{\rho}(t) \, J_c(t) = \int_{\hat{\omega}'} \hat{\rho}(0) \, J_c(0)$$

for any $\omega'(0) \subset \omega(0)$. This leads to the point-wise equality:

$$\rho(t) \, J_c(t) = \rho(0) \, J_c(0) \, . \tag{6.5}$$

6.4. Semi-discrete motion

The discrete velocity field corresponding to the mesh motion is given by

$$\mathbf{u}(\mathbf{X},t) = \sum_i \frac{d\mathbf{x}_i}{dt}(t) W_i(\mathbf{X}) = \mathbf{u}(t)^{\mathrm{tr}} \mathbf{W}(\mathbf{X})$$

Specifically,

$$\mathbf{u} = \frac{d\mathbf{x}}{dt}\,.$$

6.5. Semi-discrete momentum conservation

We formulate a discrete momentum conservation equation by applying a variational formulation to the continuous equation (2.36). Using a Galerkin approach (at a given time t) we multiply (2.36) by a moving basis test function $W_j \in \mathcal{V}(t)$ and integrate over $\omega(t)$:

$$\int_{\omega(t)} \rho \frac{d\mathbf{u}}{dt} \cdot W_j = \int_{\omega(t)} (\nabla \cdot \mathbb{T}) \cdot W_j\,. \tag{6.6}$$

Performing integration by parts on the right hand side, we obtain

$$\int_{\omega(t)} \rho \frac{d\mathbf{u}}{dt} \cdot W_j = -\int_{\omega(t)} \mathbb{T} : \nabla W_j + \int_{\partial\omega(t)} n \cdot \mathbb{T} \cdot W_j, \tag{6.7}$$

where n is the outward pointing unit normal vector of the surface $\partial\omega(t)$. Assuming the boundary integral term vanishes (which is the case e.g. for boundary conditions $\mathbf{u}\cdot n = 0$ and $\mathbb{T} = -pI$) and expanding the velocity in the moving basis gives us

$$\sum_i \frac{d\mathbf{u}_i}{dt} \int_{\omega(t)} \rho\, W_i \cdot W_j = -\int_{\omega(t)} \mathbb{T} : \nabla W_j\,. \tag{6.8}$$

In other words,

$$\mathbf{M}_{\mathcal{U}} \frac{d\mathbf{u}}{dt} = -\int_{w(t)} \mathbb{T} : \nabla \mathbf{W}, \tag{6.9}$$

where $\mathbf{M}_{\mathcal{U}}$ is the *kinematic mass matrix* which is defined by the integral

$$\mathbf{M}_{\mathcal{U}} \equiv \int_{\omega(t)} \rho \mathbf{W}\mathbf{W}^{\mathrm{tr}}\,. \tag{6.10}$$

The kinematic mass matrix is sparse and computed over the entire mesh $D(t)$ from the contributions of the degrees of freedom associated with each individual cell c as $\mathbf{M}_{\mathcal{U}} = \mathit{Assemble}(\mathbf{M}_{\mathcal{U},c})$. The process of global assembly is analogous to the concept of "nodal accumulation" that is used in a traditional SGH method where a quantity at a node is defined to be the sum of contributions from all of the zones which share this node. The difference here is that shared degrees of freedom may occur not only at nodes, but also at shared edges/faces (2D/3D).

An important feature of this approach is that this mass matrix is independent of time due to (2.17) and (6.3):

$$\frac{d\mathbf{M}_{\mathcal{U}}}{dt} = \frac{d}{dt}\int_{w(t)} \rho \mathbf{W}\mathbf{W}^{\mathrm{tr}} = \int_{\omega(t)} \rho \frac{d}{dt}(\mathbf{W}\mathbf{W}^{\mathrm{tr}}) = 0\,. \tag{6.11}$$

6.6. Approximation of integrals

In practice, the integrals for computing the kinematic mass matrix (as well as other matrices and right-hand-side vectors) are calculated by transforming the integrals over each Lagrangian cell $\omega_c(t)$ to the standard reference cell $\hat{\omega}_c$ by using the parametric mapping of (6.2). Applying this transformation to a general integral over a given Lagrangian cell gives

$$\int_{\omega_c(t)} f = \int_{\hat{\omega}_c} (f \circ \hat{\mathbf{\Phi}})\, J_c,$$

for some integrand f, where "$\circ$" denotes composition. We approximate these integrals using a quadrature rule of a specified order. A general integral over a Lagrangian mesh cell is therefore replaced with a weighted sum of the form

$$\int_{\hat{\omega}_c} (f \circ \hat{\mathbf{\Phi}})\, J_c \approx \sum_{n=1}^{N_q} \alpha_n \left\{ (f \circ \hat{\mathbf{\Phi}})\, J_c \right\}_{\hat{\hat{x}}=\hat{\hat{q}}_n}, \tag{6.12}$$

where α_n are the N_q quadrature weights and $\hat{\hat{q}}_n$ are quadrature points inside of the reference zone where the integrand is sampled at. It is important to emphasize that the integrand, f is a function which is sampled at each quadrature point inside of a cell. The use of quadrature in computing the integrals is not always exact (depending on the functional form of the integrand and the order of the quadrature rule), and we have therefore introduced an additional approximation to the solution of the continuum equations. In practice, we use Gauss–Legendre quadrature on quadrilaterals.

6.7. Semi-discrete energy conservation

The thermodynamic discretization starts with the expansion of the internal energy in the basis $\{\phi_j\}$:

$$\varepsilon(x,t) \approx \sum_j \varepsilon_j(t)\phi_j(\mathbf{x},t).$$

Consider a weak formulation of the internal energy conservation equation (2.38) obtained by multiplying it by ϕ_i and integrating over the domain $w(t)$:

$$\int_{\omega(t)} \left(\rho \frac{d\varepsilon}{dt}\right) \phi_i = \int_{\omega(t)} (\mathbb{T} : \nabla \mathbf{u})\, \phi_i. \tag{6.13}$$

Expressing the energy in the moving thermodynamic basis gives:

$$\sum_j \frac{d\varepsilon_j}{dt} \int_{\omega(t)} \rho \phi_j \phi_i = \int_{\omega(t)} (\mathbb{T} : \nabla \mathbf{u})\, \phi_i.$$

In other words,

$$\mathbf{M}_{\mathcal{E}} \frac{d\varepsilon}{dt} = \int_{\omega(t)} (\mathbb{T} : \nabla \mathbf{u})\, \boldsymbol{\phi}, \tag{6.14}$$

where $\mathbf{M}_{\mathcal{E}}$ is the *thermodynamic mass matrix* which is defined by the integral

$$\mathbf{M}_{\mathcal{E}} \equiv \int_{\omega(t)} \rho \boldsymbol{\phi} \boldsymbol{\phi}^{\mathrm{tr}}. \tag{6.15}$$

As with the kinematic mass matrix, we can consider the thermodynamic mass matrix as being assembled from the degrees of freedom associated with each individual cell c as $\mathbf{M}_{\mathcal{E}} = Assemble(\mathbf{M}_{\mathcal{E},c})$. However, due to the discontinuous nature of the thermodynamic basis, there is no sharing of degrees of freedom across cell boundaries and so the "assembled" thermodynamic mass matrix is *block diagonal* where each block is a symmetric positive definite matrix of dimension N_e by N_e, where N_e denotes the number of coefficients used in the basis function expansion for the internal energy.

Analogous to the kinematic case, we can use the fact that the thermodynamic basis functions have zero material derivatives to conclude that $\mathbf{M}_{\mathcal{E}}$ is independent of time.

6.8. The force matrix

We now introduce an $N_{\mathcal{U}} \times N_{\mathcal{E}}$ rectangular matrix $\mathbf{F}$, which we call the *force matrix* that connects the kinematic and thermodynamic spaces:

$$\mathbf{F}_{ij} = \int_{\omega(t)} (\mathbb{T} : \nabla W_i)\, \phi_j\,. \tag{6.16}$$

As with the previously defined matrices, the matrix $\mathbf{F}$ from (6.16) is assembled from individual cell contributions:

$$\mathbf{F} = Assemble(\mathbf{F}_c)$$

This local rectangular matrix is the high-order generalization of the "corner force" concept described in [48]. It represents the hydrodynamic force contributions from a given cell to a given shared kinematic degree of freedom as well as the work done by the velocity gradient in the energy equation.

Evaluating $\mathbf{F}_c$ is a locally FLOP-intensive calculation that forms the computational kernel of this finite element discretization method. Specifically, transforming each cell back to the reference element and applying quadrature yields:

$$(\mathbf{F}_c)_{ij} \approx \sum_n \alpha_n \hat{\mathbb{T}}(\hat{q}_k) : \mathbb{F}_c^{-1}(\hat{q}_k) \hat{\nabla} \hat{W}_i(\hat{q}_k) \hat{\phi}_j(\hat{q}_k) J_c(\hat{q}_k) \,. \tag{6.17}$$

Note that in general, the total stress $\mathbb{T}$, including the pressure via an EOS call and any artificial viscous stress for shock capturing, is *evaluated at each quadrature point* in (6.17). Furthermore, the density (in an EOS call for example) is *evaluated at each quadrature point* using the strong mass conservation principle of (6.5). The notion of sampling the density and pressure as functions evaluated at zone quadrature points is a key component of our high-order discretization approach and is essential for robust behavior. Indeed, it is the reason we do not require any special hourglass filters and is directly analogous to the sub-zonal pressure method of [50].

6.9. The Euler equations in semi-discrete form

Given the previous definitions, we can summarize the general semi-discrete Lagrangian conservation laws in the following simple form:

$$\text{Momentum Conservation:} \qquad \mathbf{M}_{\mathcal{U}} \frac{d\mathbf{u}}{dt} = -\mathbf{F} \cdot \mathbf{1}, \tag{6.18}$$

$$\text{Energy Conservation:} \qquad \mathbf{M}_{\mathcal{E}} \frac{d\varepsilon}{dt} = \mathbf{F}^{\mathrm{tr}} \cdot \mathbf{u}, \tag{6.19}$$

$$\text{Equation of Motion:} \qquad \frac{d\mathbf{x}}{dt} = \mathbf{u}. \tag{6.20}$$

The vector $\mathbf{1}$ above is the representation of the constant one in the thermodynamic basis $\{\phi_i\}$ (we assume that $1 \in \mathcal{E}$). Boundary conditions can be implemented by eliminating the corresponding rows and columns in $\mathbf{M}_{\mathcal{U}}$ (which will in general introduce a difference between its blocks).

6.10. Time integration and the fully-discrete approximation

In this section we consider a general high-order temporal discretization method, and demonstrate its impact on the semi-discrete conservation laws. Specifically, let $t \in \{t_n\}_{n=0}^{N_t}$, and associate with each moment in time, t_n, the computational domain $D^n \equiv D(t_n)$. We identify the quantities of interest defined on D^n with a superscript n.

Let $Y = (\mathbf{u}; \varepsilon; \mathbf{x})$ be the hydrodynamic state vector. Then the semi-discrete conservation equations of (6.18)–(6.20) can be written in the form:

$$\frac{\mathrm{d}Y}{\mathrm{d}t} = \mathcal{F}(Y,t)\,, \qquad \text{where} \qquad \mathcal{F}(Y,t) = \begin{pmatrix} -\mathbf{M}_{\mathcal{U}}^{-1}\mathbf{F} \cdot \mathbf{1} \\ \mathbf{M}_{\mathcal{E}}^{-1}\mathbf{F}^{\mathrm{tr}} \cdot \mathbf{u} \\ \mathbf{u} \end{pmatrix}.$$

Standard high-order time integration techniques (e.g. explicit Runge–Kutta methods) can be applied to this system of nonlinear ODEs. However, these standard methods may need modifications to ensure numerical stability of the scheme and to ensure exact energy conservation. An example of this is given in the next section. Note that in general, a time integration method of order N will require N evaluations of the function $\mathcal{F}(Y,t)$.

6.11. The RK2-average scheme

The midpoint Runge–Kutta second order scheme reads

$$Y^{n+\frac{1}{2}} = Y^n + \frac{\Delta t}{2}\mathcal{F}(Y^n, t^n)\,, \qquad Y^{n+1} = Y^n + \Delta t\,\mathcal{F}(Y^{n+\frac{1}{2}}, t^{n+\frac{1}{2}}).$$

In practice, it was observed that the above scheme may be unstable even for simple test problems. Therefore, a modification of the scheme has been developed, which improves its stability and to ensure total energy conservation. Its two stages are given by

$$\begin{aligned}
\mathbf{u}^{n+\frac{1}{2}} &= \mathbf{u}^n - (\Delta t/2)\,\mathbf{M}_{\mathcal{U}}^{-1}\mathbf{F}^n \cdot \mathbf{1} & \mathbf{u}^{n+1} &= \mathbf{u}^n - \Delta t\,\mathbf{M}_{\mathcal{U}}^{-1}\mathbf{F}^{n+\frac{1}{2}} \cdot \mathbf{1} \\
\varepsilon^{n+\frac{1}{2}} &= \varepsilon^n + (\Delta t/2)\,\mathbf{M}_{\mathcal{E}}^{-1}(\mathbf{F}^n)^{\mathrm{tr}} \cdot \mathbf{u}^{n+\frac{1}{2}} & \varepsilon^{n+1} &= \varepsilon^n + \Delta t\,\mathbf{M}_{\mathcal{E}}^{-1}(\mathbf{F}^{n+\frac{1}{2}})^{\mathrm{tr}} \cdot \bar{\mathbf{u}}^{n+\frac{1}{2}} \\
\mathbf{x}^{n+\frac{1}{2}} &= \mathbf{x}^n + (\Delta t/2)\,\mathbf{u}^{n+\frac{1}{2}} & \mathbf{x}^{n+1} &= \mathbf{x}^n + \Delta t\,\bar{\mathbf{u}}^{n+\frac{1}{2}},
\end{aligned}$$

where $\mathbf{F}^k = \mathbf{F}(Y^k)$ and $\bar{\mathbf{u}}^{n+\frac{1}{2}} = (\mathbf{u}^n + \mathbf{u}^{n+1})/2$.

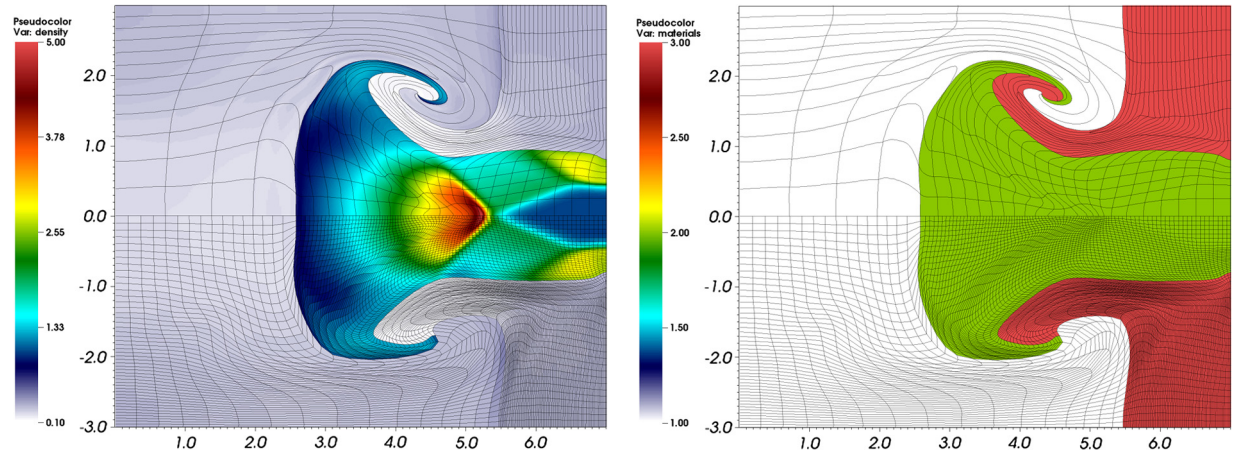

Fig. 11. Comparisons of high-order Q_4–Q_3 (top-half) and low-order Q_1–Q_0 (bottom-half) finite element SGH calculations for the shock triple-point benchmark densities (left) and materials (right). Note the extreme mesh curvature, sub-cell resolution and better resolved vortical motion of the materials for the high-order Q_4–Q_3 result.

The RK2-Average scheme described above conserves the discrete total energy exactly: The change in kinetic and internal energy can be expressed as

$$\mathcal{K}^{n+1} - \mathcal{K}^n = (\mathbf{u}^{n+1} - \mathbf{u}^n) \cdot \mathbf{M}_{\mathcal{U}} \cdot \bar{\mathbf{u}}^{n+\frac{1}{2}} = -\Delta t\, (\mathbf{F}^{n+\frac{1}{2}} \cdot \mathbf{1}) \cdot \bar{\mathbf{u}}^{n+\frac{1}{2}}$$
$$\mathcal{E}^{n+1} - \mathcal{E}^n = \mathbf{1} \cdot \mathbf{M}_{\mathcal{E}} \cdot (\varepsilon^{n+1} - \varepsilon^n) = \Delta t\, \mathbf{1} \cdot (\mathbf{F}^{n+\frac{1}{2}})^{\mathrm{tr}} \cdot \bar{\mathbf{u}}^{n+\frac{1}{2}})\,.$$

Therefore the discrete total energy is preserved: $\mathcal{K}^{n+1} + \mathcal{E}^{n+1} = \mathcal{K}^n + \mathcal{E}^n$.

6.12. High-order finite element closure model

Application of a closure model to the high-order finite element Lagrangian formulation begins with a continuum ODE treatment of the material volume fractions. Specifically, we define for each material k a so called "indicator function" η_k which has the property $|\omega_{c,k}| = \int_{\omega_c} \eta_k$. The indicator function can be viewed as a point-wise volume fraction function. We augment the semi-discrete Euler equations with an additional evolution equation for the indicator functions, namely $\frac{d\eta_k}{dt} = \kappa_k$. The special case of equal volumetric strain corresponds to the case $\kappa_k = 0$ (*i.e.* the volume fractions are treated in a pure Lagrangian manner). Use of Tipton like methods can be applied to define a more physically motivated closure model based on a pressure relaxation process which is described in detail in [74]. The main difference for the high-order finite element case is that the closure model is evaluated in a semi-discrete way at every element quadrature point, making it valid for arbitrarily high-order space/time finite element formulations.

6.13. A high-order finite element Lagrangian example

To illustrate the key features of the generalized high-order finite element SGH approach, we consider the classic shock triple-point benchmark which consists of a three state, two material, 2D Riemann problem which generates vorticity. The problem domain is $[0, 7] \times [0, 3]$ and is split into three separate regions. Each region has an ideal gas equation of state. This problem tests the ability of a code to propagate shock waves over multi-material regions and handle complex mesh motion due to vorticity.

We consider two specific choices for the orders of the finite element kinematic and thermodynamic spaces: a low order case denoted Q_1–Q_0, meaning bi-linear kinematic basis functions and piece-wise constant thermodynamic basis functions (4 kinematic and 1 thermodynamic unknowns per cell) and a high order case denoted Q_4–Q_3, meaning bi-quartic kinematic basis functions and bi-cubic thermodynamic basis functions (25 kinematic and 16 thermodynamic unknowns per cell). In each case, we keep the total number of kinematic and thermodynamic unknowns fixed, which means we run the Q_4–Q_3 method on a mesh two levels coarser than the mesh used for the Q_1–Q_0 case. The results are shown in Fig. 11, where the high-order Q_4–Q_3 results are shown in the top half while the low-order Q_1–Q_0 results are shown in the bottom half. The high-order Q_4–Q_3 generates sub-cell resolution of the shock waves and high mesh curvature. In each case, use of the general finite element approach (including the global mass matrix solve) results in a robust Lagrangian calculation which runs to a final time of $t = 5.0$; however, for a given resolution, the high-order Q_4–Q_3 is better at resolving at the vortical motion.

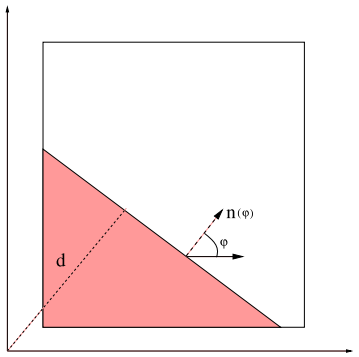

Fig. 12. Intersection of the multimaterial cell with half-plane.

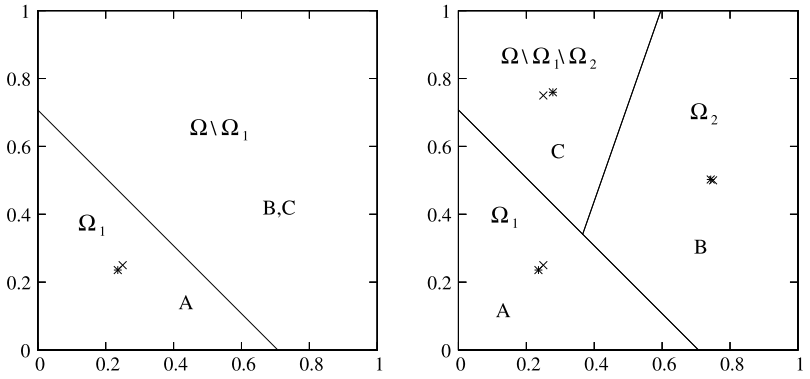

Fig. 13. Nested dissection.

7. Interface reconstruction

As well as at the initialization of the problem interface reconstruction is needed after each Lagrangian step, because the mesh has changed as well as the material data and therefore the interfaces will have changed; and also on the rezoned mesh as part of the remapping process from the Lagrangian mesh to the rezoned mesh. In each case, the input data for the interface reconstruction (IR) process consists of volumes of the materials (or equivalently volume fractions) in the multimaterial cells (MMC). The more advanced methods also use information about the positions of the centroids of each material in the multimaterial cell. In this section we consider IR only for the case of the convex MMC in 2D.

The most common representation of the interface between two materials used by IR methods consists of a single linear interface (a line in 2D and a plane in 3D). This class of interface representation is commonly called Piecewise-Linear Interface Calculation (PLIC), [156]. In this paper we will only consider such a representation of the interface.

The result of the interface reconstruction is a set of pure subcells of each material which cover the MMC without gaps or overlaps. Each subcell is a convex polygon whose volume is equal to a specified volume.

In modern IR methods these polygons are obtained as a result of a nested dissection process (ND), [81,5]. Let us assume that order in which material will be cut from MMC is specified – in the simplest case this order is specified by the user; in advanced methods this order is determined automatically as part of the IR process. The first step in the ND process is to separate out the material which is first in order from the conglomerate of the other materials. This is done by intersecting a half-plane with MMC. The half-plane is defined by a normal $\mathbf{n}$ and a signed distance, d, from this plane to the origin of coordinate system in the direction of the normal. The normal is defined by one parameter: the angle φ between the normal and the x axis. Therefore, half-plane is defined by two parameters, φ and d, Fig. 12. Let us assume that the angle φ is given (we will describe different approaches for its definition later in this section). Then one needs to find a distance d such that the intersection of the half-plane with the MMC will have the specified volume. The volume of the intersection is a monotone function of d and there are many methods to solve the equation for d, [156,80].

Let us consider the case of three materials A, B, C in the cell Ω and the order of the materials are such that material A is the first material, material B is second, and material C is third. The result of cutting material A from the cell Ω is a convex polygon Ω_1 which will represent material A. The polygon $\Omega \setminus \Omega_1$ represents the conglomerate of materials B and C – left panel in Fig. 13. Next we cut material B from the polygon $\Omega \setminus \Omega_1$ to form polygon Ω_2 which represent material B – right panel in Fig. 13. The polygon $\Omega \setminus \Omega_1 \setminus \Omega_2$ represent material C.

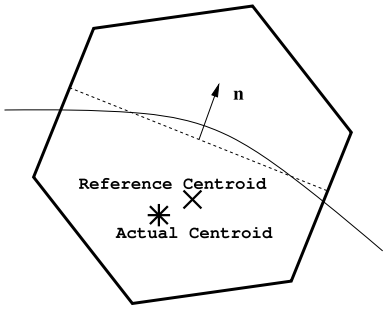

Fig. 14. Moment of fluid – curved line is actual interface, dashed line is reconstructed interface.

7.1. Definition of the normal for case of two materials

We will now discuss, three representative PLIC methods for the two material case: the gradient based method, the least squares volume-of-fluid (VOF) interface reconstruction algorithm, and the moment-of-fluid (MOF) method (this presentation follows [5]).

PLIC methods for the two material case differ in how the normal $\mathbf{n}$ is computed. Most of the standard PLIC methods, [156,155], only use information about the volume fractions, α_c^{ref} in the MMC cell c under consideration and the volume fractions, $\alpha_{c'}^{ref}$ in the neighboring cells $c' \in \mathcal{C}(c)$.

7.1.1. Standard PLIC methods

In this paper, standard PLIC methods are represented by two methods. The first method is an extension of the central difference algorithm developed by Parker and Youngs as described in [155]. This method uses an estimate of the gradient of the volume fraction field to compute a normal for the interface – we will refer to it as the GRAD method. The second method is the least squares volume-of-fluid interface reconstruction algorithm – LVIRA, [155]. LVIRA attempts to find a planar interface in the MMS, which, when extended into neighboring cells, gives the best (in a least squares sense) match to the reference volume fractions in these cells with the constraint that this planar interface must exactly reproduces the volume fraction in the cell under consideration. Both the GRAD and LVIRA methods only use information about the volume fractions.

In the GRAD method, the interface normal, $\mathbf{n}$, is computed by approximating the gradient of the volume fraction function α as $\mathbf{n} \approx -(\partial\alpha/\partial x, \partial\alpha/\partial y)$. Estimates of the derivatives can be obtained using Green theorem or a least squares approach. The GRAD method in general does not reproduce linear interfaces.

In the LVIRA method, [155], the interface normal is computed by minimizing the following error function:

$$E_c^{LVIRA}(\mathbf{n}(\varphi)) = \sum_{c' \in \mathcal{C}(c)} \left(\alpha_{c'}^{ref} - \alpha_{c'}(\mathbf{n})\right)^2, \tag{7.1}$$

where $\alpha_{c'}^{ref}$ is the reference volume fraction of the neighboring cell c' and $\alpha_{c'}(\mathbf{n})$ is the actual volume fraction in cell c' constructed by extending the interface of the central cell-c, under the constraint that the corresponding plane exactly reproduces the volume fraction in the cell under consideration. Like the GRAD method, LVIRA also requires information about the volume fractions from all the cells that are its immediate neighbors. In contrast to the GRAD method, LVIRA requires the minimization of a non-linear objective function, (7.1), with respect to one variable – angle, φ, which defines the normal. Each evaluation of the objective function for a given angle φ requires the determination of the distance d to match the volume fraction in the cell c as well as the intersection of corresponding half-plane with neighboring cells and the computing of corresponding volumes. The LVIRA method reproduces a linear interface exactly.

7.1.2. Moment of fluid (MOF) PLIC method

MOF was introduced in [80,81]. The MOF method uses information about the volume fractions, α_c^{ref} and the centroids, $\mathbf{x}_c^{ref}$ of the materials, but only from the cell c under consideration. No information from neighboring cells is used, as illustrated in Fig. 14. In the MoF method, the computed interface is chosen to match the reference volume fraction exactly and to provide the best possible approximation to the reference centroid of the material. In MOF, the interface normal, $\mathbf{n}$, is computed by minimizing the following functional (with the constraint that the corresponding pure subcell has to exactly match the reference volume fraction in the cell):

$$E_c^{MOF}(\mathbf{n}) = \|\mathbf{x}_c^{ref} - \mathbf{x}_c^{act}(\mathbf{n})\|^2,$$

where $\mathbf{x}_c^{ref}$ is the reference material centroid and $\mathbf{x}_c^{act}(n)$ is the actual material centroid (centroid of the reconstructed polygon) with the given interface normal $\mathbf{n}$. Similar to LVIRA, the implementation of the MoF method requires the minimization

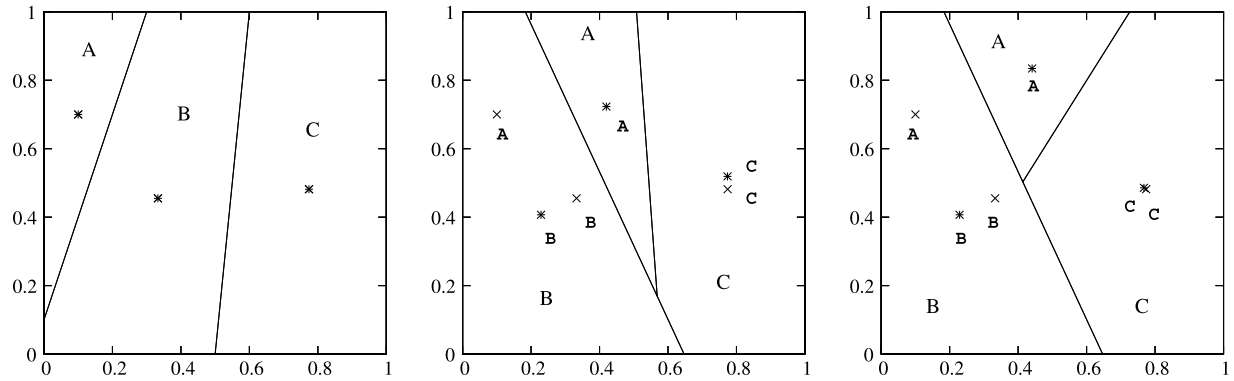

Fig. 15. Importance of the material ordering – three material case: left panel – original material configuration, which is exactly reproduced for the material ordering A, B, C – actual centroids (marked by x) coincide with actual centroids (marked by $*$); middle panel – ordering B, A, C; right panel – ordering B, C, A.

of the non-linear function of one variable; It is shown in [80] that the vector from the reference centroid to the centroid of the entire cell is a good initial guess for the interface normal. The computation of $E_c^{MOF}(\mathbf{n})$ for given $\mathbf{n}$ requires the following steps to be performed. The first step is to find the parameter d such that the volume fraction in cell c exactly matches α_c^{ref}, this step is also performed for the LVIRA algorithm. Secondly, we compute the centroid of the resulting polygon. Finally, the distance between the actual and reference centroids is computed. The MoF method is less expensive than LVIRA because it does not require the computation of any terms relating to neighboring cells and because the computation of the centroids is relatively inexpensive. The MOF method is linearity-preserving.

7.2. Multimaterial case

As we have mentioned before the ND approach requires the material ordering to be known. There have been several attempts to develop automatic-local schemes for material ordering for the standard PLIC methods – see references in [5]. All of these use information about approximate positions of the centroids for the materials in the multimaterial cell and its neighbors. To the best of our knowledge, none of these methods even treat all basic cases correctly. In the MOF method, the positions of the material centroids inside the multimaterial cell are not only used for each "two" material interface reconstruction step, but also in choosing the best material ordering that minimizes the discrepancy between the given (reference) and actual centroids computed for the reconstructed pure polygons for all the materials [81,5].

Let us consider the three material example presented in Fig. 15. The left panel in Fig. 15 shows the "exact" material polygons from which the reference volume fractions and the centroids were computed. In the case of the ABC material ordering MOF reconstructs these polygons exactly and the total discrepancy between the actual and the reference centroids is zero. The central panel in Fig. 15 shows the result of the interface reconstruction for the BAC material ordering – one can see that the overall discrepancy between the reference and the actual centroids is big, the reconstructed material polygons are far from "exact" and even the topology is wrong, for example, material A is between materials B and C. A similar situation is obtained for material ordering BCA – right panel in Fig. 15. For the MOF method one needs to try all possible orderings and then choose one which minimizes the overall discrepancy between the reference and the actual centroids. Clearly, a user may use his/her knowledge about the overall topology of the interfaces to reduce number of permutations. For example, for three materials the number of permutations is six, however, if it is known that it is a layered structure like the one presented in the left panel in Fig. 15, then any ordering that starts with material A or C will produce the correct result, and therefore, one only needs to try three cases to determine the material ordering. It is also important that the material ordering is local for each cell.

To finish this section we present results of interface reconstruction for the shapes presented in Fig. 16 – courtesy of D. Bailey (LLNL). In Fig. 17 we present the result of the interface reconstruction on a polygonal mesh for the entire domain and zoom in on two fragments.

Comparisons of the performance of different interface reconstruction methods can be found in [112].

8. Rezoning

8.1. Untangling

We assume that after the Lagrangian step the mesh is valid, but it may still have some slightly non-convex cells. Most mesh improvement algorithms are iterative in nature and require that the initial mesh at the start of this iterative process consist of convex cells. The goal of *untangling* is to locally modify the mesh to ensure that all the cells are convex – [27,173].

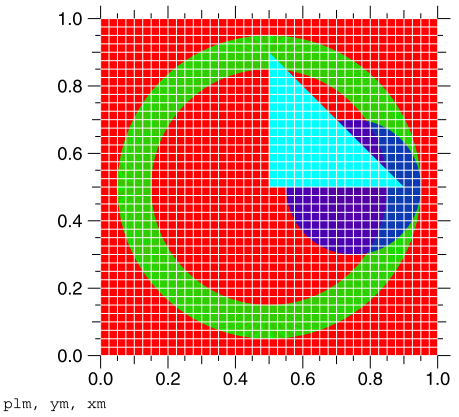

Fig. 16. Geometric shapes – Bailey's example.

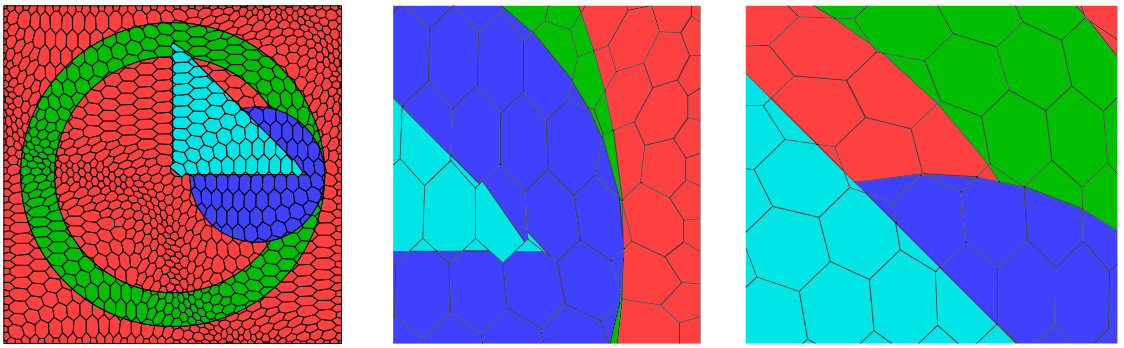

Fig. 17. Interface reconstruction for Bailey's example on polygonal mesh.

The requirement is that the mesh is only changed in the neighborhood of non-convex cells, that is, in general, it must stay close to the Lagrangian mesh. Here we describe the so-called "feasible set" approach for untangling.

In this approach we consider patches of cells connected to some point in the mesh in some order. For simplicity, let us consider the patch of four quadrilateral cells: ABED, BCFE, DEHG and EFIH with a central point E, Fig. 18. For each cell to be convex, each triangle in the pair of triangles in to which it can be subdivided has to be not inverted. For example, for cell ABED those triangles are ABD, BED and ABE, AED. If all the cells connected to point E are convex then we proceed to next point. If at least one cell connected to point E is non-convex (for example, cell ABED in the left panel in Fig. 18, then we do the following. We assume that all the points with the exception of point E are fixed at this stage. We define the *feasible set* for point E as a locus of its possible positions such that all the triangles connected to this point are valid. In this case it is the intersection of twelve half-planes – middle panel in Fig. 18. Then we move point E to the centroid of the feasible set – as in the right panel of Fig. 18. In Fig. 18 we demonstrate the untangling process. In general, it maybe possible that the feasible set is empty, then we proceed to the next point. It may take several loops over the entire mesh to untangle the mesh. In practice, tangling is local and the process described above works very well, [27]. In Fig. 19 we give an example of untangling for the so-called *Shestakov* mesh.

8.2. Reference Jacobian rezoning

For simplicity let us consider a logically rectangular grid in physical space that results from a map of a uniform grid to a unit square of logical space. The most fundamental object describing the map is its Jacobian matrix. Therefore, one can expect that if two maps have similar Jacobian matrices, then the maps themselves are similar and so will produce grids that are close to each other. This leads to the following strategy. We first construct Jacobian matrices termed the reference Jacobian matrices (RJMs). The formation of the RJMs is supposed to incorporate the desired local properties of the mesh (this could be, for example, mesh quality or adaptation). It is not possible to use the RJMs defined at each vertex by themselves to derive the new (*i.e.*, rezoned) positions of the vertices. To begin with, each RJM related to the cell specifies the vectors that form the cell edges, and there is no guarantee that these vectors will form a closed figure. Furthermore, the cells do not exist in isolation from each other. Each vertex (not on a boundary) belongs to four cells; a simple rezone strategy applied individually to each cell will lead, in general, to four incompatible specifications of the rezoned position of any vertex. The solution to this problem is to construct a global functional that measures the difference between the RJM (which depends on the Lagrangian grid and the smoothing process) and the rezoned Jacobian (not known before the

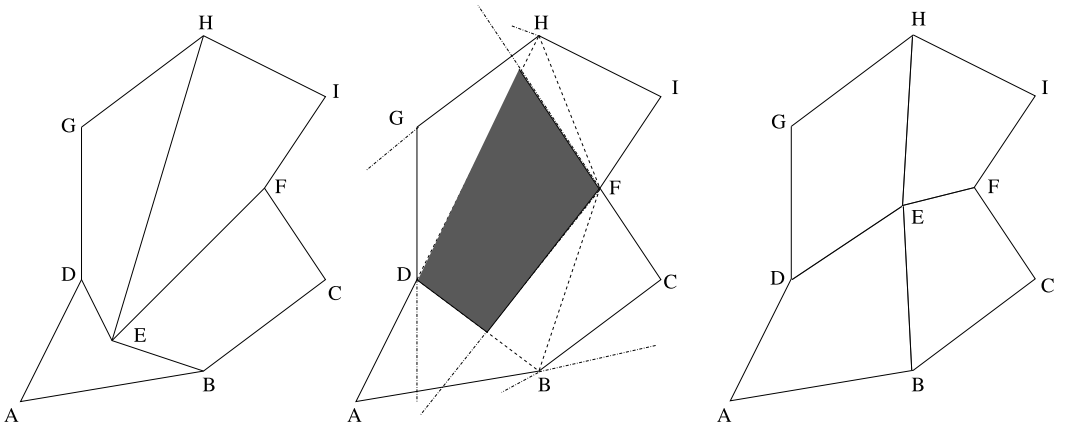

Fig. 18. Untangling process. Left panel – cell ABED is non-convex; central panel – construction of feasible set (shaded area) by intersection of half-planes, new position of the point E in the centroid of feasible set – all cells are convex.

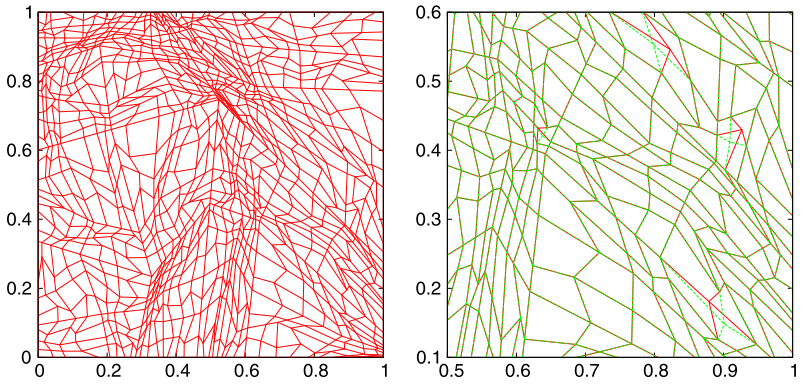

Fig. 19. Untangling. Left panel – mesh with several tangled cells; Right panel – fragment of the superimposed original and untangled mesh (green lines). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

minimization process) for all cells in the grid. Minimizing this functional over the entire grid (as a function of each of the vertex coordinates) then leads to the rezoned grid. The global functional includes a barrier, which is the determinant of the Jacobian matrix. It effectively penalizes any solution where the area of the cell is much smaller than the area derived from the RJM. If we start the optimization process with a valid mesh that has been obtained by untangling the Lagrangian mesh, then the presence of the barrier guarantees that the resulting rezoned mesh will be valid and contain only convex cells.

Let us give some details. Consider a cell of a general polygonal mesh Fig. 1. In this case the Jacobian matrix can be connected to the corner pc of the cell – $\mathbb{J}_{pc}$:

$$\mathbb{J}_{pc} = \left(\mathbf{e}^c_{p^+p} | \mathbf{e}^c_{p^+p}\right), \quad \mathbf{e}^c_{p^+p} = \mathbf{x}_{p^+(c)} - \mathbf{x}_{p(c)}, \quad \mathbf{e}^c_{p^-p} = \mathbf{x}_{p^-(c)} - \mathbf{x}_{p(c)}, \tag{8.1}$$

where the vectors $\mathbf{e}^c_{p^+p}, \mathbf{e}^c_{p^-p}$ are pointing from the vertex p to vertices p^+ and p^- and form columns of the Jacobian matrix.

Let us assume that we have formed the RJM matrix $\mathbb{J}^{ref}_{pc}$, which locally incorporates the desired properties of the mesh, and $|\mathbb{J}^{ref}_{pc}| > 0$, then the global optimization problem is to minimize the following functional with respect to the coordinates of the mesh points

$$F = \sum_p \left(\sum_{c \in \mathcal{C}(p)} \frac{|\mathbb{J}^{ref}_{pc}|}{|\mathbb{J}_{pc}|} \cdot \frac{\|\mathbb{J}_{pc} - \mathbb{J}^{ref}_{pc}\|^2}{\|\mathbb{J}^{ref}_{pc}\|^2} \right), \tag{8.2}$$

the $|\mathbb{J}_{pc}|$ in the denominator plays the role of the barrier, because it is the area of the triangle based on vectors $\mathbf{e}^c_{p^+p}, \mathbf{e}^c_{p^-p}$ and it goes to zero if the corresponding corner gets close to 0 or 2π.

Let us give one example of forming the RJM, [107]. The essential idea in this paper is the recognition that the Lagrangian solution before rezoning contains sufficient information about the flow to constrain our measure of the smoothness of the mesh. More specifically, the Lagrangian mesh reflects both the physical motion of the fluid and the non-physical distortion.

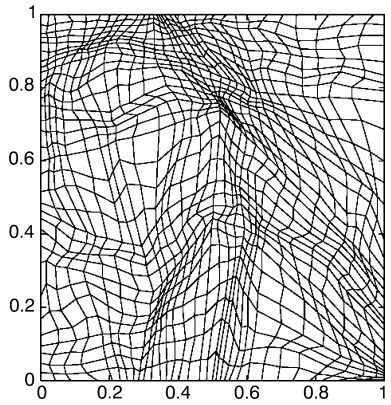

Fig. 20. Rezoned mesh for "Lagrangian" mesh in left panel of Fig. 19.

We assume that the non-physical distortion of a computational grid has a much shorter wavelength and so can be separated from the physical motion by averaging over a small neighborhood of the cell. This assumption permits the rezoned grid to remain close to the Lagrangian (or mesh after untangling) grid but be "smoother" (*i.e.*, have better geometrical quality). At each node we form a local patch from the adjacent cells of the Lagrangian grid and construct a local realization of the Winslow smoothness functional ([180,107]), on this patch:

$$F_p^{sm} = \sum_{c \in \mathcal{C}(p)} \frac{|\mathbf{e}^c_{p^+p}|^2 + |\mathbf{e}^c_{p^-p}|^2}{|\mathbf{e}^c_{p^+p} \times \mathbf{e}^c_{p^-p}|}. \tag{8.3}$$

Minimization of this functional with respect to the position of the central node defines its "virtual" location (the node is not actually moved at this stage). By connecting this virtually moved node to its (stationary) neighbors, we define a reference Jacobian for each corner related to point p that represents the best locally achievable geometric grid quality. If the locally Lagrangian mesh is already smooth, then the resulting RJMs are very close to one which correspond to a Lagrangian mesh. Because in this local optimization functional all points with the exception of the central point are in their Lagrangian positions the resulting RJMs have "memory" of the Lagrangian mesh. In Fig. 20 we show the rezoned mesh obtained from the mesh shown in the left panel in Fig. 19 ("Lagrangian" mesh). One can see that the rezoned mesh preserves the "long-waves" features of the Lagrangian mesh but is much smoother.

Finally, we note we note that the global minimization problem belongs to the class of so-called nonlinear least squares, [151] for which effective minimization procedures have been developed – see some example in [107,79].

In some cases it is possible to keep the cells pure by imposing the constraint that the vertices of the mesh move along the interface. It leads to a constrained optimization problem, [168].

8.3. High-order linear mesh relaxation

Now let us describe some rezone strategies used in ALE methods, which use the high-order finite element SGH discretization for the Lagrangian stage.

We begin with a vector of (high-order) mesh position values decomposed into interior and boundary sets such that

$$\mathbf{x} = (\mathbf{x}_I, \mathbf{x}_B).$$

Now define a "mesh Laplacian" operator $\mathbf{L} = (\mathbf{L}_{II}, \mathbf{L}_{IB})$. The matrix $\mathbf{L}$ should be a *topological* operator (independent of the geometry $\mathbf{x}$), with

$$\sum_j \mathbf{L}_{ij} = 0.$$

Then

$$\mathbf{x}_I^{n+1} = \mathbf{x}_I^n + (\mathbf{f}_I - \mathbf{L}_{II}\mathbf{x}_I^n)$$

where $\mathbf{f}_I = -\mathbf{L}_{IB}\mathbf{x}_B^0$ and $\mathbf{x}_B^n = \mathbf{x}_B^0$. This is a simple linear iteration, which in limit converges to the harmonic extension of the boundary nodes to the interior.

More generally, we consider a class of linear harmonic relaxation schemes:

$$\mathbf{x}^{n+1} = \mathbf{x}^n + \mathbf{M}^{-1}(\mathbf{f} - \mathbf{L}\mathbf{x}^n) \tag{8.4}$$

where

- $\mathbf{x}^n$ are the nodes of the high-order mesh after n relaxation steps.
- $\mathbf{L}$ is a "mesh Laplacian" extended with $\mathbf{L}_{BI} = 0$, $\mathbf{L}_{BB} = I$, and $\mathbf{f} = (-\mathbf{L}_{IB}\mathbf{x}_B^0, \mathbf{x}_B^0)$.
- $\mathbf{M}$ is a smoother (preconditioner) for $\mathbf{L}$.

A particular algorithm is specified by the choice of $\mathbf{L}$, $\mathbf{M}$ and the number of smoothing steps. The mesh Laplacian $\mathbf{L}$ can be defined in different ways, including:

- Using the FEM sparsity to connect the high-order nodes with *equal weights*.
- Assembling the high-order stiffness matrix on the *reference element*.
- $\mathbf{L} = \mathbf{G}^t\mathbf{G}$, where $\mathbf{G}$ is the *"discrete gradient"* matrix between the high-order nodal H^1 and Nedelec $H(curl)$ spaces.

There are also several choices for the smoothing matrix $\mathbf{M}$, for example:

- Simple diagonal scaling

$$\mathbf{M}_{ii} = \sum_j |\mathbf{L}_{ij}| \,.$$

- Custom low-frequency preserving *polynomial smoothers*, e.g. following Berndt and Carlson's work on polynomial filtering [26].

8.4. High-order non-linear mesh relaxation

More generally, we consider an integral minimization process applied to a so called "mesh energy function" of the form

$$E(\mathbf{x}) = \sum_c \int_{\hat{\omega}_c} B(\hat{\mathbb{F}}_c(\hat{\mathbf{x}})),$$

for some functional B. In this form, harmonic optimization simply corresponds to the choice

$$B(\hat{\mathbb{F}}_c) \equiv \frac{1}{2}(\hat{\mathbb{F}}_c : \hat{\mathbb{F}}_c) = \frac{1}{2}\operatorname{tr}(\mathbb{F}_c^T\mathbb{F}_c).$$

The inverse harmonic approach of Winslow minimizes the gradient norm of the *inverse* map $\hat{\mathbf{\Phi}}_c^{-1}$:

$$E(\mathbf{x}) = \frac{1}{2}\sum_c \int_{\omega_c} \mathbb{F}_c^{-1} : \mathbb{F}_c^{-1}.$$

Using a change of variables, we can write each cell integral on the reference cell as

$$\int_{\omega_c} (\mathbb{F}_c^{-1} : \mathbb{F}_c^{-1}) = \int_{\hat{\omega}_c} (\mathbb{F}_c^{-1} : \mathbb{F}_c^{-1}) J_c$$

Thus, in the general setting the inverse harmonic optimization method corresponds to the energy functional

$$B(\hat{\mathbb{F}}_c) \equiv \frac{1}{2} J_c \operatorname{tr}(\mathbb{F}_c^{-T}\mathbb{F}_c^{-1}) \,.$$

In contrast with the harmonic case, the stationary point equations are *nonlinear*.

Newton's method is then applied for the stationary point equation $\nabla E(\mathbf{x}) = 0$, namely

$$\mathbf{x}^{n+1} = \mathbf{x}^n - \left[\mathcal{H}E(\mathbf{x}^n)\right]^{-1} \nabla E(\mathbf{x}^n)$$

For a given $\mathbf{x}$, the energy function $E(\mathbf{x})$, the gradient vector $\nabla E(\mathbf{x})$ and the Hessian matrix $\mathcal{H}E(\mathbf{x})$ can be assembled element-by-element based on $B(\hat{\mathbb{F}}_c)$ and its derivatives.

9. Remapping stage

The remap is the final stage of the ALE method – it is used to transfer data from the Lagrangian mesh to the mesh obtained as result of the rezoning. This section is devoted to describing the basic techniques that are applied to provide a remap with the important properties of a successful remap method [144]: conservation of mass, momentum and total energy; at least second order accuracy; bound-preservation – after the remap all the parameters have to stay in physically justified bounds.

The CGH method only requires the remapping of cell-centered quantities, while the SGH method require the remapping of both cell-centered and nodal quantities, which in our case are the components of the velocity field. Remapping of the nodal quantities for the SGH method requires special attention and it is addressed in a special Section 9.2.

The remapping for HO-SGH requires special consideration and is described in Section 9.5.

We will denote the cells after the Lagrangian step by ω_c, where c is generic cell index. The cells after rezone stage will be denoted by $\tilde{\omega}_{\tilde{c}}$ – we have tilde, $\tilde{}$, over ω to emphasize that it is a cell of the rezoned mesh and tilde over c to emphasize that rezoned mesh cannot be related to the Lagrangian mesh and can have a different number of cells and a different connectivity. We will use the same convention for all other quantities on the Lagrangian and the rezoned meshes, for example, for the density we will have ρ_c and $\tilde{\rho}_{\tilde{c}}$. We assume that the Lagrangian and the rezoned meshes cover the same computational domain without gaps and overlaps $\cup_c \omega_c = \cup_{\tilde{c}} \omega_{\tilde{c}}$.

All remapping methods for cell-centered quantities start with a function reconstruction on the Lagrangian mesh – Section 9.1.3. The most typical example is the piece-wise linear reconstruction. The piece-wise linear reconstruction usually involves some limiting procedure, which limits the slope to keep the values of the reconstructed function in physically and geometrically justified bounds. Special attention has to be paid to the reconstruction and the limiting of vector quantities, a simple component by component approach can lead to non-physical results.

The most general case of remapping is when the Lagrangian and the rezone meshes are not related to each other, in this case an intersection (overlay, geometric) remap is required, which involves the intersection of the cells of the rezoned mesh, ω_c with cells of the Lagrangian mesh, $\tilde{\omega}_{\tilde{c}}$. If the Lagrangian and the rezone mesh both have the same connectivity but the displacement of the rezoned cell $\tilde{\omega}_c$ with respect to the Lagrangian cell ω_c is arbitrary then an intersection based remap is still required. If the two meshes have the same connectivity and the rezoned mesh is contained within the union of the Lagrangian mesh with the same index and its nearest neighbors, then a flux-based remap can be used. All these types of remap described in Section 9.1.4.

We want to emphasize again that whether the remap used is an intersection-based (geometric, overlay) or a flux-based (swept region, incremental, advective) it should <u>NOT</u> be considered as part of an operator split method.

We strongly believe that for multimaterial flows an interface reconstruction and an intersection-based remapping method needs to be used. However, there are other methods for meshes with same connectivity and relatively small displacements, which do not use interface reconstruction, employ flux-based remapping and so-called *steepening*, which we briefly describe in Section 9.4.

The remapping process usually starts with the density. In this process the mass is remapped first in a conservative way to define the mass of each material in the rezoned cell and then the average density of each material in this cell is defined as a ratio of mass to the volume of the material. Remapping of all other cell centered quantities has to be consistent with remap of mass, that is, for example, when remapping the internal energy it has to take the form of a mass contribution multiplied by some value of the specific internal energy. This requirement is especially important for multimaterial case, because if it is not satisfied then one may have a mass of some material in the multimaterial cell, but not have any internal energy. This is addressed in Section 9.1.5. Next, for SGH, one can remap total internal energy for each material and define specific internal energy as a ratio of the remapped internal energy and remapped mass. For CGH one directly remaps total energy and then a closure model is used to distribute the internal energy between materials.

The next step in the remapping process is to remap the velocity, which involves conservative remapping of the momentum. Then the remapped velocity is defined as a ratio of the remapped momentum and the remapped mass. This remapped velocity and remapped mass defines the *actual* kinetic energy in the cell after remap.

Finally, for SGH, a so-called *kinetic energy fix* is required to restore the conservation of total energy, because the kinetic energy computed from the remapped velocity and remapped mass is not conservative – Section 9.1.6.

9.1. Remap of cell-centered quantities

In this section we will consider the remap of the cell-centered quantities, which is common to both the SGH and CGH hydro methods.

9.1.1. Statement of the remapping – single material case

We will demonstrate the statement of the remapping problem on the example of remapping the density. We can always assume that there is an underlying function $\rho(\mathbf{x})$, which is defined throughout the problem domain. The only information that we are given about this function after Lagrangian step is its mean value in each of the cells on the Lagrangian grid:

$$\bar{\rho}_c = \left(\int_{\omega_c} \rho(\mathbf{x})\, dV \right) \Big/ |\omega_c| \,, \tag{9.1}$$

where the integral in the numerator of (9.1) is the cell mass:

$$m_c \equiv \int_{\omega_c} \rho(\mathbf{x})\, dV \,, \tag{9.2}$$

and therefore

$$\bar{\rho}_c = m_c / |\omega_c| \,.$$

The total problem mass is defined as

$$\mathcal{M} \equiv \int_{\Omega} \rho(\mathbf{x})\, dV = \sum_c \int_{\omega_c} \rho(\mathbf{x})\, dV = \sum_c m_c = \sum_c \bar{\rho}_i \, |\omega_c| \,. \tag{9.3}$$

The goal of the remapping process is to find an accurate approximation $\tilde{m}_{\tilde{c}}$ for the masses of the new cells

$$\tilde{m}_{\tilde{c}} \approx \tilde{m}^{ex}_{\tilde{c}} = \int_{\tilde{\omega}_{\tilde{c}}} \rho(\mathbf{x})\, dV \,. \tag{9.4}$$

We need to define what is meant by "accurate", since the underlying density field is not known in detail. For example, the standard accuracy requirement is linearity preservation – that is, if $\bar{\rho}_c$ are obtained from a global linear function then the remapping has to lead to exact masses which correspond to this linear function on rezoned mesh.

There are constraints that the new cell masses must also obey that follow from the positivity of the density function and from the global conservation of the total mass. In particular, the minimal requirement is that all $\tilde{m}_{\tilde{c}}$ are positive and

$$\sum_{\tilde{c}} \tilde{m}_{\tilde{c}} = \mathcal{M} \,, \tag{9.5}$$

which is the statement of global conservation.

The approximate mean values of the density in the new cells are then defined by

$$\bar{\tilde{\rho}}_{\tilde{c}} = \tilde{m}_{\tilde{c}} / |\tilde{\omega}_{\tilde{c}}| \,.$$

Usually, in addition to positivity of the density *local bound-preservation* is also required, which is a more general property which can also be applied to other quantities that are also being remapped. The bound-preservation property is defined as follows

$$\min_{c \in I(\tilde{c})} \bar{\rho}_c \le \bar{\tilde{\rho}}_{\tilde{c}} \le \max_{c \in I(\tilde{c})} \bar{\rho}_c \tag{9.6}$$

where $I(\tilde{c}) = c : \tilde{\omega}_{\tilde{c}} \cap \omega_c \neq \emptyset$ – that is a set of cells on the Lagrangian mesh with which the rezoned cell $\tilde{c}$ overlaps.

9.1.2. Statement of the remapping – multimaterial case

In the multimaterial case cells of the Lagrangian mesh may be pure but contain different materials or can be multimaterial and contain several different materials in one cell. The materials in the multimaterial cell can be represented by pure material polygons, which result from the interface reconstruction or can just be represented by volume fractions, and other material quantities. The first task in multimaterial remapping is to define which cells of the rezoned mesh will contain which materials and then the corresponding volumes and masses of different materials in the cells need to be defined. The total volume and total mass of each material must be conserved. The formal statement of the multimaterial remapping for density is similar to the one described in the previous section but per material. For example, the bound preservation can be defined in a similar way to the single material case, but taking into account only densities of the same material in the neighboring cells.

The formal definition of accuracy of the remap in the multimaterial case is not well defined for multimaterial cells and we will not discuss it here.

9.1.3. Reconstruction and limiting on the Lagrangian mesh

Independent of the remapping method one first need to reconstruct a function in the cells of the Lagrangian mesh. The reconstruction clearly depends on what primary degrees of freedom are available as a result of the Lagrangian step. In SGH and CGH it is the mean values.

Single material case

As a result of Lagrangian step we have mean values of the density $\bar{\rho}_c = m_c / |\omega_c|$.

The simplest reconstruction is the piece-wise constant reconstruction

$$\rho_c(\mathbf{x}) = \bar{\rho}_c \,, \mathbf{x} \in \omega_c \,, \tag{9.7}$$

however such a reconstruction is not enough for an accurate remap. All modern remapping methods use at least piece-wise linear reconstruction of the form

$$\rho_c(\mathbf{x}) = \bar{\rho}_c + (\mathbf{GRAD}\rho)_c \cdot \left(\mathbf{x} - \mathbf{x}^{centr}_c\right) \,, \quad \mathbf{x} \in \omega_c \,, \tag{9.8}$$

where $(\mathbf{GRAD}\rho)_c$ is an approximation of the gradient in the cell and $\mathbf{x}_c^{centr}$ is the centroid of the cell ω_c

$$\mathbf{x}_c^{centr} = \left(\int_{\omega_c} \mathbf{x}\, \rho\, dv \right) / |\omega_c| \,. \tag{9.9}$$

The important property of the form (9.8) is conservation

$$\int_{\omega_c} \rho_c(\mathbf{x})\, dv = \bar{\rho}_c\, |\omega_c| = m_c \,, \tag{9.10}$$

which follows from definition of the centroid. This is a necessary condition that the reconstruction must satisfy for the intersection-based remap to be conservative. The so-called "unlimited" discrete gradient can be computed in different ways. For general polygonal meshes several approaches are possible, e.g., the least squares method and the Green–Gauss procedure [146,4]. The minimal requirement for the discrete gradient is linearity preservation, that is, it has to be exact in the case when the mean values of the density are obtained from a global linear function. Sometimes these are also called a second-order accuracy preserving gradients.

This discrete gradient is called unlimited, because in the general case values of the corresponding linear function $\rho_c(\mathbf{x})$ inside the cell ω_c can be outside of the physically justified bounds, for example, the density can be negative. In general, given a mesh cell and its immediate neighbors the bounds are usually defined as the minimum and maximum values for the density from this set,

$$\rho_c^{\mathrm{MIN}} = \min_{c' \in C(c)} \rho_c \,, \quad \rho_c^{\mathrm{MAX}} = \max_{c' \in C(c)} \rho_c \,, \tag{9.11}$$

which we will consider as the bounds within which we want the values of the function $\rho_c(\mathbf{x})$ to lie.

To keep the values of the reconstructed function in bounds we use limiters to limit the discrete gradient. The limited gradient is constructed with the following form

$$(\mathbf{GRAD}\rho)_c^{lim} = \phi_c\, (\mathbf{GRAD}\rho)_c \,, \tag{9.12}$$

where $0 \le \phi_c \le 1$ is a limiter.

The limiter can then be defined as follows

$$\phi_c = \min\left(1,\, \left(\rho_c^{\mathrm{MAX}} - \bar{\rho}_c\right) \Big/ \left(\rho_c^{\max} - \bar{\rho}_c\right),\, \left(\rho_c^{\mathrm{MIN}} - \bar{\rho}_c\right) \Big/ \left(\rho_c^{\min} - \bar{\rho}_c\right)\right) , \tag{9.13}$$

where $\rho_c^{\min}$ and $\rho_c^{\max}$ are the minimum and maximum values of the reconstructed function with the *unlimited* gradient. It should immediately be noted that if the cell value itself is a minimum or maximum for the neighborhood, the limiter will set the value of the gradient in the cell to zero. These limiters were devised by Dukowicz and Kodis [78] for remap. If we only want to enforce positivity of the density, then we can set

$$\phi_c = \min\left(1,\, \left(\rho_c^{\mathrm{MIN}} - \bar{\rho}_c\right) \Big/ \left(\rho_c^{\min} - \bar{\rho}_c\right)\right) , \tag{9.14}$$

where $\rho_c^{\mathrm{MIN}} = 0$ or some small number. This limiter was first described in a paper by Rider and Margolin [157] and similar schemes have been extensively developed by Shu and an array of his collaborators [182].

The reconstruction and limiting of vectors will be described at the end of Section 5.9.1.

Multimaterial case

If materials in the multimaterial cells are represented by pure material polygons then the function reconstruction on Lagrangian mesh is similar to the single material case, but performed in each material polygon. If material polygons are not used to represent materials then there are several ad-hoc methods (not published in readily available form) available in which the function corresponding to each material in multimaterial cell is reconstructed over entire multimaterial cell – see some descriptions in [9].

9.1.4. Types of meshes and types of remap methods

We start this section by mentioning that all remap methods require the reconstructed piece-wise linear function to be integrated over polygons. This integration can always be performed exactly by reducing the area integral to a sum of line integrals over the edges of the polygon, where because the edges of the polygon are straight lines it eventually reduces to simple one-dimensional integrals.

General meshes

The most general case is where the Lagrangian mesh and the rezoned mesh have different connectivity and different number of cells. In the left panel of Fig. 21 we show the case where the Lagrangian mesh is a polygonal mesh and rezoned mesh is a logically rectangular orthogonal mesh. In this case a cell of the rezoned mesh can be represented as follows $\tilde{\omega}_{\tilde{c}} = \cup_c \left(\tilde{\omega}_{\tilde{c}} \cap \omega_c\right)$, and therefore $|\tilde{\omega}_{\tilde{c}}| = \sum_c \left|\tilde{\omega}_{\tilde{c}} \cap \omega_c\right|$. This representation naturally leads to the following expression for remapped mass

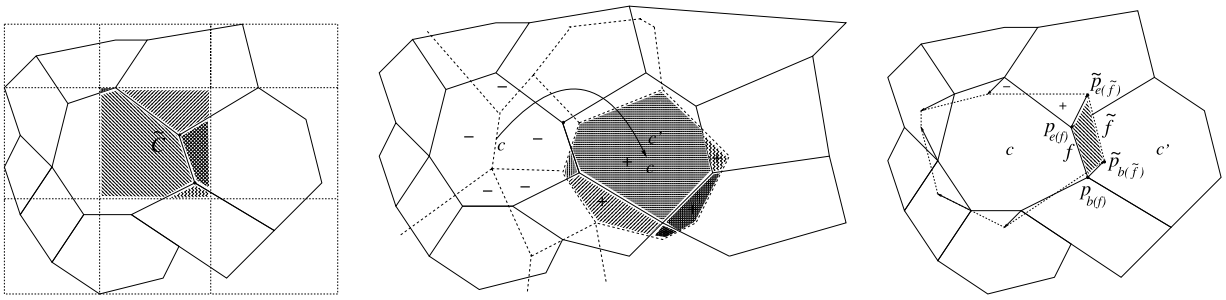

Fig. 21. Different types of the remap. Lagrangian mesh – solid lines, rezoned mesh – dashed Left panel: Intersection-based remap. Cell of rezoned mesh $\tilde{c}$ overlaps with five cells of the Lagrangian mesh – marked by different hatching; middle panel: Generalized intersection-flux-based remap; right panel: Swept region-based remap.

$$\tilde{m}_{\tilde{c}} = \sum_{c} \int\limits_{\tilde{\omega}_{\tilde{c}} \cap \omega_c} \rho_c(\mathbf{x})\, dV\,. \tag{9.15}$$

We will call remapping based on expressions like (9.15) by *direct* an intersection-based remap. In order for the direct intersection-based remap to be conservative the density reconstruction in the cells of the Lagrangian mesh must also be conservative. The proof of conservation is then trivial because the Lagrangian and rezoned mesh are composed of the same intersection pieces. The mean values of the remapped density are then given by

$$\bar{\tilde{\rho}}_{\tilde{c}} = \frac{\tilde{m}_{\tilde{c}}}{|\tilde{\omega}_{\tilde{c}}|} = \sum_{c} \left(\frac{\int_{\tilde{\omega}_{\tilde{c}} \cap \omega_c} \rho_c(\mathbf{x})\, dV}{|\tilde{\omega}_{\tilde{c}} \cap \omega_c|} \right) \cdot \frac{|\tilde{\omega}_{\tilde{c}} \cap \omega_c|}{|\tilde{\omega}_{\tilde{c}}|} = \sum_{c} \rho_{\tilde{c},c} \cdot \frac{|\tilde{\omega}_{\tilde{c}} \cap \omega_c|}{|\tilde{\omega}_{\tilde{c}}|}\,, \tag{9.16}$$

where according to the mean value theorem, [110], $\rho_{\tilde{c},c}$ is the value of $\rho_c(\mathbf{x})$ at some point in $\tilde{\omega}_{\tilde{c}} \cap \omega_c$. The expression (9.16) for the remapped density will be used later to analyze bound preservation – Section 9.3.

Meshes with the same connectivity and arbitrary displacement

The next case is when two meshes have the same connectivity and arbitrary displaced with respect to each other. Clearly in this case one can use the *direct* intersection-based remap described above. However, in this case one also can use the *generalized intersection-flux-based* remap. In the middle panel of Fig. 21 we present the case where cell c has moved outside its nearest neighbors. In this case the rezoned cell with index c can be represented as

$$\tilde{\omega}_c = \omega_c \bigcup \left(\bigcup_{c' \neq c} \tilde{\omega}_c \bigcap \omega_{c'} \right) \setminus \left(\bigcup_{c' \neq c} \omega_c \bigcap \tilde{\omega}_{c'} \right)\,,$$

which leads to the following expression for the remapped mass

$$\tilde{m}_c = m_c + \sum_{c' \neq c} \mathcal{F}^m_{c,c'}\,, \tag{9.17}$$

where

$$\mathcal{F}^m_{c,c'} = \int\limits_{\tilde{\omega}_c \bigcap \omega_{c'}} \rho_c(\mathbf{x})\, dV - \int\limits_{\omega_c \bigcap \tilde{\omega}_{c'}} \rho_c(\mathbf{x})\, dV\,. \tag{9.18}$$

Please note that in the above formulas there is no ˜ over c because both meshes have the same connectivity and the indexing of the cells is the same. In the middle panel of the Fig. 21 plus and minus signs identify corresponding terms in the (9.18). The only piece which is left from cell c is hatched by vertical lines.

The main advantage of the form (9.17) is that it is in *flux* form and it will be conservative even if the reconstruction of the density on Lagrangian cells is not conservative. In the case when the same reconstruction is used, expression (9.18) is analytically equivalent to expression (9.15).

Meshes with same connectivity and small displacements.

In the case of two meshes with small displacement, when the rezoned cell c is contained within the union of the Lagrangian cell c and its immediate neighbors, as shown in the right panel of the Fig. 21, the following *approximate* flux form can be used

$$\tilde{m}_c = m_c + \sum_{f \in \mathcal{F}(c)} \mathcal{F}^m_f\,, \tag{9.19}$$

where there are only fluxes corresponding to the faces of the cell c.

To define an expression for the face mass flux $\mathcal{F}_f^m$ we need to introduce the notion of a region swept by a face f when it moves from its position in the Lagrangian mesh to its position in the rezoned mesh. Let us assume that face f of the Lagrangian mesh is defined by its beginning and ending points: $p_{b(f)}$, $p_{e(f)}$, and the face $\tilde{f}$ of rezoned mesh defined by the points $\tilde{p}_{b(f)}$, $\tilde{p}_{e(f)}$ – as shown in the right panel of the Fig. 21. The region swept by the face f is a quadrilateral defined as follows

$$\delta V_f = \left\{ p_{b(f)}, \tilde{p}_{b(f)}, \tilde{p}_{e(f)}, p_{e(f)} \right\} , \tag{9.20}$$

and it is shown hatched in the right panel of the Fig. 21.

If the cells c and c' share the face f then

$$\mathcal{F}_f^m = \begin{cases} \int_{\delta V_f} \rho_{c'}(\mathbf{x})\, dV \,, & \text{when } |\delta V_f|_{signed} \geq 0 \,, \\ \int_{\delta V_f} \rho_c(\mathbf{x})\, dV \,, & \text{when } |\delta V_f|_{signed} < 0 \,, \end{cases} \tag{9.21}$$

where $|\delta V_f|_{signed}$ is the signed volume of the swept region according to the orientation defined by (9.20).

The face fluxes use only the information from the neighbors that share the face, no information from the cells that share the corner point is used, this can lead to large errors for discontinuous density fields. Another potential problem with the approximate flux approach can occur when the swept region is self intersecting as can happen for example with face which come next in counter-clock-wise order to the face f in the right panel of Fig. 21. If the triangles marked by plus and minus signs in this figure have approximately the same area then the overall flux can be close to zero even though the density in cell c and its face neighbor can be very different and the real mass exchange between these two cells should be defined by difference in the densities in these two cells. In 2D there is a relatively simple fix for this problem which only require the intersection of faces f and $\tilde{f}$ to be found and the splitting the face flux into two pieces – positive and negative parts corresponding to the two triangles and using the appropriate reconstruction for each triangle, [98].

9.1.5. Remapping of other cell-centered quantities

For both the SGH and CGH discretizations we need to remap the cell-centered internal energy, in CGH we also need to remap components of the cell-centered velocity. In both cases we may want to remap kinetic energy. In all these cases the remap must be conservative. This is important for so-called *kinetic energy fix* explained in Section 9.1.6.

Let us first consider the intersection-based remap for remapping of the internal energy in single material cells. On a Lagrangian mesh the total internal energy in a cell is $\mathcal{E}_c = m_c\, \bar{\varepsilon}_c = |\omega_c|\, \bar{\rho}_c\, \bar{\varepsilon}_c$, where $\bar{\varepsilon}_c$ is the mass averaged mean specific internal energy.

There are two approaches for producing a conservative reconstruction. The standard approach is to reconstruct the product of the density and the internal energy

$$(\rho\, \varepsilon)_c(\mathbf{x}) = \bar{\rho}_c\, \bar{\varepsilon}_c + (\mathbf{GRAD}(\rho\, \varepsilon))_c \cdot (\mathbf{x} - \mathbf{x}_c^{centr}) \quad \mathbf{x} \in \omega_c \,, \tag{9.22}$$

which is a conservative reconstruction

$$\int_{\omega_c} (\rho\, \varepsilon)_c(\mathbf{x})\, dV = |\omega_c|\, \bar{\rho}_c\, \bar{\varepsilon}_c = m_c\, \bar{\varepsilon}_c \,. \tag{9.23}$$

In this case the reconstruction and the limiting is similar to the density reconstruction. The remapped total internal energy, $\tilde{\mathcal{E}}_{\tilde{c}}$, is defined by the integration of $(\rho\, \varepsilon)_c(\mathbf{x})$ over the intersections of the Lagrangian and the rezoned meshes. The remapped specific internal energy is then defined as

$$\tilde{\bar{\varepsilon}}_{\tilde{c}} = \tilde{\mathcal{E}}_c \Big/ \tilde{m}_{\tilde{c}} \,.$$

In this case there is no guarantee that value of the remapped specific internal energy will be in bounds. Also, this kind of remap does not satisfy our requirements that the contributions to remapped total internal energy has to be equal to the mass contribution multiplied by the value of the specific internal energy.

The second approach, which was introduced in [77] in a different context, is based on the reconstruction of the specific internal energy itself but with respect to a center of mass computed from the reconstructed density

$$\varepsilon_c(\mathbf{x}) = \bar{\varepsilon}_c + (\mathbf{GRAD}\varepsilon)_c \cdot \left(\mathbf{x} - \mathbf{x}_c^{cm}\right) \,, \quad \mathbf{x} \in \omega_c \,, \tag{9.24}$$

where

$$\mathbf{x}_c^{cm} = \left(\int_{\omega_c} \rho_c(\mathbf{x})\, dV \right) \Big/ m_c \,,$$

is the center of mass of cell ω_c with respect to the reconstructed density.

It is a conservative reconstruction in the following sense

$$\int_{\omega_c} \rho_c(\mathbf{x})\, \varepsilon_c(\mathbf{x})\, dV = |\omega_c|\, \bar{\rho}_c\, \bar{\varepsilon}_c = m_c\, \bar{\varepsilon}_c\,. \tag{9.25}$$

This approach requires the integration of the quadratic function (product of the linear reconstruction of the density and the specific internal energy) over the polygons.

For the intersection-based remap the mass averaged values of the remapped internal energy can be expressed in a similar manner to the volume averaged density, (9.16),

$$\bar{\tilde{\varepsilon}}_{\tilde{c}} = \sum_c \varepsilon_{\tilde{c},c} \cdot \frac{\int_{\tilde{\omega}_{\tilde{c}} \cap \omega_c} \rho_c(\mathbf{x})\, dV}{\sum_c \left(\int_{\tilde{\omega}_{\tilde{c}} \cap \omega_c} \rho_c(\mathbf{x})\, dV \right)}\,, \tag{9.26}$$

where according to the general mean value theorem $\varepsilon_{\tilde{c},c}$ is the value of $\varepsilon_c(\mathbf{x})$ at some point in $\tilde{\omega}_{\tilde{c}} \cap \omega_c$. The expression (9.26) means that the contribution to the total internal energy is proportional to the corresponding mass contribution. This approach guarantees bound preservation – Section 9.3.

A similar approach can be used for the velocity components in the CGH case.

The equation (9.26) also means that the remap satisfies the so-called DeBar condition, [60]. That is the remap is exact for constant internal energy, but arbitrary density distribution. This also applies to any other quantity that is also remapped with mass.

Where it is possible to perform a flux based remap, and the mass is remapped according to Eq. (9.19), the remap of internal energy is performed as follows

$$\tilde{\mathcal{E}}_c = \mathcal{E}_c + \sum_{f \in \mathcal{F}(c)} \varepsilon_f\, \mathcal{F}_f^m\,, \tag{9.27}$$

where ε_f is a value of the specific internal energy associated with a face, this could for example be a value of the reconstructed specific energy from the same cell from which the reconstructed density was used to construct the mass flux. The requirements on this value with regard to bound preservation are considered in Section 9.3.

9.1.6. Kinetic energy fix

For the SGH method if the total energy is computed as the sum of the conservatively remapped internal energy and the kinetic energy computed from the remapped velocity and the remapped mass it will not be conservative, because in general, there is some dissipation missing. Some codes ignore this discrepancy. Other codes use an approach often termed the *kinetic energy fix* to restore total energy conservation. In this approach one also conservatively remaps the kinetic energy. A modification is then made to ensure the remapped kinetic energy plus remapped internal energy sums to give total energy conservation. The discrepancy between actual and remapped kinetic energy is added to the internal energy of the materials to achieve this.

If we introduce notation $\tilde{K}_c^{act}$ for the actual kinetic energy after remap and $\tilde{K}_c^{rem}$ for the conservatively remapped kinetic energy then the final values of the remapped specific energy in the cell $\tilde{c}$ will be

$$\bar{\tilde{\varepsilon}}_{\tilde{c}}^{final} = \bar{\tilde{\varepsilon}}_{\tilde{c}} + \left(\tilde{K}_c^{rem} - \tilde{K}_c^{act} \right) \Big/ \tilde{m}_{\tilde{c}}\,. \tag{9.28}$$

In the multimaterial case the discrepancy in the kinetic energy is distributed between internal energies of the materials.

9.2. Remap of nodal quantities – SGH

The remapping of the velocity for the SGH method requires special consideration. A single velocity model is used in our discretizations and therefore velocity does not “see” materials.

Let us first consider the case of two arbitrary meshes. First of all, because we want to conserve momentum we need to remap nodal momentum and nodal mass and then define the nodal remapped velocities as a ratio of the remapped momentum and the remapped nodal masses. This brings into consideration the need for consistent remapping of cell and nodal masses.

At first glance, there is a straightforward solution based on the reconstruction of the velocity components on cells of the dual mesh surrounding nodes and then using the intersection-based remap to perform intersection with cells of the dual meshes. However, cells of dual meshes can be non-convex and the intersection of non-convex polygons is very expensive and a non-trivial task. However, a more serious problem is the potentially inconsistent definition of the nodal and cell masses.

One solution is suggested in [129], where one remaps the subcell masses which participate in the Lagrangian equations and then combine remapped cell and nodal masses from these subcell masses. In [129], momentum that is conserved by the Lagrangian phase is also distributed between the subcells and then remapped on a subcell basis and then the remapped subcell momentum is combined to define nodal remapped momentum. The subcell remapping of mass and momentum is performed using an intersection-based approach.

There are other more recent approaches based on the redistribution principle [170,171], but we do not have space in this paper to give the details.

Now, let us consider the case where a flux-based remap is possible. To define the remapped nodal mass we will use the following expression

$$\tilde{m}_p = m_p + \sum_{p' \in \mathcal{P}(p)} \mathcal{F}^m_{p,p'}, \tag{9.29}$$

where $\mathcal{P}(p)$ is the set of all the points which are connected to point p by an edge. The expression (9.29) is similar to (9.19), but is applied to a cell of the dual mesh. The mass flux between these points has to be consistent with the mass fluxes between the cells. There are several approaches that can be used to define such fluxes, [24,115,153,103], and interested reader can refer to these papers. Here we just mention that because we use a single velocity model the mass flux includes the contributions from all the materials.

Then using the nodal mass flux the remapping of the momentum $\boldsymbol{\mu}_p$ proceeds as follows

$$\tilde{\boldsymbol{\mu}}_p = \boldsymbol{\mu}_p + \sum_{p' \in \mathcal{P}(p)} \mathbf{u}_{p,p'} \mathcal{F}^m_{p,p'},$$

where $\mathbf{u}_{p,p'}$ is some velocity associated with the edge p, p'. The bound preservation of the nodal velocity remap will be addressed in the next Section 9.3.

9.3. Enforcing bound preservation

We will need to distinguish several cases. We start with the remapping of cell-centered scalar quantities using intersection-based remap. If the density reconstruction satisfies local bounds then from Eq. (9.16) we can conclude that the remapped density will also be in bounds because the density $\rho_{\tilde{c},c}$ is obtained from the reconstructed density in cell c, which is in bounds and the weights $\left|\tilde{\omega}_{\tilde{c}} \cap \omega_c\right| / |\tilde{\omega}_{\tilde{c}}|$ are positive and sum to one. Similar considerations hold for the remapping of the internal energy, if the reconstruction of the specific internal energy satisfies the local bounds. The same conclusion holds for the components of the velocity. In principle, the kinetic energy fix, (9.28) can produce negative internal energies. If it does then the *repair* procedure should be used, [116,144,129,169]. The idea of this method is conservative redistribution – that is, one brings the value of, in this case, the internal energy to the desired level and then distribute the discrepancy of the total internal energy between neighbors to restore conservation.

The enforcing of bound preservation for flux-based method is more involved. One possible approach is based on the *flux corrected remap*, (see, for example, Section 4.5 in [115] and references herein), which in turn is based on ideas from flux-corrected transport [117]. In this class of methods, one assumes that there is low-order method which guarantees bound preservation, and there is high-order method, which is more accurate for smooth solutions but can produce values out of bounds for non-smooth data. The idea is to combine fluxes from the low-order and high-order methods to produce a bound-preserving method, which is more accurate than the low-order method.

9.4. Multimaterial remapping without interface reconstruction and intersections

There are some methods for multimaterial remapping without interface reconstruction or intersections which can be used for two meshes with same connectivity and small relative displacements. These methods utilize a steep continuous remap method that minimizes numerical diffusion. The volume fraction information is computed on the grid like a remapped field. In a sense the method relies upon the use of base methods that are intrinsically unstable, which are then stabilized using monotonicity-preserving limiters, [158,95,59]. The simplest form of these schemes uses anti-upwind methods to propagate the volumes, which only applies to the incremental remap methodology. The procedure is relatively simple where the flux is computed using an anti-upwind procedure as long as this flux does not cause the creation of a new minimum or maximum in the field. Anti-upwinding is violently unstable and intrinsically anti-diffusive (in the same way that classic upwinding is diffusive). This method can be improved by moving to a specialized piece-wise linear reconstruction. The method works by choosing a steeper gradient for the reconstruction, which is generally aligned to the same direction as the accurate gradient. This has the effect of lowering any intrinsic numerical dissipation in the remap. This steeper representation is generally linearly unstable and relies upon the application of a monotonicity preserving gradient for stability. An alternative that can work well in practice is to use the steepest gradient that does not violate the monotonicity-preserving bounds. In this connection we would like to mention the following paper which deals with the design of an anti-dissipative transport scheme [67].

9.5. High-order discontinuous Galerkin advection based remap

This section provides a very brief overview of the high-order Discontinuous Galerkin (DG) advection-based ALE remap scheme. The goal of ALE remap is to transfer a field, e.g. the mass density ρ, defined on an initial spatial domain by solving the pseudo-time advection equation

$$\frac{\mathrm{d}\rho}{\mathrm{d}\tau} = u \cdot \nabla \rho \,, \tag{9.30}$$

where u is the mesh displacement vector and $\tau \in [0,1]$ is a fictitious time interval associated with a linearly interpolated transition between the original mesh and a new mesh obtained by a relaxation process.

We begin by approximating the density function (or any other thermodynamic field which is remapped) using an expansion of the form

$$\rho(\mathbf{x},\tau) = \sum_j \boldsymbol{\rho}_j(\tau)\psi_j(\mathbf{x},\tau) = \boldsymbol{\rho}(\tau)^{\mathrm{tr}}\boldsymbol{\psi}(\mathbf{x},\tau)\,, \tag{9.31}$$

where $\{\psi_i\}_{i=1}^{N_{\mathcal{A}}}$ is a (moving) basis of a *discontinuous* finite element advection space, $\boldsymbol{\rho}(\tau)$ is an unknown pseudo-time dependent vector of size $N_{\mathcal{A}} = \dim \mathcal{A}(\tau)$ and $\boldsymbol{\psi}$ is a column vector of all the basis functions. The finite element functions in the advection space $\mathcal{A}(\tau)$ are discontinuous across the interior faces of the mesh.

Consider a weak formulation of the pseudo-time advection equation (9.30) obtained by multiplying it by ψ_i and integrating over the domain $\omega(\tau)$. Then, using the Reynolds transport theorem, the fact that $\frac{\mathrm{d}\psi_i}{\mathrm{d}\tau} = 0$ and integrating by parts we get

$$\begin{aligned}\frac{d}{d\tau}\int_{\omega(\tau)} \rho\psi_i &= \int_{\omega(\tau)} \frac{\mathrm{d}\rho}{\mathrm{d}\tau}\psi_i + \rho\frac{\mathrm{d}\psi_i}{\mathrm{d}\tau} + \rho\psi_i\nabla\cdot u = \int_{\omega(\tau)} (u\cdot\nabla\rho)\psi_i + \rho\psi_i\nabla\cdot u \\ &= \int_{\omega(\tau)} \nabla\cdot(\rho u)\psi_i = -\int_{\omega(\tau)} \rho u\cdot\nabla\psi_i + \int_{\partial\omega(\tau)} \rho u\cdot n\psi_i\,.\end{aligned} \tag{9.32}$$

Expressing the density in the moving pseudo-time advection basis, it can be shown that a semi-discrete form of the advection equation is

$$\mathbf{M}_{\boldsymbol{\rho}}\frac{\partial\boldsymbol{\rho}}{\partial\tau} = \mathbf{K}_{\boldsymbol{\rho}}\boldsymbol{\rho}\,,$$

where the mass matrix is defined as

$$(\mathbf{M}_{\boldsymbol{\rho}})_{ij} \equiv \int_{\omega(\tau)} \psi_j\psi_i\,.$$

The advection matrix is defined as

$$(\mathbf{K}_{\boldsymbol{\rho}})_{ij} \equiv \sum_c \int_{\omega_c(\tau)} u\cdot\nabla\psi_j\psi_i - \sum_f \int_{\partial\omega_c(\tau)} (u\cdot n)[\![\psi_j]\!](\psi_i)_u$$

where the jump $[\![\cdot]\!]$ operator corresponds to the difference in values of the function ψ at a shared interior face:

$$[\![\psi]\!] = \psi^- - \psi^+,$$

and the term $(\psi_i)_u$ indicates the use of the upwind value associated with the shared interior face.

It is imperative that the remap process be monotonic, *i.e.* not generate any new extrema in the field variables. It is well known that the only linear methods which are guaranteed to be monotonic for such problems are first order accurate. Therefore, in order to ensure that the advection remap process is monotonic, we employ the use of non-linear Flux Corrected Transport which blends both high and low (first) order solutions to achieve monotonicity and preserve high order accuracy when the field is sufficiently smooth. The details of how this is done can be found in [7].

We can summarize the overall high-order finite element ALE process in semi-discrete form as

	Lagrangian Phase	**Remap Phase**
Mass:	$\rho(t)J_c(t) = \rho(0)J_c(0)$	$\mathbf{M}_{\boldsymbol{\rho}}\frac{\partial\boldsymbol{\rho}}{\partial\tau} = \mathbf{K}_{\boldsymbol{\rho}}\boldsymbol{\rho}$
Momentum:	$\mathbf{M}_{\mathcal{U}}\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\mathbf{F}\cdot\mathbf{1}$	$\mathbf{M}_{\mathcal{U}}\frac{\partial\mathbf{u}}{\partial\tau} = \mathbf{K}_{\mathcal{U}}\mathbf{u}$
Energy:	$\mathbf{M}_{\mathcal{E}}\frac{\mathrm{d}\varepsilon}{\mathrm{d}t} = \mathbf{F}^{\mathbf{T}}\cdot\mathbf{v}$	$\mathbf{M}_{\mathcal{E}}\frac{\partial\varepsilon}{\partial\tau} = \mathbf{K}_{\mathcal{E}}\varepsilon$

where $\mathbf{F}$ is the rectangular force matrix, $\boldsymbol{\rho}$ is the density with discontinuous basis ψ, $\mathbf{u}$ is the velocity with continuous vector basis W, and ε is the energy with discontinuous basis ϕ.

The mass matrices are defined as

$$(\mathbf{M}_{\boldsymbol{\rho}})_{ij} = \int_{\Omega} \psi_j \psi_i \,,$$

$$(\mathbf{M}_{\mathcal{U}})_{ij} = \int_{\Omega} \rho W_j W_i \,,$$

$$(\mathbf{M}_{\mathcal{E}})_{ij} = \int_{\Omega} \rho \phi_j \phi_i \,,$$

and the advection matrices are defined as

$$(\mathbf{K}_{\boldsymbol{\rho}})_{ij} = \sum_c \int_{\omega_c} u \cdot \nabla \psi_j \psi_i - \sum_f \int_{\partial\omega_c(\tau)} (u \cdot n)[\![\psi_j]\!](\psi_i)_d \,,$$

$$(\mathbf{K}_{\mathcal{U}})_{ij} = \sum_c \int_{\omega_c} \rho u \cdot \nabla W_j W_i \,,$$

$$(\mathbf{K}_{\mathcal{E}})_{ij} = \sum_c \int_{\omega_c} \rho u \cdot \nabla \phi_j \phi_i - \sum_f \int_{\partial\omega_c(\tau)} \rho_u (u \cdot n)[\![\phi_j]\!](\phi_i)_d \,.$$

Note that in our formulation, the mass and energy remap equations use a Discontinuous Galerkin (DG) method while the momentum remap uses continuous Galerkin (CG).

10. Further reading

In this section we give an annotated list of the references for further reading, which mostly include references to topics which are not covered in this paper. This mostly consists of papers by the authors of the current paper and their collaborators. This is a very incomplete list and there has been no attempt to cover the history of the development of the multimaterial ALE methods.

The community of researches working on the modeling of high-speed compressible multimaterial flows have regular meetings, which started in 2001 as a minisymposia at the SIAM Annual meeting and then transformed to the International Conference on Numerical Methods for Multi-Material Fluid Flow which is now held every two years, [1]. It focuses on the mathematical and numerical aspects of Lagrangian, Eulerian, and Arbitrary Lagrangian Eulerian methods for multimaterial and multiphase flow problems.

10.1. Lagrangian phase SGH

Theoretical questions

Further details on the construction of compatible discretization for unstructured meshes are given in [47]. An important issue related to the consistency of the volume change and the $p\,dv$ work term in the internal energy equation are also considered in [128,20]. The angular momentum conservation and incremental objectivity and stability properties of a predictor/multi-corrector method for Lagrangian shock hydrodynamics are considered in [131,130].

Axisymmetric extension

The extension to axisymmetric geometry is essential to cope with real life applications. The issues related to the extension to axisymmetric geometry and in particular the preservation of spherical symmetry are presented in [103,12].

Curvilinear meshes

The extension of the compatible discretization to curvilinear meshes is considered in [140]. The application of curvilinear meshes in the isogeometric analysis of Lagrangian hydrodynamics is presented in [21,22].

Artificial viscosity

Artificial viscosity is critical to the success of Lagrangian hydrodynamics. Reader can refer to the following papers which describe several approaches for constructing artificial viscosity methods and provide some comparisons of the numerical results obtained with these different methods, [51,46,121,120,127].

Closure models

A comparative study of multimaterial Lagrangian and Eulerian methods with pressure relaxation presented in [83].

3D extension

Conceptually compatible discretization for SGH can easily be extended to $3D$, however, there are a lot of bookkeeping issues related to the fact that in $3D$ the faces of a cell are not flat and typically need to be subdivide it into triangles, [49,42].

Solids

Because compatible discretizations are formulated in terms of corner forces or discrete operators it is relatively straightforward to extend these methods to solids, [42,145]. Readers who are interested in the history of applying staggered discretizations for elastic–plastic flows can refer to [149,179].

Slide lines

Slide line can be viewed as boundary conditions that couple Lagrangian grids together by enforcing contact conditions (continuity of the normal velocity component and the normal stress component). A number of different slide line (slide surfaces) algorithms are described in [179,24,183,118,102].

Reconnection-based ALE – ReALE

Readers interested in the recently developed reconnection-based ALE – ReALE method should refer to [126,125,94,31,32].

10.2. CGH approach

The capabilities of both GLACE and EUCCLHYD have been extend to second-order [134] two-dimensional axisymmetric [147,133] and three-dimensional [53,139,89] geometries, and also to provide an indirect ALE capability [84]. The robustness of the CGH formulation might be improved further by employing an ad hoc stabilization mechanism based on subzonal entropy, refer to [65]. A CGH formulation with the capability to preserve angular momentum is also presented in [66].

The successful and promising results obtained with the CGH method have promoted an increased interest in the development of cell-centered FV methods for Lagrangian hydrodynamics, [18,17,56,124,55]. At the end of the 2000's a third variant of cell-centered Lagrangian scheme named CCH (Cell-Centered Hydrodynamics) was proposed [43,163,162,45]. Following the methodology developed by the preceding investigators, CCH relies on a general formalism which addresses the extension to solids and general multi-dimensional unstructured grids employing the data structures initially introduced for the SGH method in [41].

Solids

The CGH scheme has also been extended to solids. Firstly, in the context of hyperelasticity a CGH scheme was proposed in [104] to solve the two-dimensional nonlinear elasticity equations. Secondly, in [163,162,136] the solid dynamics equations based on a hypoelastic constitutive law (the so-called Wilkins model) have been solved using a CGH approach.

Slide lines

The slide line approach can be viewed as a boundary condition that couples Lagrangian grids together by enforcing contact conditions (continuity of the normal velocity component and the normal stress component). Slide line methods have been developed in the context of SGH method for many years. The extension of the CGH capability to slide lines is more recent and can be found in [150]. Original approaches to model this geometrical constraint have also been proposed in [58,28].

Curvilinear grids and very high-order extension (beyond second-order)

A generalization of the CGH formulation to curvilinear grids has been proposed in [61]. The very high-order extension of cell-centered Lagrangian hydrodynamics is far from being a trivial task since this requires curvilinear grids as noted in [54] where a third-order ENO CGH scheme was developed which employs a curvilinear quadrangular grid. In addition, a very high-order CGH discretization of the gas dynamics equations in total Lagrangian form has recently been constructed based on the Discontinuous Galerkin approach. The interested reader may refer to [3,101,176].

Links between SGH and CGH approaches

It is worth mentioning that the links between SGH and CGH approaches have been explored by developing SGH schemes where the discretization of momentum equation is obtained by means of a node-centered FV approximation. In this approach the numerical flux approximations are constructed using a cell-centered multi-dimensional approximated Riemann solver. This results in a multidimensional staggered Lagrangian hydrodynamics scheme with an artificial viscosity based upon a cell-centered Riemann solver, refer to [40,138,127].

ALE, mesh adaptation and ReALE

An indirect ALE CGH approach which utilizes curvilinear grids is described in [37]. A metric-based mesh adaptation methodology is described in [62]. Finally, we note in passing that the ReALE methodology has been developed employing both the SGH and the CGH Lagrangian schemes, refer to [126]. The important question of reducing the spurious entropy production within rarefaction regions has also been studied within the context of a Lagrange-Remap CGH-based method, refer to [39].

10.3. High-order finite element SGH approach

Axisymmetric extension

In the axisymmetric extension of the high-order FE SGH approach described in [72], we formulate our semi-discrete axisymmetric conservation laws directly in 3D and reduce them to a 2D variational form in a meridian cut of the original domain. This approach is a natural extension of the high-order curvilinear finite element framework we have developed for 2D and 3D problems in Cartesian geometry, leading to a rescaled momentum conservation equation which includes new radial terms in the pressure gradient and artificial viscosity forces.

Solids

The extension of the high-order FE SGH approach to solids is described in [75] where a semi-discrete equation for the deviatoric stress rate is developed for 2D planar, 2D axisymmetric and full 3D geometries. For each case, the strain rate is approximated via a collocation method at zone quadrature points while the deviatoric stress is approximated using an L_2 projection onto the thermodynamic basis.

Closure Models

Full details on the development a multi-material closure model for the high-order FE SGH approach can be found in [74].

10.4. Rezone

The details of the untangling method are fully described in [173,27]. The foundation, different flavors and full details of the optimization-based rezoning can be found in [109,106,105,87,88,86,107,79].

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 shapes (or contact discontinuity) in spite of efforts at regularization. The most general solution to this difficulty, while preserving a Lagrangian nature to the mesh motion is to relax the constraints on mesh topology and allow reconnection at the rezone stage. We briefly describe one possible reconnection-based rezone strategy, [126], which uses the machinery of Voronoi meshes, [10]. The details of reconnection-based rezoning can be found in [126, 31,32].

10.5. Interface reconstruction

A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations has been considered in [164]. Some details of interface reconstruction on an analytical formula are considered in [70, 71].

10.6. Remap

A complete description of the remap for multimaterial SGH ALE is described in [115]. The different aspects of the multimaterial intersection-based remap are described in [77,25,114,143,113,116,111]. The specific questions which relate to the remap of nodal quantities have been addressed in [129,170,171,153]. Some questions relate to preservation of symmetry at remap stage addressed in [174]. A method for reducing dissipation at remap stage is also presented in [11]. The main ideas related to flux-corrected remapping are also presented in [122,123]. A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations is considered in [181]. Remapping in 3D has been addressed in [85]. The repair process for enforcing bound preservation is described in [116,144,129,169]. Some ideas about high-order remap are described in [175]. The optimization-based remap is described in [35,34,33]. The remap of divergence-free fields is described in [36]. Methods for enforcing monotonicity for the high-order finite element DG remap are described in [8]. Remapping methods based on multi-dimensional optimal order detection (MOOD), [30], has been presented at several conferences.

11. Acknowledgment

The work of R. Rieben was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

The work of M. Shashkov was carried out under the auspices of the National Nuclear Security Administration of the U.S. Department of Energy at Los Alamos National Laboratory under Contract No. DE-AC52-06NA25396 and the DOE Office of Science Advanced Scientific Computing Research (ASCR) Program in Applied Mathematics Research.

This work of P.-H. Maire has been carried out with financial support from the French State, managed by the French National Research Agency (ANR) in the frame of the Investments for the future Programme IdEx Bordeaux (ANR-10-IDEX-03-02).

We would like to that following people for fruitful discussions over many years: D. Benson, D. Bailey, M. Owen, D. Miller, A. Favorskii, A. Samarskii, V. Tishkin, V. Goloviznin, Y. Yanilkin, L. Margolin, D. Burton, M. Kenamond, R. Loubere, R. Hill, E. Caramana, P. Whalen, B. Wendroff, S. Sambasivan, J. Hyman, J. Breil, S. Galera, W. Bo, J. Velechovskỳ, G. Scovazzi, P. Bochev, D. Ridzal, V. Dyadechko, H.-T. Ahn, S. Schofield, A. Harrison, J. Fung, M. Berndt, K. Lipnikov, M. Kucharik, R. Liska, P. Vachal, J. Campbell, A. Dawes, P. Knupp, B. Despres, R. Garimella, Y. Bazilevs, I. Sofronov, V. Rasskazova, S. Bakrach, J. Dukowicz, P. Hoch, S. Del Pino, J. Grandy, H. Trease, M. Gunzburger, J. Brock, B. Kashiva, F. Adessio, B. Rebourcet, P. Smolarkiewicz, D. Kothe, A. Solovjov, R. Pember, M. Andrews, J. Quirk, G. Ball, T. Barth, R. Lowrie, J. Morel, A. Robinson, G. Lutwak, M. Francois, E. Dendy, B. Swartz, G. Dilts, J. Grove, T. Masser, R. Abgrall, B. Archer, B. Webster, P. Roe, J. Ovadia, F. Vilar, S. Gavrilyuk, B. Nkonga, T. Kolev, V. Dobrev, V. Tomov, R. Anderson, T. Brunner, T. Ellis, E. Love, J.S. Peery, T. Trucano.

References

[1] Multimat Conferences Series, https://www.multimat.math.tu-dortmund.de;

https://multimat13.llnl.gov;
http://multimat2011.celia.u-bordeaux1.fr;
http://onlinelibrary.wiley.com/doi/10.1002/fld.2546/full;
http://www-troja.fjfi.cvut.cz/~multimat07;
http://www.icfd.reading.ac.uk/Multimaterial-Workshop.htm;
http://www.ann.jussieu.fr/~despres/hydro2.html;
http://meetings.siam.org/sess/dsp_programsess.cfm?SESSIONCODE=558.
[2] R. Abgrall, J. Breil, P.-H. Maire, J. Ovadia, Un schéma centré pour l'hydrodynamique Lagrange bidimensionnelle, Technical Report LRC MAB CEA 01/2004, Université de Bordeaux, 2004, available at https://www.researchgate.net/publication/236646889.
[3] R. Abgrall, R. Loubère, J. Ovadia, A Lagrangian discontinuous Galerkin-type method on unstructured meshes to solve hydrodynamics problems, Int. J. Numer. Methods Fluids 44 (6) (2004) 645–663.
[4] M. Aftosmis, D. Gaitonde, T.S. Tavares, Behavior of linear reconstruction techniques on unstructured meshes, AIAA J. 33 (11) (1995) 2038–2049.
[5] H.-T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2) (2007) 2096–2132.
[6] A.A. Amsden, C.W. Hirt, YAQUI: an arbitrary Lagrangian–Eulerian computer program for fluid flow at all speeds, Technical Report LA-5100, Los Alamos National Laboratory, 1973, http://www.osti.gov/scitech/biblio/4495964-yaqui-arbitrary-lagrangian-eulerian-computer-program-fluid-flow-all-speeds.
[7] R. Anderson, V. Dobrev, Tz. Kolev, R. Rieben, Monotonicity in high-order curvilinear finite element ALE remap, Int. J. Numer. Methods Fluids 77 (2015) 249–273.
[8] R.W. Anderson, V.A. Dobrev, T.V. Kolev, R.N. Rieben, Monotonicity in high-order curvilinear finite element arbitrary Lagrangian–Eulerian remap, Int. J. Numer. Methods Fluids 77 (5) (2014) 249–273.
[9] P. Anninos, New VOF interface capturing and reconstruction algorithms, Technical Report UCRL-ID-135084, Lawrence Livermore National Laboratory, 1999.
[10] F. Aurenhammer, Voronoi diagrams – a survey of fundamental geometric data structures, ACM Comput. Surv. 23 (3) (1991) 345–405.
[11] D. Bailey, M. Berndt, M. Kucharik, M. Shashkov, Reduced-dissipation remapping of velocity in staggered arbitrary Lagrangian–Eulerian methods, J. Comput. Appl. Math. 233 (12) (2010) 3148–3156.
[12] A. Barlow, D. Burton, M. Shashkov, Compatible, energy and symmetry preserving 2D lagrangian hydrodynamics in rz-cylindrical coordinates, Proc. Comput. Sci. 1 (1) (2010) 1893–1901.
[13] A. Barlow, R. Hill, M. Shashkov, Constrained optimization framework for interface-aware sub-scale dynamics closure models for multimaterial cells in Lagrangian and arbitrary Lagrangian–Eulerian hydrodynamics, Technical Report LA-UR-13-26180, Los Alamos National Laboratory, 2013.
[14] A. Barlow, R. Hill, M. Shashkov, Constrained optimization framework for interface-aware sub-scale dynamics closure model for multimaterial cells in Lagrangian and arbitrary Lagrangian–Eulerian hydrodynamics, J. Comput. Phys. 276 (2014) 92–135.
[15] A.J. Barlow, ALE and AMR mesh refinement techniques for multi-material hydrodynamics problems, in: ICFD Workshop on Mesh Refinement Techniques, 7th December 2005; http://www.icfd.rdg.ac.uk/Workshops/AMR/meshreftechf.pdf.
[16] A.J. Barlow, Challenges and recent progress in developing numerical methods for multi-material ALE hydrocodes, in: ICFD 25 year Anniversary Conference, Oxford University, 15–16th September 2008, http://www.icfd.rdg.ac.uk/ICFD25/Talks/ABarlow.pdf.
[17] A.J. Barlow, A high-order cell-centred dual grid Lagrangian Godunov scheme, Comput. Fluids 83 (2013) 15–26.
[18] A.J. Barlow, P.L. Roe, A cell-centred Lagrangian Godunov scheme for shock hydrodynamics, Comput. Fluids 46 (2011) 133–136.
[19] T.J. Barth, D.C. Jespersen, The design and application of upwind schemes on unstructured meshes, in: AIAA Paper 89-0366, 27th Aerospace Sciences Meeting, Reno, Nevada, 1989.
[20] A.L. Bauer, D.E. Burton, E.J. Caramana, R. Loubère, 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.
[21] Y. Bazilevs, I. Akkerman, D.J. Benson, G. Scovazzi, M.J. Shashkov, Isogeometric analysis of Lagrangian hydrodynamics, J. Comput. Phys. 243 (2013) 224–243.
[22] Y. Bazilevs, C.C. Long, I. Akkerman, D.J. Benson, M.J. Shashkov, Isogeometric analysis of Lagrangian hydrodynamics: axisymmetric formulation in the rz-cylindrical coordinates, J. Comput. Phys. 262 (2014) 244–261.
[23] M. Ben-Artzi, J. Falcovitz, Generalized Riemann Problem in Computational Fluid Dynamics, Cambridge Monographs on Applied and Computational Mathematics, Cambridge University Press, 2003.
[24] D.J. Benson, Computational methods in Lagrangian and Eulerian hydrocodes, Comput. Methods Appl. Mech. Eng. 99 (2–3) (1992) 235–394.
[25] M. Berndt, J. Breil, S. Galera, M. Kucharik, P.-H. Maire, M. Shashkov, Two-step hybrid conservative remapping for multimaterial arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 230 (17) (2011) 6664–6687.
[26] M. Berndt, N.N. Carlson, Using polynomial filtering for rezoning in ale, Technical Report LA-UR 11-05015, Los Alamos National Laboratory, 2011.
[27] M. Berndt, M. Kucharik, M.J. Shashkov, Using the feasible set method for rezoning in ALE, Proc. Comput. Sci. 1 (1) (2010) 1885–1892.
[28] S. Bertoluzza, S. Del Pino, E. Labourasse, A conservative slide line method for cell-centered semi-Lagrangian and ALE schemes in 2D, Modél. Math. Anal. Numér. 50 (2016) 187–214.
[29] H.A. Bethe, The theory of shock waves for an arbitrary equation of state, Technical Report OSRD-545, Office of Scientific Research and Development, 1942.
[30] G. Blanchard, R. Loubère, High order accurate conservative remapping scheme on polygonal meshes using a posteriori MOOD limiting, Comput. Fluids 136 (2016) 83–103.
[31] W. Bo, M. Shashkov, R-adaptive reconnection-based arbitrary Lagrangian Eulerian method – R-ReALE, J. Math. Study 48 (2) (2015) 125–167.
[32] W. Bo, M. Shashkov, Adaptive reconnection-based arbitrary Lagrangian Eulerian method, J. Comput. Phys. 299 (2015) 902–939.
[33] P. Bochev, D. Ridzal, K. Peterson, Optimization-based remap and transport: a divide and conquer strategy for feature-preserving discretizations, J. Comput. Phys. 257 (2014) 1113–1139.
[34] P. Bochev, D. Ridzal, G. Scovazzi, M. Shashkov, Formulation, analysis and numerical study of an optimization-based conservative interpolation (remap) of scalar fields for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 230 (13) (2011) 5199–5225.
[35] P. Bochev, D. Ridzal, M. Shashkov, Fast optimization-based conservative remap of scalar fields through aggregate mass transfer, J. Comput. Phys. 246 (2013) 37–57.
[36] P. Bochev, M. Shashkov, Constrained interpolation (remap) of divergence-free fields, Comput. Methods Appl. Mech. Eng. 194 (2) (2005) 511–530.
[37] B. Boutin, E. Deriaz, P. Hoch, P. Navaro, Extension of ALE methodology to unstructured conical meshes, ESAIM Proc. 32 (2011) 31–55.
[38] J.U. Brackbill, J.S. Saltzman, Adaptive zoning for singular problems in two dimensions, J. Comput. Phys. 46 (3) (1982) 342–368.
[39] J.-P. Braeunig, Reducing the entropy production in collocated Lagrange-remap scheme, J. Comput. Phys. 314 (2016) 127–144.
[40] A. Burbeau-Augoula, A node-centered artificial viscosity method for two-dimensional Lagrangian hydrodynamics calculations on a staggered grid, Commun. Comput. Phys. 8 (2010) 877–900.
[41] D. Burton, Multidimensional discretization of conservation laws for unstructured polyhedral grids, Technical report, in: Proceedings of the Second International Workshop on Analytical Methods and Process Optimization in Fluid and Gas Mechanics, VNIIEF, Sarov, Russia, 1994, 1994, available as UCRL-JC-118306; CONF-9409314-1 at http://www.osti.gov.

[42] D. Burton, Lagrangian hydrodynamics in the FLAG code, Technical Report LA-UR-07-7547, Los Alamos National Laboratory, 2007, available at https://www.researchgate.net/publication/237101823.
[43] D.E. Burton, T.C. Carney, N.R. Morgan, S.K. Sambasivan, M.J. Shashkov, A cell-centered Lagrangian Godunov-like method for solid dynamics, Comput. Fluids 83 (2013) 33–47.
[44] D.E. Burton, N.R. Morgan, T.C. Carney, M.A. Kenamond, Reduction of dissipation in Lagrange cell-centered hydrodynamics (CCH) through corner gradient reconstruction (CGR), J. Comput. Phys. 299 (2016) 229–280.
[45] D.E. Burton, T.C. Carney, N.R. Morgan, On the question of area weighting in cell-centered hydrodynamics, Technical Report LA-UR-13-23155.2, Los Alamos National Laboratory, 2013.
[46] J.C. Campbell, M.J. Shashkov, A tensor artificial viscosity using a mimetic finite difference algorithm, J. Comput. Phys. 172 (2) (2001) 739–765.
[47] J.C. Campbell, M.J. Shashkov, A compatible Lagrangian hydrodynamics algorithm for unstructured grids, Selçuk J. Appl. Math. 4 (2) (2003) 53–70.
[48] 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.
[49] E.J. Caramana, C.L. Rousculp, D.E. Burton, A compatible, energy and symmetry preserving Lagrangian hydrodynamics algorithm in three-dimensional Cartesian geometry, J. Comput. Phys. 157 (2000) 89–119.
[50] E.J. Caramana, M.J. Shashkov, Elimination of artificial grid distortion and hourglass-type motions by means of Lagrangian subzonal masses and pressures, J. Comput. Phys. 142 (2) (1998) 521–561.
[51] E.J. Caramana, M.J. Shashkov, P.P. Whalen, Formulations of artificial viscosity for multi-dimensional shock wave computations, J. Comput. Phys. 144 (1) (1998) 70–97.
[52] G.F. Carey, Computational Grids: Generation, Adaptation and Solution Strategies, Series in Computational and Physical Processes in Mechanics and Thermal Sciences, Taylor & Francis, 1997.
[53] G. Carré, S. Del Pino, B. Després, E. Labourasse, A cell-centered Lagrangian hydrodynamics scheme on general unstructured meshes in arbitrary dimension, J. Comput. Phys. 228 (14) (2009) 5160–5183.
[54] J. Cheng, C.-W. Shu, A third-order conservative Lagrangian type scheme on curvilinear meshes for the compressible Euler equations, Commun. Comput. Phys. 4 (2008) 1008–1024.
[55] J. Cheng, C.-W. Shu, A cell-centered Lagrangian scheme with the preservation of symmetry and conservative properties for compressible fluid flows in two-dimensional cylindrical geometry, J. Comput. Phys. 229 (2010) 7191–7206.
[56] J.C. Cheng, C.-W. Shu, A high-order ENO conservative Lagrangian type scheme for the compressible Euler equations, J. Comput. Phys. 227 (2007) 1567–1596.
[57] S.-W. Cheng, T.K. Dey, J.R. Shewchuk, Delaunay Mesh Generation, CRC Press, 2012.
[58] G. Clair, B. Després, E. Labourasse, A new method to introduce constraints in cell-centered Lagrangian schemes, Comput. Methods Appl. Mech. Eng. 261–262 (2013) 56–65.
[59] P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations, J. Comput. Phys. 54 (1) (1984) 174–201.
[60] R.B. DeBar, Fundamentals of the KRAKEN code. [Eulerian hydrodynamics code for compressible nonviscous flow of several fluids in two-dimensional (axially symmetric) region], Technical report, California Univ, Livermore (USA), Lawrence Livermore Lab, 1974.
[61] S. Del Pino, A curvilinear finite-volume method to solve compressible gas dynamics in semi-Lagrangian coordinates, C. R. Acad. Sci. Paris, Ser. I 348 (2010) 1027–1032.
[62] S. Delpino, Metric-based mesh adaptation for 2D Lagrangian compressible flows, J. Comput. Phys. 230 (2011) 1793–1821.
[63] J.-A. Desideri, A. Dervieux, Compressible flow solvers using unstructured grids, in: Computational Fluid Dynamics, in: Lecture Series 1988-05, Von Karman Institute, 1988, available at https://hal.archives-ouvertes.fr/file/index/docid/76971/filename/RR-1732.pdf.
[64] B. Després, Weak consistency of the cell-centered lagrangian GLACE scheme on general meshes in any dimension, Comput. Methods Appl. Eng. 199 (2010) 2669–2679.
[65] B. Després, E. Labourasse, Stabilization of cell-centered compressible Lagrangian methods using subzonal entropy, J. Comput. Phys. 231 (20) (2012) 6559–6595.
[66] B. Després, E. Labourasse, Angular momentum preserving cell-centered Lagrangian and Eulerian schemes on arbitrary grids, J. Comput. Phys. 290 (2015) 28–54.
[67] B. Després, F. Lagoutière, E. Labourasse, I. Marmajou, An antidissipative transport scheme on unstructured meshes for multicomponent flows, Int. J. Finite Vol. 7 (2010) 30–65.
[68] B. Després, C. Mazeran, Symmetrization of Lagrangian gas dynamics in dimension two and multidimensional solvers, C. R., Méc. 331 (2003) 475–480.
[69] B. Després, C. Mazeran, Lagrangian gas dynamics in two dimensions and Lagrangian systems, Arch. Ration. Mech. Anal. 178 (3) (2005) 327–372.
[70] S. Diot, M.M. Francois, An interface reconstruction method based on an analytical formula for 3D arbitrary convex cells, J. Comput. Phys. 305 (2016) 63–74.
[71] S. Diot, M.M. Francois, E.D. Dendy, An interface reconstruction method based on analytical formulae for 2D planar and axisymmetric arbitrary convex cells, J. Comput. Phys. 275 (2014) 53–64.
[72] V.A. Dobrev, T.E. Ellis, T.V. Kolev, R.N. Rieben, High order curvilinear finite elements for axisymmetric Lagrangian hydrodynamics, Comput. Fluids 83 (5) (2013) 58–69.
[73] V.A. Dobrev, T.V. Kolev, R.N. Rieben, High-order curvilinear finite element methods for Lagrangian hydrodynamics, SIAM J. Sci. Comput. 34 (5) (2012) B606–B641.
[74] V.A. Dobrev, T.V. Kolev, R.N. Rieben, V.Z. Tomov, Multi-material closure model for high-order finite element Lagrangian hydrodynamics, Int. J. Numer. Methods Fluids (2016), http://dx.doi.org/10.1002/fld.4236, in press.
[75] V.A. Dobrev, T.V. Kolev, R.N. Rieben, High-order curvilinear finite elements for elastic–plastic Lagrangian dynamics, J. Comput. Phys. 257(B) (2014) 1062–1080.
[76] J.K. Dukowicz, A general, non-iterative Riemann solver for Godunov's method, J. Comput. Phys. 61 (1) (1985) 119–137.
[77] J.K. Dukowicz, J.R. Baumgardner, Incremental remapping as a transport/advection algorithm, J. Comput. Phys. 160 (1) (2000) 318–335.
[78] J.K. Dukowicz, J.W. Kodis, Accurate conservative remapping (rezoning) for arbitrary Lagrangian–Eulerian computations, SIAM J. Sci. Comput. 8 (3) (1987) 305–321.
[79] V. Dyadechko, R. Garimella, M. Shashkov, Reference Jacobian rezoning strategy for arbitrary Lagrangian–Eulerian methods on polyhedral grids, Technical Report LA-UR-05-8159, Los Alamos National Laboratory, 2005.
[80] V. Dyadechko, M. Shashkov, Moment-of-fluid interface reconstruction, Technical Report LA-UR-05-7571, Los Alamos National Laboratory Report, 2005.
[81] V. Dyadechko, M. Shashkov, Multi-material interface reconstruction from the moment data, Technical Report LA-UR-06-5846, Los Alamos National Laboratory Report, 2006.
[82] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (11) (2008) 5361–5384.
[83] M.M. Francois, M.J. Shashkov, T.O. Masser, E.D. Dendy, A comparative study of multimaterial Lagrangian and Eulerian methods with pressure relaxation, Comput. Fluids 83 (2013) 126–136.
[84] S. Galera, P.-H. Maire, J. Breil, A two-dimensional unstructured cell-centered multi-material ALE scheme using VOF interface reconstruction, J. Comput. Phys. 229 (2010) 5755–5787.

[85] R. Garimella, M. Kucharik, M. Shashkov, An efficient linearity and bound preserving conservative interpolation (remapping) on polyhedral meshes, Comput. Fluids 36 (2) (2007) 224–237.
[86] R.V. Garimella, M.J. Shashkov, Polygonal surface mesh optimization, Eng. Comput. 20 (3) (2004) 265–272.
[87] R.V. Garimella, M.J. Shashkov, P.M. Knupp, Optimization of surface mesh quality using local parametrization, in: International Meshing Roundtable, 2002, pp. 41–52.
[88] R.V. Garimella, M.J. Shashkov, P.M. Knupp, Triangular and quadrilateral surface mesh quality optimization using local parametrization, Comput. Methods Appl. Mech. Eng. 193 (9) (2004) 913–928.
[89] G. Georges, J. Breil, P.-H. Maire, A 3G GCL compatible cell-centered Lagrangian scheme for solving gas dynamics equations, J. Comput. Phys. 305 (2016) 921–941.
[90] S. Godunov, A. Zabrodine, M. Ivanov, A. Kraiko, G. Prokopov, Résolution numérique des problèmes multidimensionnels de la dynamique des gas, Editions Mir, 1979.
[91] S.K. Godunov, Reminescences about difference schemes, J. Comput. Phys. 153 (1999) 6–25.
[92] V.M. Goloviznin, A.A. Samarskii, A.P. Favorskii, A variational approach to constructing finite-difference mathematical models in hydrodynamics, Sov. Phys. Dokl. 22 (8) (1977) 432–434, American Institute of Physics.
[93] M.E. Gurtin, E. Fried, L. Anand, The Mechanics and Thermodynamics of Continua, Cambridge University Press, 2010.
[94] T. Harribey, J. Breil, P.-H. Maire, M. Shashkov, A swept-intersection-based remapping method in a ReALE framework, Int. J. Numer. Methods Fluids 72 (6) (2013) 697–708.
[95] A. Harten, The artificial compression method for computation of shocks and contact discontinuities. III. Self-adjusting hybrid schemes, Math. Comput. 32 (142) (1978) 363–389.
[96] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 14 (3) (1974) 227–253.
[97] C.W. Hirt, A.A. Amsden, J.L. Cook, An arbitrary Lagrangian–Eulerian computing method for all flow speeds, J. Comput. Phys. 135 (2) (1997) 203–216.
[98] 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, September 10–14, 2007, http://www-troja.fjfi.cvut.cz/~multimat07/presentations/tuesday/Rebourcet_Hoch.pdf.
[99] P. Hoch, E. Labourasse, A frame invariant and maximum principle enforcing second-order extension for cell-centered ALE schemes based on local convex hull preservation, Int. J. Numer. Methods Fluids 76 (12) (2014) 1043–1063.
[100] V.N. Isaev, I.D. Sofronov, Construction of discrete models for equations of gas dynamics using law of transformation of kinetic and internal energy of continuum media, VANT, Methods Codes Numer. Solut. Probl. Math. Phys. 15 (1) (1984) 3–7, in Russian.
[101] Z. Jia, S. Zhang, A new high-order discontinuous Galerkin spectral finite element for Lagrangian gas dynamics in two-dimensions, J. Comput. Phys. 230 (7) (2011) 2496–2522.
[102] M. Kenamond, M. Bement, Slideline modeling in the FLAG hydrocode, Technical Report LA-UR 11-04993, Los Alamos National Laboratory, 2011, available at https://www.researchgate.net/publication/237043721.
[103] M. Kenamond, M. Bement, M. Shashkov, Compatible, total energy conserving and symmetry preserving arbitrary Lagrangian–Eulerian hydrodynamics in 2D rz-cylindrical coordinates, J. Comput. Phys. 268 (2014) 154–185.
[104] G. Kluth, B. Després, Discretization of hyperelasticity on unstructured mesh with a cell-centered Lagrangian scheme, J. Comput. Phys. 229 (2010) 9092–9118.
[105] P. Knupp, Achieving finite element mesh quality via optimization of the Jacobian matrix norm and associated quantities. II. A framework for volume mesh optimization and the condition number of the Jacobian matrix, Int. J. Numer. Methods Eng. 48 (8) (2000) 1165–1185.
[106] P. Knupp, Introducing the target-matrix paradigm for mesh optimization via node-movement, in: S. Shontz (Ed.), Proceedings of the 19th International Meshing Roundtable, 2000, pp. 67–83, available at www.imr.sandia.gov/papers/imr19/Knupp2A.1.pdf.
[107] P. Knupp, L.G. Margolin, M. Shashkov, Reference Jacobian optimization-based rezone strategies for arbitrary Lagrangian Eulerian methods, J. Comput. Phys. 176 (1) (2002) 93–128.
[108] P. Knupp, S. Steinberg, Fundamentals of Grid Generation, CRC Press, 1993.
[109] P.M. Knupp, Algebraic mesh quality metrics for unstructured initial meshes, Finite Elem. Anal. Des. 39 (3) (2003) 217–241.
[110] C.A. Korn, T.M. Korn, Mathematical Handbook for Scientists and Engineers, McGraw-Hill Book Company, 1968.
[111] M. Kucharik, J. Breil, S. Galera, P.-H. Maire, M. Berndt, M. Shashkov, Hybrid remap for multi-material ALE, Comput. Fluids 46 (1) (2011) 293–297.
[112] M. Kuchařík, R.V. Garimella, S.P. Schofield, M.J. Shashkov, A comparative study of interface reconstruction methods for multi-material ALE simulations, J. Comput. Phys. 229 (7) (2010) 2432–2452.
[113] M. Kucharik, M. Shashkov, Extension of efficient, swept-integration-based conservative remapping method for meshes with changing connectivity, Int. J. Numer. Methods Fluids 56 (8) (2008) 1359–1365.
[114] M. Kucharik, M. Shashkov, One-step hybrid remapping algorithm for multi-material arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 231 (7) (2012) 2851–2864.
[115] M. Kucharik, M. Shashkov, Conservative multi-material remap for staggered multi-material arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 258 (2014) 268–304.
[116] M. Kucharik, M. Shashkov, B. Wendroff, An efficient linearity-and-bound-preserving remapping method, J. Comput. Phys. 188 (2) (2003) 462–471.
[117] D. Kuzmin, R. Lohner, S. Turek (Eds.), Flux-Corrected Transport: Principles, Algorithms, and Applications, Springer, 2012.
[118] W.H. Lee, Computer Simulation of Shaped Charge Problems, World Scientific, 2006.
[119] K. Lipnikov, G. Manzini, M. Shashkov, Mimetic finite difference method, J. Comput. Phys. 257 (2014) 1163–1227.
[120] K. Lipnikov, M. Shashkov, A framework for developing a mimetic tensor artificial viscosity for Lagrangian hydrocodes on arbitrary polygonal meshes, J. Comput. Phys. 229 (20) (2010) 7911–7941.
[121] K. Lipnikov, M. Shashkov, A mimetic tensor artificial viscosity method for arbitrary polyhedral meshes, Proc. Comput. Sci. 1 (1) (2010) 1921–1929.
[122] R. Liska, M. Shashkov, P. Váchal, B. Wendroff, Optimization-based synchronized flux-corrected conservative interpolation (remapping) of mass and momentum for arbitrary Lagrangian–Eulerian methods, J. Comput. Phys. 229 (5) (2010) 1467–1497.
[123] R. Liska, M. Shashkov, P. Váchal, B. Wendroff, Synchronized flux corrected remapping for ALE methods, Comput. Fluids 46 (1) (2011) 312–317.
[124] W. Liu, J. Cheng, C.-W. Shu, High-order conservative Lagrangian schemes with Lax–Wendroff type time discretization for the compressible Euler equations, J. Comput. Phys. 228 (2009) 8872–8891.
[125] R. Loubère, P.-H. Maire, M. Shashkov, ReALE: a reconnection arbitrary-Lagrangian–Eulerian method in cylindrical geometry, Comput. Fluids 46 (1) (2011) 59–69.
[126] R. Loubère, P.-H. Maire, M. Shashkov, J. Breil, S. Galera, ReALE: a reconnection-based arbitrary-Lagrangian–Eulerian method, J. Comput. Phys. 229 (12) (2010) 4724–4761.
[127] R. Loubère, P.-H. Maire, P. Vachal, 3D staggered Lagrangian hydrodynamics scheme with cell-centered Riemann solver-based artificial viscosity, Int. J. Numer. Methods Fluids 72 (2013) 22–42.
[128] R. Loubère, M. Shashkov, B. Wendroff, Volume consistency in a staggered grid Lagrangian hydrodynamics scheme, J. Comput. Phys. 227 (8) (2008) 3731–3737.

[129] R. Loubère, M.J. Shashkov, A subcell remapping method on staggered polygonal grids for arbitrary-Lagrangian–Eulerian methods, J. Comput. Phys. 209 (1) (2005) 105–138.
[130] E. Love, W.J. Rider, G. Scovazzi, Stability analysis of a predictor/multi-corrector method for staggered-grid Lagrangian shock hydrodynamics, J. Comput. Phys. 228 (20) (2009) 7543–7564.
[131] E. Love, G. Scovazzi, On the angular momentum conservation and incremental objectivity properties of a predictor/multi-corrector method for Lagrangian shock hydrodynamics, Comput. Methods Appl. Mech. Eng. 198 (41–44) (2009) 3207–3213.
[132] G. Luttwak, J. Falcovitz, Slope limiting for vectors: a novel limiting algorithm, Int. J. Numer. Methods 65 (2011) 1365–1375.
[133] P.-H. Maire, A high-order cell-centered Lagrangian scheme for compressible fluid flows in two-dimensional cylindrical geometry, J. Comput. Phys. 228 (18) (2009) 6882–6915.
[134] P.-H. Maire, A high-order cell-centered Lagrangian scheme for two-dimensional compressible fluid flows on unstructured meshes, J. Comput. Phys. 228 (7) (2009) 2391–2425.
[135] P.-H. Maire, Contribution to the numerical modeling of inertial confinement fusion, Habilitation Thesis, Bordeaux University, 2011, available at https://tel.archives-ouvertes.fr/tel-00589758/.
[136] P.-H. Maire, R. Abgrall, J. Breil, R. Loubère, B. Rebourcet, A nominally second-order cell-centered Lagrangian scheme for simulating elastic–plastic flows on two-dimensional unstructured grids, J. Comput. Phys. 235 (2013) 626–665.
[137] P.-H. Maire, R. Abgrall, J. Breil, J. Ovadia, A cell-centered Lagrangian scheme for two-dimensional compressible flow problems, SIAM J. Sci. Comput. 29 (4) (2007) 1781–1824.
[138] P.-H. Maire, R. Loubère, P. Vachal, Staggered Lagrangian discretization based on cell-centered Riemann solver and associated hydrodynamics scheme, Commun. Comput. Phys. 10 (2011) 940–978.
[139] P.-H. Maire, B. Nkonga, Multi-scale Godunov-type method for cell-centered discrete Lagrangian hydrodynamics, J. Comput. Phys. 228 (3) (2009) 799–821.
[140] L. Margolin, M. Shashkov, Using a curvilinear grid to construct symmetry-preserving discretizations for Lagrangian gas dynamics, J. Comput. Phys. 149 (2) (1999) 389–417.
[141] L.G. Margolin, Introduction to "An Arbitrary Lagrangian–Eulerian Computing Method for All Flow Speeds", J. Comput. Phys. 135 (2) (1997) 198–202.
[142] L.G. Margolin, Arbitrary Lagrangian–Eulerian (ALE) methods a personal perspective, Technical Report LA-UR-13-24124, Los Alamos National Laboratory, 2013.
[143] L.G. Margolin, M. Shashkov, Second-order sign-preserving conservative interpolation (remapping) on general grids, J. Comput. Phys. 184 (1) (2003) 266–298.
[144] L.G. Margolin, M. Shashkov, Remapping, recovery and repair on a staggered grid, Comput. Methods Appl. Mech. Eng. 193 (39) (2004) 4139–4155.
[145] L.G. Margolin, M. Shashkov, P.K. Smolarkiewicz, A discrete operator calculus for finite difference approximations, Comput. Methods Appl. Mech. Eng. 187 (3) (2000) 365–383.
[146] D.J. Mavriplis, Revisiting the least-squares procedure for gradient reconstruction on unstructured meshes, AIAA Pap. 3986 (2003) 2003.
[147] C. Mazeran, Sur la structure mathématique et l'approximation numérique de l'hydrodynamique lagrangienne bidimensionnelle, PhD thesis, Université de Bordeaux, 2007.
[148] R. Menikoff, B.J. Plohr, The Riemann problem for fluid flow of real materials, Rev. Mod. Phys. 61 (1) (1989) 75–130.
[149] M.L. Wilkins, Calculation of elastic–plastic flow, in: B. Alder, S. Fernbach, M. Rotenberg (Eds.), Fundamental Methods in Hydrodynamics. Methods in Computational Physics, in: Advances in Research and Applications, vol. 3, Academic Press, 1964, pp. 211–262. See also Report UCRL-7463. Lawrence Radiation Lab., Univ. of California, Livermore, 1963.
[150] N.R. Morgan, M. Kenamond, D.D. Burton, T. Carney, D. Ingraham, A contact surface algorithm for cell-centered Lagrangian hydrodynamics, J. Comput. Phys. 250 (2013) 527–554.
[151] J. Nocedal, S.J. Wright, Numerical Optimization, Springer Series in Operations Research, Springer-Verlag, 1999.
[152] W.F. Noh, CEL: a time-dependent, two-space-dimensional, coupled Eulerian–Lagrange code, in: B. Alder, S. Fernbach, M. Rotenberg (Eds.), Fundamental Methods in Hydrodynamics. Methods in Computational Physics, in: Advances in Research and Applications, vol. 3, Academic Press, 1964, pp. 117–179. See also Report UCRL-7463. Lawrence Radiation Lab., Univ. of California, Livermore, 1963.
[153] J.M. Owen, M. Shashkov, Arbitrary Lagrangian Eulerian remap treatments consistent with staggered compatible total energy conserving Lagrangian methods, J. Comput. Phys. 273 (2014) 520–547.
[154] J. Pilliod Jr., E. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465–502.
[155] J.E. Pilliod Jr., E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2) (2004) 465–502.
[156] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (2) (1998) 112–152.
[157] W.J. Rider, L.G. Margolin, Simple modifications of monotonicity-preserving limiter, J. Comput. Phys. 174 (1) (2001) 473–488.
[158] M. Rudman, Volume-tracking methods for interfacial flow calculations, Int. J. Numer. Methods Fluids 24 (7) (1997) 671–691.
[159] A.A. Samarskii, V.F. Tishkin, A.P. Favorksii, M.Yu. Shashkov, Employment of the reference-operator methods in the construction of finite difference analogs of tensor operations, Differ. Equ. 18 (7) (1983) 881–885.
[160] A.A. Samarskii, V.F. Tishkin, A.P. Favorskii, M.Yu. Shashkov, On the representation of finite difference schemes of mathematical physics in operator form, Sov. Phys. Dokl. 26 (6) (1981) 590–592. American Institute of Physics.
[161] A.A. Samarskii, V.F. Tishkin, A.P. Favorskii, M.Yu. Shashkov, Operational finite-difference schemes, Differ. Equ. 17 (1981) 854–862.
[162] S.K. Sambasivan, M.J. Shashkov, D.E. Burton, A finite volume cell-centered Lagrangian hydrodynamics approach for solids in general unstructured grids, Int. J. Numer. Methods Fluids 72 (7) (2013) 770–810.
[163] S.K. Sambasivan, M.J. Shashkov, D.E. Burton, A cell-centered Lagrangian finite volume approach for computing elasto–plastic response of solids in cylindrical axisymmetric geometries, J. Comput. Phys. 237 (2013) 251–288.
[164] S.P. Schofield, R.V. Garimella, M.M. Francois, R. Loubère, A second-order accurate material-order-independent interface reconstruction technique for multi-material flow simulations, J. Comput. Phys. 228 (2009) 731–745.
[165] W.D. Schulz, Two-dimensional Lagrangian hydrodynamics difference equations, in: B. Alder, S. Fernbach, M. Rotenberg (Eds.), Fundamental Methods in Hydrodynamics. Methods in Computational Physics. Advances in Research and Applications, vol. 3, Strategies, Academic Press, 1964.
[166] M. Shashkov, Conservative Finite-Difference Methods on General Grids, vol. 6, CRC Press, 1995.
[167] M. Shashkov, Closure models for multimaterial cells in arbitrary Lagrangian–Eulerian hydrocodes, Int. J. Numer. Methods Fluids 56 (8) (2008) 1497–1504.
[168] M. Shashkov, P. Knupp, Optimization-based reference-matrix rezone strategies for arbitrary Lagrangian–Eulerian methods on unstructured meshes, Selcuc J. Appl. Math. 3 (1) (2002) 81–99.
[169] M. Shashkov, B. Wendroff, The repair paradigm and application to conservation laws, J. Comput. Phys. 198 (1) (2004) 265–277.
[170] D.P. Starinshak, J.M. Owen, A subzone reconstruction algorithm for efficient staggered compatible remapping, J. Comput. Phys. 296 (2015) 263–292.
[171] D.P. Starinshak, J.M. Owen, A multimaterial extension to subzonal reconstruction, J. Comput. Phys. 313 (2016) 594–616.
[172] J.F. Thompson, B.K. Soni, N.P. Weatherill, Handbook of Grid Generation, CRC Press, 1998.
[173] P. Vachal, R.V. Garimella, M.J. Shashkov, Untangling of 2D meshes in ALE simulations, J. Comput. Phys. 196 (2) (2004) 627–644.

[174] J. Velechovskỳ, M. Kuchařík, R. Liska, M. Shashkov, P. Váchal, Symmetry and essentially-bound-preserving flux-corrected remapping of momentum in staggered ALE hydrodynamics, J. Comput. Phys. 255 (2013) 590–611.
[175] J. Velechovskỳ, R. Liska, M. Shashkov, High-order remapping with piece-wise parabolic reconstruction, Comput. Fluids 83 (2013) 164–169.
[176] F. Vilar, P.-H. Maire, R. Abgrall, A discontinuous Galerkin discretization for solving the two-dimensional gas dynamics equations written under total Lagrangian formulation on general unstructured grids, J. Comput. Phys. 276 (2014) 188–234.
[177] F. Vilar, C.-W. Shu, P.-H. Maire, Positivity-preserving cell-centered Lagrangian schemes for multi-material compressible flows: from first-order to high-order. Part II: the two-dimensional case, J. Comput. Phys. 312 (2016) 416–442, http://dx.doi.org/10.1016/j.jcp.2016.01.037.
[178] M.L. Wilkins, Use of artificial viscosity in multidimensional fluid dynamic calculations, J. Comput. Phys. 36 (1980) 281–303.
[179] M.L. Wilkins, Computer Simulation of Dynamic Phenomena, Scientific Computation Series, Springer, 1999.
[180] A.M. Winslow, Numerical solution of the quasilinear Poisson equation in a nonuniform triangle mesh, J. Comput. Phys. 1 (2) (1966) 149–172.
[181] Z. Zeng, G. Scovazzi, A frame-invariant vector limiter for flux corrected nodal remap in arbitrary Lagrangian–Eulerian flow computations, J. Comput. Phys. 270 (2014) 753–783.
[182] C.-W. Zhang, X. Shu, Maximum-principle-satisfying and positivity-preserving high-order schemes for conservation laws: survey and new developments, Proc. R. Soc. Lond. A, Math. Phys. Eng. Sci. 467 (2011) 2752–2776.
[183] J. Zukas, Introduction to Hydrocodes, Studies in Applied Mechanics, vol. 49, Elsevier, 2004.